Master Thesis

Evaluating RAG Pipelines with Real-World User Data

Johann Zapf
(matriculation number 1657576)

September 2, 2025

Submitted to
Data and Web Science Group
Prof. Dr. Christian Bizer
University of Mannheim

Abstract

This thesis proposes a unified framework for evaluating Retrieval-Augmented Generation (RAG) pipelines using real-world enterprise data and live user preference feedback. Motivated by the lack of empirically validated evaluation metrics for RAG, as well as the need to distinguish between retrieval and generation performance, the study combines controlled pipeline experiments with data from a production RAG application serving 250 employees, along with live user feedback from that system.

Retrieval evaluation covers embedding models, chunking strategies, `top_k` selection, and reranking, assessed using recall and LLM-based Context Relevance. Results show that large embedding and reranking models generally improve retrieval, though the optimal context size depends on the dataset and query characteristics. In addition, evaluations demonstrate that Context Relevance is a promising metric for evaluating retrieval without ground truth.

Generation evaluation compares traditional metrics (BLEU, ROUGE, BERTScore), reference-guided and reference-free LLM-as-a-Judge approaches, and RAG-specific scores (Groundedness, Answer Relevance, RAG Triad) against a human-aligned baseline. Results show that larger LLMs generally improve answer quality and that reference-guided LLM-as-a-Judge metrics have the highest correlations with human judgement, closely followed by ROUGE, BLEU, and BERTScore-Recall. Reference-free LLM-based metrics can also yield moderate correlations with the baseline, though their performance depends on the metric configuration and the underlying data and is observed to be generally lower than given in the corresponding literature. In general, the impact of using reference answers is identified as higher than the impact of using LLMs as evaluators.

This work delivers a reproducible evaluation pipeline, a novel real-world RAG dataset, large-scale empirical results on RAG pipeline design and metric validity, and a live human feedback approach adapted to RAG, offering guidance for both academic research and industrial RAG deployment.

Contents

List of Figures

List of Tables

1. Introduction

In recent years, large language models (LLMs) have unlocked powerful new capabilities for natural language understanding and generation. However, these models continue to face two practical limitations in real-world deployments: restricted access to recent or proprietary knowledge and the tendency to produce confident but ungrounded statements (commonly referred to as hallucinations).

Retrieval-Augmented Generation (RAG) addresses both issues by retrieving relevant content from files and other external data sources and conditioning the LLM's answers on that evidence. Designing effective RAG systems, however, remains challenging: Retrieval quality depends on factors such as embedding models, chunking strategies, the number of retrieved chunks, and reranking methods, while generation quality is influenced by all of these parameters in addition to the choice of language model.

These design considerations necessitate metrics that can accurately determine which configurations perform better than others. A variety of metrics have been proposed for evaluating RAG systems, ranging from traditional measures such as recall and n-gram-based scores to embedding-based approaches, and more recently, LLM-driven metrics, including those specifically designed for RAG.

However, the performance of certain metrics remains unclear in the context of RAG, and several widely cited RAG metrics lack rigorous empirical validation (Yu et al. 2024; Es et al. 2023; tru 2025). This motivates a careful, end-to-end study of RAG system design and evaluation metrics grounded in real data.

Some of these new LLM-driven metrics operate without ground-truth labels, thereby eliminating the resource-intensive process of acquiring accurate annotations. For this reason, this work focuses on comparing the performance of such reference-free metrics against metrics that rely on ground-truth annotations.

Consequently, this thesis proposes a unified framework for evaluating RAG pipelines across both retrieval and generation, combining controlled pipeline experiments with data from a production RAG application and live user preferences. A central contribution is a new dataset for RAG evaluation that is derived from real prompts in sovanta AG's Document Chat application, a production RAG system used by approximately 250 employees to query company documents, private uploads, and external knowledge bases such as SharePoint and Confluence. In contrast to synthetic or LLM-generated benchmark datasets, this corpus captures the genuine usage patterns of a production RAG system, including multilingual queries, heterogeneous sources, and questions that are not answerable based on the underlying context.

To ensure comparability, all metrics are assessed under a consistent experimental setting and against meaningful baselines. Metrics are computed over thousands of pipeline runs, varying key RAG hyperparameters to measure their effects on outputs. Retrieval

and generation are evaluated separately, enabling independent measurement of the performance of each stage. This is in contrast to many RAG evaluations that evaluate both stages together (Yu et al. 2024).

In addition to offline evaluation, this work incorporates live side-by-side preference feedback directly into the production Document Chat application. With a controlled variation of the LLM used and the number of retrieved chunks, users compare answers in pairs and select the preferred one. These live preferences are then used to complement and validate the findings from the offline evaluations.

In total, this thesis makes six main contributions:

1. A unified RAG evaluation framework that separates retrieval and generation and enables comparisons of different RAG pipeline configurations and evaluation metrics.

2. A new RAG evaluation dataset that is based on data from a real-world enterprise RAG application and contains multilingual queries, heterogeneous sources, and explicit unanswerable questions.

3. A large-scale retrieval study that quantifies the effects of embeddings, chunking, `top_k`, and reranking, and assesses the performance and limitations of the LLM-based Context Relevance metric.

4. A comprehensive generation evaluation that evaluates various RAG hyperparameter settings and measures the performance of traditional metrics (BLEU, ROUGE, BERTScore), LLM-as-a-Judge approaches, as well as RAG-specific scores in terms of correlation with a human-judgement-aligned baseline.

5. A method for analyzing metric errors through an LLM-based heuristic.

6. A live user preference evaluation approach that is specifically adapted to RAG.

The remainder of this thesis is organized as follows: Chapter 2 reviews RAG, retrieval and generation evaluation, and relevant metrics, motivating the need for a unified evaluation approach. Chapter 3 details the datasets, preprocessing steps, the evaluation pipeline setup, and live user feedback collection. Chapter 4 reports the retrieval and generation results with metric comparisons, runtime and token analyses, and error studies. Finally, Chapter 5 discusses the results and their implications and Chapter 6 concludes with key findings, limitations, and directions for future work.

All accompanying code and datasets are available on GitHub.[1]

[1] https://github.com/johannzapf/masterthesis

2. Related Work

2.1. Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) systems have emerged as a powerful paradigm to enhance the capabilities of large language models (LLMs) by integrating external knowledge. These systems effectively address two common challenges of LLMs: access to recent and proprietary data and the tendency of LLMs to produce hallucinations (Lewis et al. 2020; Huang et al. 2025).

The process of answering a user query in RAG is divided into two distinct phases: retrieval and generation. To this end, RAG operates on a database of predefined context that is used by the system to answer the query. For this reason, the first step in any RAG pipeline is the ingestion of the context into a vector database. The context can be anything that is supported by modern language and embedding models, i.e., text-based content such as documents, web pages, and tabular data, but also images and other types of content. Nevertheless, most RAG applications and evaluations focus on text. This content is first split into chunks of a specific size, for example 256 tokens, and then transformed into vector representations by an embedding model. The goal of the chunking is to produce self-contained pieces of information that are as inherently complete as possible. For this reason, several studies focus exclusively on optimizing chunking for RAG (Zhong et al. 2024). The resulting tuples of vectors and their corresponding content are then stored in the index of a vector database (Lewis et al. 2020). The ingestion step therefore yields a database that may contain millions of embedded content chunks that can be used to answer queries.

To make use of this indexed knowledge, the goal of the retriever is to identify the chunks in the vector database that are relevant for answering a given query. The retriever component achieves this by calculating the embedding of the query and finding a predetermined number of chunks (`top_k`) that have the highest cosine similarity with the query and thus share the closest semantic meaning (Lewis et al. 2020). The speed of this search may be improved with algorithms such as the hierarchical navigable small-world (HNSW) index, which performs an approximate nearest neighbor search for increased performance (Ma et al. 2023). The hyperparameter `top_k` is already one evaluation target, though it is also influenced by the available context size of the generator and the necessary response time, which usually increases with larger context sizes (see Section 4.2.8).

It should be noted that this only represents the most commonly used approach for retrieval; there are also other possible realizations for the retrieval step, for example approaches based on TF-IDF and BM25, which rely on keyword matching rather than embeddings (Chen et al. 2024).

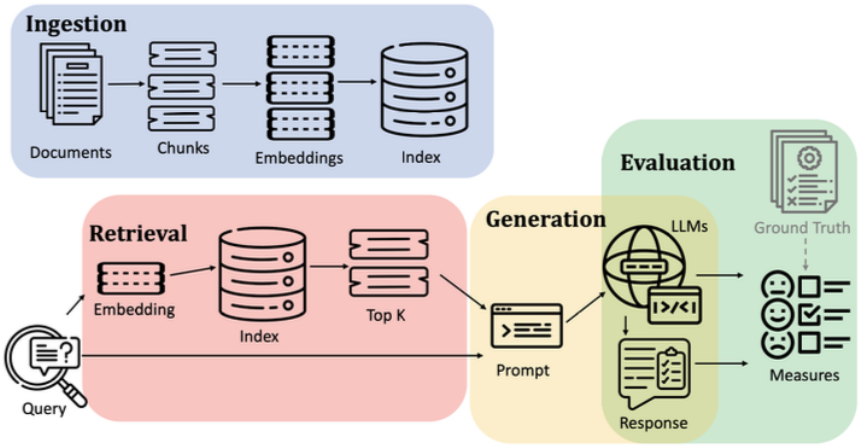

Figure 2.1.: Visualization of a Retrieval-Augmented Generation system (Oro et al. 2024)

In order to improve the results of the retrieval step, reranking procedures can be employed. These are specialized models that are trained to output similarity scores for each chunk given a query. An example of such a model is the BGE-Reranker, which is based on the BGE-M3 embedding model (Chen et al. 2024; bge 2024). There are different ways to set up such a reranking step. A common approach is to have the base retriever select more than `top_k` chunks (e.g., twice as many) and to use the reranker to make the final selection of `top_k` chunks (Mortaheb et al. 2025).

When the relevant passages have been selected by the retriever, they are passed to the generator along with the original query. The task of the generator is to answer the question based on the retrieved context and its prior knowledge. In practice, the generator component is realized by an LLM that is able to find the answer to the query in the retrieved context or to state that the question is not answerable. This architecture is a primary reason why RAG applications are generally less prone to hallucinations (Yu et al. 2024; Huang et al. 2025).

The generation step can be optimized with different system prompts and with techniques like Chain-of-Thought (CoT) reasoning. One common approach is to add an instruction to the system prompt that forbids the LLM from using its prior knowledge, forcing it to only focus on the retrieved information and therefore reducing the probability of hallucinations (lla 2025). Figure 2.1 provides a visual summary of the steps described above.

From this section it already becomes clear that RAG systems involve many hyperparameters that can be optimized. The following list is an overview of the most important ones and also represents the parameters that we optimize for in our study:

- **Embedding Model**: The embedding model used for retrieval
- **Chunk Size**: The maximum size (in tokens) for splitting input content into chunks
- **top_k**: The number of chunks to select during retrieval

- **Reranking**: Whether reranking is applied, and if so, which model is applied
- **Language Model**: The choice of LLM used as the generator

Since the performance of a RAG system depends on the specific configuration of these parameters, thorough evaluation is essential for determining which combinations yield the best results. Therefore, the following section reviews methods for evaluating RAG pipelines.

2.2. RAG Evaluation

Since retrieval and generation are largely independent processes in RAG, evaluations must assess these two stages separately. For the retrieval phase, the focus is on evaluating the relevance of the retrieved context to the input query. In contrast, the generation phase is assessed based on the quality of the final response with respect to both the query and the retrieved context (Yu et al. 2024).

As highlighted in the survey by Yu et al., although substantial work has been done on RAG evaluation, most approaches primarily assess the system's final output while neglecting the intermediate output produced by the retriever (Yu et al. 2024). We argue that this is a significant oversight. As LLMs become more capable, optimizing the retrieval component is increasingly crucial for overall performance and thus warrants greater attention from the research community.

Given the differing evaluation objectives for each component, distinct metrics are used for retrieval and generation. The following section provides an overview of these metrics and their development, categorizing them into those that rely on ground-truth labels and those that do not.

2.2.1. Retrieval Evaluation

In addition to the distinction between metrics that do or do not require ground-truth labels, which applies to all metrics discussed here, retrieval metrics can be further divided into rank-based and non-rank-based metrics. Non-rank-based metrics assess binary outcomes, i.e., whether the retrieved context contains the relevant information. These metrics do not consider the position of the relevant information piece in the context, meaning that there is no penalty for finding unrelated content first. On the other hand, rank-based metrics consider the position of the relevant information piece in the context, giving higher scores if the relevant content is at the start of the retrieved context.

Non-rank-based ground-truth metrics

The first type of retrieval metrics is ground-truth-based metrics that do not consider the order of the retrieved items. Ground-truth-based means that these metrics require gold labels that define which pieces of content are relevant to a certain query and therefore need to be retrieved. The evaluation is then done by comparing this relevant content to what the retriever component has actually retrieved given the query.

The metrics in this class are familiar metrics of data science and are calculated at per-chunk level, meaning that each retrieved chunk is evaluated and the results are aggregated across chunks to yield the final score. Accuracy is the proportion of correctly identified relevant and irrelevant chunks. Precision is the fraction of relevant chunks among the retrieved chunks and Recall@k measures the fraction of relevant instances that have successfully been retrieved by the retriever. In this case, k is simply the `top_k` value used by the retriever (Yu et al. 2024).

Among these three metrics, recall is arguably the most meaningful. The key concern in retrieval evaluation is whether all relevant text pieces have been found. Cuconasu et al. even show that although adding similar documents that do not contain the answer to the context reduces LLM performance, adding random, unrelated documents can even improve the output accuracy of LLMs by up to 35% (Cuconasu et al. 2024). Recall is therefore a particularly suitable metric for retrieval, as LLMs are generally robust to irrelevant information — and may even benefit from certain types of noise — but they perform poorly in RAG applications when relevant content is missing.

Since these metrics are mostly calculated based on text, one question that naturally arises is how to handle retrieved context that only partially contains the relevant information outlined in the gold labels. The standard approach of the metrics is to assign a score of 0 to partially matching chunks, treating them as entirely irrelevant. For this reason, optimized metrics have been proposed that take the longest common substring (LCS) into account (Lin and Och 2004). To our knowledge, this issue has not been explicitly addressed in retrieval evaluation literature, so we also take this into account.

Rank-based ground-truth metrics

As mentioned before, ground-truth-based metrics that consider the rank also evaluate the position of the relevant sections in the retrieved chunks. This means that if there are multiple relevant items across multiple documents that should be retrieved, the gold labels also need to consider the relative importance of these items. This consideration may increase the annotation effort required to produce ground-truth labels.

The three main metrics in this class are Mean Reciprocal Rank, Mean Average Precision, and Normalized Discounted Cumulative Gain. Mean Reciprocal Rank (MRR) calculates the multiplicative inverse of the rank of the first relevant item among the retrieved chunks, i.e., 1 for first, 0.5 for second, and so on. Mean Average Precision (MAP) works by calculating the precision for every possible cutoff k (with $1 \leq k \leq$ `top_k`) of the retrieved chunks and then dividing by the number of relevant documents. This yields a rank-sensitive precision measure (Yu et al. 2024).

Normalized Discounted Cumulative Gain (NDCG) was introduced in 2002 by Järvelin and Kekäläinen and works by comparing the retrieval results to an ideal order. The gain of each relevant item in the retrieved list is discounted by a logarithmic function of its rank. This Discounted Cumulative Gain (DCG) is then normalized by the DCG of the ideal ranking to get the NDCG (Järvelin and Kekäläinen 2002). We leave out the calculation details for these metrics here and refer to Yu et al. 2024 and Järvelin and Kekäläinen 2002 instead.

An open question that remains is whether the order of the retrieved chunks matters in a RAG system, and therefore which type of metric should be preferred, especially since rank-based metrics usually require greater labeling effort. In 2023, Liu et al. showed that placing relevant information at the start or end of the context that is input to an LLM leads to better results when querying that information than placing it in the middle. Their evaluation uses two distinct tasks: answering questions based on multiple documents and retrieving key-value pairs of UUIDs from randomly generated JSON content. For question answering, all tested LLMs perform worse when the relevant content appears in the middle of the input, although Claude models are significantly less affected than GPT models. For key-value retrieval, some models also exhibit reduced accuracy with middle-positioned keys, while others maintain consistent performance regardless of key position (Liu et al. 2023).

While these findings offer compelling evidence that positional effects can impact LLM performance, some limitations must be considered. The models used in the study are relatively outdated and feature small context windows compared to today's more capable LLMs. It is plausible that modern models, with significantly expanded context windows and potential fine-tuning for position robustness, may exhibit reduced sensitivity to such effects. Also, the generalization of the results to real-world RAG retrieval scenarios remains uncertain.

Importantly, the results imply that an ideal evaluation metric should penalize the presence of relevant content in the middle of the context more heavily than at the beginning *or end*. However, to our knowledge, no existing metric incorporates such positional asymmetry. Nevertheless, the study reinforces the potential value of using rerankers in RAG pipelines to reorder retrieved content for improved model performance.

Context Relevance

Unlike traditional retrieval metrics that rely on ground truth, recent studies have proposed LLM-based retrieval metrics that operate without it, with Context Relevance being the most notable example. As highlighted in the survey by Yu et al., this remains essentially the only LLM-based retrieval metric introduced so far (Yu et al. 2024), underscoring the limited emphasis of current RAG evaluations on the retrieval aspect.

Context Relevance was introduced by the authors of the RAGAs evaluation framework in 2023 and several variants of the metric have been proposed since (tru 2025; Saad-Falcon et al. 2023). Context Relevance uses an LLM to judge whether retrieved context is relevant to answering a given query (Es et al. 2023).

In its original variant, Context Relevance is calculated by using an LLM to extract a subset of sentences from the context that is relevant to answering the question and then dividing the number of these sentences by the total number of sentences in the context. This means that the metric evaluates to what extent the context *exclusively* contains information that is relevant to the query (Es et al. 2023). This variant of the metric is therefore more closely related to precision than to recall.

In the version proposed by the authors of the ARES evaluation framework, Context Relevance is obtained by asking a fine-tuned version of DeBERTa-v3-Large with a binary

classifier head whether the retrieved context is sufficient for answering a given question or not. The fine-tuning was performed on their human preference validation set (Saad-Falcon et al. 2023).

Finally, the TruLens evaluation framework obtains the Context Relevance by asking the LLM to evaluate the retrieved context given a query on a scale of 0 to 3, giving additional instructions on how the context should be interpreted. It also provides the option to use CoT reasoning and allows setting a custom evaluation scale. Both the TruLens and the ARES version of the metric aim to evaluate whether the context contains the relevant information and do not consider irrelevant information (tru 2025). They are therefore more closely related to recall than to precision.

One key question is how these metrics are validated. The authors of RAGAs created a dataset called WikiEval, which is based on 50 Wikipedia pages covering events after the knowledge cutoff of their tested LLMs. For each page, they asked ChatGPT to create a question-answer-context triplet and then created a secondary context column (context_v2) that contains additional, irrelevant text (wik 2023). Two human annotators evaluated the Context Relevance of these data points with a mutual agreement of 95%.

The final evaluation was conducted by counting how often the preferred context by the humans coincides with the context that yields the higher LLM-based Context Relevance score. This yields an accuracy of 0.70. The authors compare this result to a baseline they call GPT Score, where they simply ask ChatGPT to give a score between 1 and 10 for a question-context pair in terms of Context Relevance. This baseline achieves a score of 0.63 (Es et al. 2023). Although this result generally validates their approach, the evaluation design leaves open questions. For instance, it does not report the correlation between the LLM-derived scores and human judgements across diverse retrieval systems. Additionally, no direct comparison is made with classical retrieval metrics such as recall or precision, which would further validate the value of Context Relevance.

The authors of ARES evaluate their metric using a synthetic benchmark consisting of nine mock RAG systems, each with 150 query–context–answer triplets, totaling 1,350 data points. Each mock system was constructed to have a predefined success rate ranging from 0.7 to 0.9 in 0.025 increments by manually introducing controlled negative examples. This setup allows the authors to define a ground-truth ranking of the systems. The evaluation measures how well ARES can recover this known ranking using its Context Relevance scores. The authors compare ARES to RAGAs on this task and find that ARES performs better, with a Kendall's tau that is 0.065 higher on average, indicating more accurate system ranking (Saad-Falcon et al. 2023). TruLens does not provide any evaluations of its metrics (tru 2025).

Table 2.1 summarizes the discussed retrieval metrics.

2.2.2. Generation Evaluation

Similar to retrieval evaluation, metrics for assessing the generation component in RAG systems can be broadly categorized into two groups: ground-truth-based metrics, which rely on reference answers, and metrics that assess output quality without requiring gold-standard labels. In this section, we begin with traditional metrics that originated prior

Table 2.1.: Summary of retrieval evaluation metrics

Metric	Type	Description
Accuracy (Yu et al. 2024)	Ground truth	Proportion of correctly identified relevant and irrelevant chunks
Precision (Yu et al. 2024)	Ground truth	Fraction of relevant chunks among the retrieved ones
Recall@k (Yu et al. 2024)	Ground truth	Fraction of relevant chunks retrieved out of all relevant chunks, up to `top_k`
LCS-based Recall@k (Lin and Och 2004)	Ground truth	Uses longest common substring to compute partial match score
Mean Reciprocal Rank (MRR) (Yu et al. 2024)	Rank-based ground truth	Reciprocal of the rank of the first relevant chunk
Mean Average Precision (MAP) (Yu et al. 2024)	Rank-based ground truth	Average precision at all relevant ranks, normalized
Normalized Discounted Cumulative Gain (NDCG) (Järvelin and Kekäläinen 2002)	Rank-based ground truth	Gain discounted logarithmically by rank and normalized by ideal gain
Context Relevance (RAGAs) (Es et al. 2023)	LLM judgement	Proportion of relevant sentences in context extracted by LLM
Context Relevance (ARES) (Saad-Falcon et al. 2023)	LLM judgement	Binary LLM classifier determines sufficiency of context
Context Relevance (TruLens) (tru 2025)	LLM judgement	LLM scores context relevance on a custom scale (e.g. 0–3)

to the development of transformer-based models and RAG architectures and then turn to more recent LLM-based evaluation approaches.

Human Judgement

Despite the rise of many metrics for generation evaluation, human judgement remains the gold standard. This is intuitive, as the primary criterion for any LLM-generated output is its perceived value and correctness from a human perspective. For this reason, most generation metrics are validated by comparing their outputs to human judgement (Papineni et al. 2002; Lin 2004; Zhang et al. 2020; Zheng et al. 2023).

An important question in this regard is what constitutes a good correlation with human judgement. Zheng et al. created the MT-Bench dataset, which contains 80 multi-turn questions across eight categories. They then let six different LLMs generate answers for these questions and presented pairs of these answers to 58 expert human labelers, who chose the better answer. In this manner, 3,000 votes were collected. They

use this data to validate their own LLM-based judges, but, more importantly here, also show that the agreement among the human judges is only 0.81 (Zheng et al. 2023). This implies that any LLM-based judge achieving a higher agreement with human judgement than 0.81 is more closely aligned with human preferences than humans are with one another and can therefore be seen as a meaningful replacement for human judges.

This setup already describes one popular way of incorporating human judgement into RAG evaluations: creating a dataset with ground-truth labels, running the RAG pipeline on it using different hyperparameter combinations and then comparing the output of the metric being evaluated with the output of human annotators. Chiang et al. call this a *static* approach. On the other hand, *live* approaches use preferences from real users on open questions. These settings are substantially more scalable because they do not require a dataset with ground-truth labels and are better at mirroring actual real-world usage (Chiang et al. 2024).

One such live approach is the benchmarking platform LMArena (formerly known as Chatbot Arena), which was introduced in 2024. LMArena is a free and open website where users can send prompts and get replies from two anonymous LLMs. The user then casts a vote for the answer that is deemed better, after which the identity of the LLMs is revealed. This crowdsourced data is then used to construct a leaderboard of LLMs (Chiang et al. 2024).

By August 2025, over three million votes across 230 models have been collected. The current top three in its leaderboard are Gemini-2.5-Pro, GPT-5, and Claude-4-Opus (lma 2025). According to its authors, LMArena is now one of the most referenced LLM leaderboards and is often cited by leading LLM developers (Chiang et al. 2024).

Despite this success, such an approach has to our knowledge not been applied to RAG yet. While LMArena focuses on varying a single parameter (the underlying LLM), RAG systems introduce a more complex evaluation space. They involve multiple configurable components such as retrievers, rerankers, and generation models, as well as tunable hyperparameters like `top_k` and chunk size. A benchmarking platform for RAG would thus enable more fine-grained insights into how different configurations affect human preferences.

BLEU

BLEU (Bilingual Evaluation Understudy), introduced by Papineni et al. in 2002, is a widely used metric for evaluating the quality of machine translation systems. It compares a candidate translation against one or more high-quality human-created references by analyzing overlapping n-grams. Although initially developed for machine translation, BLEU has since been adopted for evaluating tasks such as text summarization and image captioning (Papineni et al. 2002; Liu et al. 2023).

At its core, BLEU measures the precision of n-grams in the candidate text with respect to the reference text, i.e., the number of candidate text n-grams that appear in any reference text divided by the number of n-grams in the candidate text. In order to prevent high scores for text candidates that simply repeat an n-gram that is present in the reference text, BLEU introduces modified, clipped n-gram precision. To achieve

this, the appearance count of each candidate n-gram is upper-bounded by the maximum number of times it occurs in any reference translation. This value is then summed for each n-gram in the candidate text and divided by the total number of candidate n-grams. For example, without clipping, the candidate sentence "the the the the the the the" would receive a perfect precision score against the reference "the cat is on the mat". However, BLEU clips repeated n-grams, resulting in a much lower score of 2/7 (Papineni et al. 2002).

In the next step, the geometric mean for the modified n-gram precision is computed for $1 \leq n \leq N$. The authors of BLEU suggest $N = 4$, which is also the default value used by the HuggingFace implementation of BLEU (hug 2025; Papineni et al. 2002).

Finally, BLEU also contains a sentence brevity penalty BP, defined as

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{1-\frac{r}{c}} & \text{if } c \leq r \end{cases}$$

where c is the length of the candidate text and r is the length of the reference text that is closest in length to the candidate text (Papineni et al. 2002).

The final BLEU score is then calculated as

$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

where BP is the brevity penalty, p_n is the modified n-gram precision, w_n are the weights for precision (usually uniform, i.e., $w_n = 1/N$), and N is the number of n-grams to consider (Papineni et al. 2002).

The dataset used for the original BLEU evaluation is composed of 500 sentences from 40 general news stories. Each sentence was translated independently by two professional translators as a reference and then translated by three machine translation systems and two non-professional humans with varying language proficiency. The dataset exclusively contains Chinese-to-English translations (Papineni et al. 2002).

To evaluate BLEU, the authors compute the correlation of the BLEU score with human judgements on these five systems. The human judges were divided into a group of 10 native English speakers (the monolingual group) and a group of 10 people proficient in English and Chinese (the bilingual group). Each translation was rated by the judges on a scale of 1 to 5. The result is a correlation of 0.99 between BLEU and the monolingual group and 0.96 between BLEU and the bilingual group, indicating that BLEU tracks human judgement very well (Papineni et al. 2002).

Follow-up work by the authors of BLEU also evaluates their metric on other language pairs. Here, they use the DARPA-94 evaluation dataset which contains professional translations of 100 documents from Spanish to English and French to English and the same groups of human judges. The result is an average correlation of 0.93 for both languages (Ward and Reeder 2002).

ROUGE

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) was introduced in 2004 and is a family of metrics that can be used to score output text against high-quality reference text. It was initially proposed as a metric to score computer-generated summaries based on ideal summaries created by humans, but has since been applied to various other NLP tasks, including the evaluation of RAG generation outputs (Lin 2004; Yu et al. 2024).

Similarly to BLEU, ROUGE is based on measuring word overlap between the candidate text and a reference text. It was initially introduced as a recall-based metric — where BLEU is precision-based — but already the original work provides formulas for recall, precision, and F1 score for most variants (Lin 2004). By now, ROUGE evaluations mostly report the F1 score (Ganesan 2018).

The first metric is ROUGE-N, which evaluates the n-gram overlap between candidate and reference text. ROUGE-N recall is calculated as the number of n-gram matches between the candidate and the reference text divided by the total number of n-grams in the reference text (Lin 2004). ROUGE-N precision is therefore the number of n-gram matches divided by the number of n-grams in the candidate text and F1 score follows in the usual manner (Ganesan 2018). The most commonly used variants of ROUGE-N are ROUGE-1, evaluating unigrams, and ROUGE-2, evaluating bigrams (Lin 2004).

In contrast, ROUGE-L considers the longest common subsequence (LCS) between two sentences instead of n-gram overlap. Longest common subsequence here means that this sequence of unigrams does not necessarily need to be consecutive, but must be in the same order. ROUGE-L recall is then defined as the LCS between candidate and reference sentence divided by the number of unigrams in the reference sentence and ROUGE-L precision is the LCS divided by the number of unigrams in the candidate sentence (Lin 2004).

As ROUGE metrics are typically computed on a sentence level, the author also proposes a generalization for longer texts that works by using the union LCS matches between a reference text sentence and each sentence in the candidate text. An adaptation of the metric for multiple references and further metrics like ROUGE-W, ROUGE-S, and ROUGE-SU have also been proposed (Lin 2004). However, these variants are substantially less common in practice, especially for RAG applications (Yu et al. 2024; Liu et al. 2023), so we refer to the original work for their definition.

The original evaluation of the ROUGE metric is based on a summarization evaluation dataset from the Document Understanding Conference (DUC). It contains 5,918 single-document summaries generated by 26 systems, 8,736 very short summaries generated by 14 systems, and 4,288 short multi-document summaries generated by 110 systems. The dataset includes human judgements for all generated summaries (Lin 2004).

Table 2.2 gives a summary of the original ROUGE evaluation results. ROUGE-2 and ROUGE-L perform very similarly on single-document summarization and all metrics benefit from a higher sample size. ROUGE-L works best on short summaries and ROUGE-2 on multi-document summarization (Lin 2004).

The author also finds that using multiple references improves the results, that stem-

Table 2.2.: ROUGE metrics correlation with human judgement across different settings (Lin 2004)

Metric	Single-Doc Summary 149 Samples	Single-Doc Summary 295 Samples	Short Summary	Multi-Doc Summary
ROUGE-1	0.76	0.98	0.96	up to 0.84
ROUGE-2	**0.84**	**0.99**	0.75	**up to 0.93**
ROUGE-L	0.83	**0.99**	**0.97**	up to 0.88

ming does not have a significant positive effect on the results, and that stopword removal can improve the results for multi-document summarization (Lin 2004).

BERTScore

One major drawback of n-gram-based metrics such as BLEU and ROUGE is that they measure word overlap rather than the actual meaning of candidate and reference text. As a result, they often fail to match paraphrases that are semantically similar but lexically different. Additionally, they fail to capture semantically important ordering changes, for example when swapping cause and effect clauses (Yu et al. 2024; Zhang et al. 2020).

BERTScore was introduced in 2020 to address these issues and is now a widely used metric to evaluate the similarity between generated and reference text. Its core idea is to use contextual embeddings of both texts to evaluate the semantic similarity between them. This is achieved by tokenizing the candidate and reference text and then transforming them into vector representations using a contextual embedding model (Zhang et al. 2020). A contextual embedding model like BERT generates different vectors for the same word depending on the surrounding words, thereby capturing the word's meaning in context. This contextual understanding is achieved during training through attention mechanisms (Devlin et al. 2019).

In BERTScore, the resulting vectors are then used to calculate the pairwise cosine similarity of the embedding of each token in the candidate text with the embedding of each token in the reference text. To calculate recall, each token x_i in the reference sentence x is matched to the most similar token y_j in the candidate sentence y and each y_j is matched to the most similar x_i for precision. The final values for precision and recall are then given as

$$\textbf{Recall:} \quad R = \frac{1}{|x|} \sum_{x_i \in x} \max_{y_j \in y} \cos(\vec{x_i}, \vec{y_j}) \quad \textbf{Precision:} \quad P = \frac{1}{|y|} \sum_{y_j \in y} \max_{x_i \in x} \cos(\vec{y_j}, \vec{x_i})$$

and the F1 score follows as usual (Zhang et al. 2020).

The authors also propose a variant of BERTScore that uses inverse document frequency (idf) weighting to incorporate the notion that rare words are more relevant to sentence similarity than common words. However, they later demonstrate that the effectiveness of this adjustment is domain-dependent and therefore recommend not to use it for most tasks (Zhang et al. 2020).

It should be noted that BERTScore can in principle be computed using any embedding model. Its authors evaluate twelve different models, including variants of BERT, RoBERTa, and XLM, and conclude that RoBERTa-large works best for English and BERT-cased-multilingual for other languages (Zhang et al. 2020). However, since the release of BERTScore, more capable embedding models such as OpenAI's text-embedding-3-large have been released (ope 2024) and may outperform the original models evaluated in the BERTScore paper.

To evaluate their metric, the authors of BERTScore compare its correlation with human judgement against that of other metrics using the WMT18 dataset. WMT18 is a machine translation dataset that contains predictions of 149 translation systems across 14 language pairs, gold answers, and human judgements (Bojar et al. 2018). Its test set portion that is used by the authors contains the translations of 2,998 sentences. Results show that BERTScore-F1 ranks among the top-performing metrics with correlations of around 0.99 for most languages, consistently outperforming BLEU. However, it should be noted that BLEU also achieves consistently good results with correlations of around 0.97 for most languages, making the improvements of BERTScore relatively small in absolute terms (Zhang et al. 2020).

Furthermore, the BERTScore metric is also evaluated on the validation set of the COCO 2015 image captioning dataset. It contains 80,000 images with five reference captions each, predictions from twelve captioning systems, and human judgement scores (Lin et al. 2014; Zhang et al. 2020). In this setting, ROUGE and BLEU achieve near-zero correlation with human judgements, showing their inability to capture the semantic meaning of the captions in this more challenging evaluation. In contrast, BERTScore-F1 achieves a correlation of 0.32, BERTScore-Recall 0.89 and idf weighted BERTScore-Recall even 0.92 (Zhang et al. 2020).

One more consideration is the speed of the evaluation metrics. Embedding-based metrics are inherently slower than n-gram-based metrics because they rely on large pre-trained models. Thus, computing BERTScore on 3,000 sentences with an NVIDIA GTX-1080 GPU takes 16 seconds, compared to 5 seconds for BLEU (Zhang et al. 2020). Although this represents a three-fold increase in inference time, BERTScore remains significantly faster than metrics based on large language models, which are discussed in the following section.

LLM-as-a-Judge

Since the emergence of powerful LLMs in 2022, numerous approaches have explored leveraging these models as evaluators of machine-generated outputs, offering an alternative to human annotations and n-gram-based metrics that often fail to capture the semantic nuances of language. These approaches have also been increasingly applied to RAG (Yu et al. 2024).

Zheng et al. were among the first to propose the use of LLM-as-a-Judge for evaluating chatbot responses. They introduce three distinct evaluation metrics: pairwise comparison, single answer grading, and reference-guided grading. In pairwise comparison, the LLM is presented with a user query and two candidate responses and is tasked with

selecting the better response. In single answer grading, the model assigns a quality score to a single response without requiring a reference answer. Both of these methods operate without ground-truth labels. In contrast, reference-guided grading incorporates a reference (ground-truth) answer alongside the response and query, enabling the LLM to evaluate the response based on its alignment with the reference. The system prompts for all of these metrics are available in the appendix of the original paper (Zheng et al. 2023).

These foundational metrics have since influenced a wide range of subsequent LLM-as-a-Judge approaches. For example, the Answer Correctness metric used in RAGAs is a variation of reference-guided grading (Es et al. 2023; Oro et al. 2024). Furthermore, Zheng et al. evaluate variations on their initial approach, for example using CoT reasoning and few-shot prompts (Zheng et al. 2023).

Similarly to most other approaches presented here, Zheng et al. evaluate their metrics by comparing their outputs to human annotations. For the evaluation, they use the previously described MT-Bench and LMArena datasets. For LMArena, they randomly sample 3,000 votes that cover eight different LLMs. These votes and the 3,000 votes from the MT-Bench dataset are then compared with the LLM-generated votes from pairwise comparison and the results from single answer grading, which are turned into pairwise comparison scores for the evaluation. They then compute the agreement between two types of judges, i.e., the probability that they give the same vote on a question (Zheng et al. 2023).

Of their tested judges, GPT-4 performs best with a pairwise comparison agreement of 0.66 with human votes on both datasets. With single answer grading, GPT-4 achieves an agreement of 0.6 on MT-Bench and 0.62 on LMArena. Since both datasets also allow for tie votes, the authors also report the scores when excluding tie votes. This increases agreement to 0.85 between GPT-4 and humans for both datasets and metric variants, which is higher than the agreement that humans have with one another. GPT-3.5 and Claude-1 also demonstrate reasonable performance, although GPT-4 performs better (Zheng et al. 2023).

While the results for pairwise comparison and single answer grading are comprehensive, the authors provide less detailed evaluations for reference-guided grading. They observe that LLM judges struggle with math and reasoning questions and thus evaluate 20 such questions using reference-guided grading, showing that it reduces the error rate from 0.7 to 0.15. However, they do not report the full results of reference-guided grading on their two datasets, which is likely because the two datasets do not contain gold answers. Furthermore, their limited evaluation of reference-guided grading is based on LLM-generated reference answers instead of human-written gold labels (Zheng et al. 2023). This suggests that while reference-guided grading exhibits significant potential, the true performance of the metric remains unclear. Furthermore, this metric could be compared with other ground-truth-based metrics like BLEU and ROUGE to clarify its value. We identify this as a weakness of the work.

In addition to quantitative evaluations, Zheng et al. highlight several systemic biases that affect the reliability of LLM judges and that need to be considered when using such an approach. The first problem is position bias, which means that the LLM judgements

in pairwise comparison depend on the order of the two answers. They find that all LLMs are at least somewhat susceptible to position bias, with GPT-4 being affected the least. Furthermore, they show that the phenomenon is more present on open questions like writing than on math or coding questions. The authors suggest two simple solutions for this problem: assigning positions randomly and running judgement with both orders and taking the average (Zheng et al. 2023).

The second problem is verbosity bias, which means that the LLM judge favors longer responses even if they are of lower quality. They show that Claude-1 and GPT-3.5 are prone to this issue, but that GPT-4 is relatively robust to it. Finally, self-enhancement bias is the notion that LLMs favor outputs generated by themselves. Zheng et al. show that certain LLMs seem to be susceptible to this bias, but find no conclusive evidence that it is a general phenomenon (Zheng et al. 2023).

Overall, the findings of Zheng et al. show that LLM-as-a-Judge offers a scalable and reasonable evaluation paradigm that is aligned with human preferences. However, caution must be exercised due to potential biases and the real performance of reference-guided grading remains uncertain.

Groundedness

While LLM-as-a-Judge approaches introduced some of the first reference-free evaluation metrics for machine outputs, more targeted metrics that have been proposed recently build on this idea by assessing specific aspects of a response.

One such metric is Groundedness. Groundedness, also known as Faithfulness, was introduced along with Context Relevance by the authors of RAGAs in 2023 and evaluates whether the response of a RAG system is grounded in (or faithful to) the retrieved text, meaning that it contains factual information and no hallucinations (Es et al. 2023).

Similarly to Context Relevance, different methods on how to calculate this score have evolved over time. In the original version presented by Es et al., an LLM is used to extract a set of statements from a machine-generated answer. These statements are then passed to an LLM as a list together with the retrieved context, which determines for each statement whether it is supported by the information present in the context. The final score is calculated as the proportion of statements supported by the context (Es et al. 2023).

In the ARES evaluation framework, question, answer, and context are passed to the fine-tuned DeBERTa-v3-Large with a binary classifier head for a binary decision on whether or not the answer is grounded in the context (Saad-Falcon et al. 2023). This approach does not evaluate individual statements, which results in less granular scoring compared to RAGAs.

The TruLens version of the metric is similar to the RAGAs version, but includes several options and variations. As in RAGAs, the system answer is transformed into a set of statements. This split can either be performed by an LLM or by a simple sentence tokenizer. Furthermore, TruLens provides the option to filter out trivial statements from the resulting set. This filtering removes non-informative content such as pleasantries or introductory remarks by asking an LLM to go over the set of statements, extracting only

the relevant ones. Each retained statement is scored by an LLM on a scale of 0 to 3, based on how well it is supported by the context. This calculation can optionally use CoT reasoning. The final score is then given as the normalized average of the scores of all statements (tru 2025).

Additionally, the TruLens framework provides a variant called Groundedness with answerability that takes abstentions into account. This addition is important, as a key strength of RAG systems is their ability to recognize when a question cannot be confidently answered. To achieve this, an LLM first checks for each statement if it contains an abstention such as *I don't know*. If that is the case, another LLM is asked whether the question can be answered based on the context. Answerable abstentions receive a Groundedness score of 0, whereas unanswerable abstentions get a score of 1. If the statement does not contain an abstention, the regular Groundedness calculation follows (tru 2025).

The RAGAs authors evaluate their metric on WikiEval by comparing Groundedness with a baseline called GPT Score, where they simply ask ChatGPT for a Groundedness score based on context and answer. This baseline achieves an agreement of 0.72 with human annotators, while Groundedness improves this to 0.95 (Es et al. 2023). This shows that the approach is valid, although a comparison with ground-truth-based metrics like reference-guided grading could validate the results even more.

ARES evaluates Groundedness on the Attributable to Identified Sources (AIS) benchmark dataset. The portion of the dataset used contains 1,217 quadruplets of queries, contexts, answers, and a ground-truth label that indicates whether the answer can be attributed to the context. The dataset was not specifically created for RAG, raising questions about how well the results generalize to real-world RAG scenarios (Rashkin et al. 2023). The ARES Groundedness metric achieves an average overall accuracy of 0.72 (Saad-Falcon et al. 2023). The TruLens version of the metric does not come with any evaluations (tru 2025).

In summary, Groundedness is a promising reference-free metric for evaluating outputs of RAG systems that can be realized in different variants, but its performance remains insufficiently explored in comparative evaluations with other reference-free or ground-truth-based metrics.

Answer Relevance

Answer Relevance, a metric closely related to Groundedness, was introduced alongside it by the RAGAs authors in 2023. It evaluates whether the generated answer directly and appropriately addresses the user's original query (Es et al. 2023).

For Answer Relevance, three distinct calculation methods have emerged. In the original RAGAs variant, an LLM is used to generate n potential questions given the answer. The original question as well as the generated questions are then vectorized using a large embedding model and the Answer Relevance score is calculated as the average cosine similarity of all potential questions with the actual question (Es et al. 2023).

In ARES, Answer Relevance is again obtained as a binary relevance score from the fine-tuned DeBERTa-v3-Large given question, context, and answer (Saad-Falcon et al. 2023).

This approach therefore considers not only the relevance of the answer to the question, but also takes the context into account. This makes the ARES version conceptually closer to Groundedness and LLM judgement, as it incorporates both question and context when evaluating the response.

Again, the TruLens variant leaves more room for configuration. In it, an LLM is prompted to score Answer Relevance given question and answer on a scale of 0 to 3. Optionally, CoT reasoning can be used in the prompts (tru 2025).

The RAGAs version is evaluated similarly to Groundedness, i.e., by comparing its agreement with human annotators to that of their GPT Score baseline, which asks ChatGPT directly for an Answer Relevance score. Answer Relevance achieves an agreement of 0.78, which is an improvement over the baseline's 0.52 (Es et al. 2023). These results indicate that Answer Relevance achieves lower agreement with human judgements compared to Groundedness.

In ARES, the evaluation of Answer Relevance follows the same methodology as for Context Relevance: The authors compare their own Answer Relevance scores against those produced by RAGAs, using the same synthetic dataset of mock RAG systems. They report that ARES achieves a Kendall's tau that is 0.132 higher on average than RAGAs, indicating improved ranking accuracy. However, a key caveat is that the RAGAs results are based on GPT-3.5, which may limit their strength as a baseline (Saad-Falcon et al. 2023). Given that the performance margin between ARES and RAGAs is relatively small, it is plausible that newer, more capable LLMs could yield better RAGAs results, potentially closing or even reversing the gap. As before, the TruLens version of the metric does not come with any evaluations (tru 2025).

As with Groundedness, Answer Relevance presents a promising approach for reference-free evaluation in RAG systems. However, its comparative effectiveness, particularly against reference-based metrics, remains underexplored.

RAG Triad

Because metrics such as Groundedness and Answer Relevance evaluate distinct aspects of a RAG system's output, it is natural to consider combining them into a unified evaluation score. This is especially relevant for deployment decisions in real-world RAG applications. For this reason, TruEra introduced the concept of the **RAG Triad** in 2023.

Figure 2.2 presents a conceptual overview of the RAG Triad, which integrates Context Relevance, Groundedness, and Answer Relevance into a single evaluation framework for RAG systems (tru 2025).

However, although the TruEra RAG Triad and its corresponding Python library TruLens have been frequently cited in RAG evaluation surveys (e.g., Yu et al. 2024), there is a notable absence of empirical research on how to implement the RAG Triad effectively and how it performs in practice.

As previously discussed, each of the component metrics offers multiple design choices. This raises the further question of how best to combine them: Should each metric be weighted equally, or should their relative importance vary, perhaps even depending on

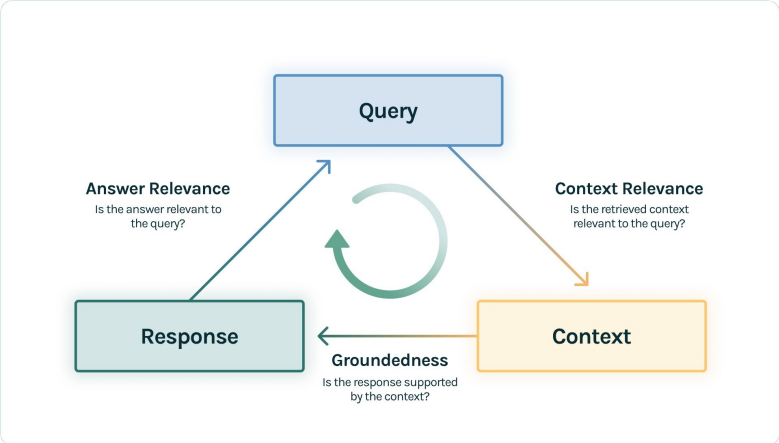

Figure 2.2.: The RAG Triad (tru 2025)

the use case?

Finally, the comparative performance of the RAG Triad remains unexplored, both in relation to other LLM-based metrics and to ground-truth-based evaluation methods. These represent key open questions in the current RAG evaluation literature.

Summary

Table 2.3 gives a summary of the discussed generation metrics. We present an overview of the metrics' type, the size and content of the used evaluation datasets, and the resulting alignment with human judgement. It is important to note that the reported evaluation scores are not directly comparable due to variations in datasets, evaluation protocols, and the human annotators involved.

However, it is visible that especially ROUGE and BERTScore use vastly larger datasets for their evaluations compared to the newer LLM-based metrics, which may indicate a higher degree of empirical validation and reliability in practical applications. Notably, certain LLM-based metrics, such as the RAG Triad, have not yet been rigorously evaluated at all, underscoring the need for further empirical validation.

To address these discrepancies, this work evaluates all metrics under a unified experimental setting to assess their relative performance and alignment with human judgement.

2.3. sovanta AG Document Chat

In order to create this unified experimental setting, this study leverages real user interaction data from sovanta AG's Document Chat application. sovanta AG is a software company and SAP Partner headquartered in Heidelberg, Germany and Document Chat is one of their commercial products that realizes RAG on SAP's platforms. Document Chat is sold to sovanta's customers, but is also used internally by its around 250 employees, generating real-world usage data suitable for empirical evaluation in this study.

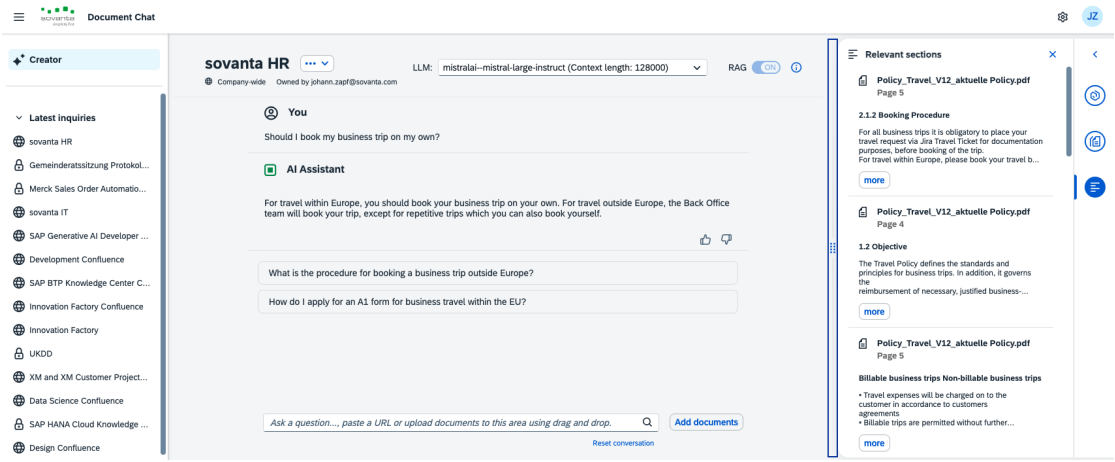

Figure 2.3.: The user interface of Document Chat

Figure 2.3 shows the user interface of Document Chat, which includes features for uploading documents or web pages, asking questions, and managing external data sources. To understand the data created by the system, it is necessary to examine the underlying RAG pipeline and its configuration options.

The RAG Pipeline of Document Chat builds on LlamaIndex, a Python framework for creating RAG Pipelines and agentic workflows. LlamaIndex provides native support for document chunking, embedding, vector storage, and retrieval of the `top_k` most relevant chunks for a given query. To influence these processes, the framework allows the configuration of the chunk size and `top_k` values and the definition of custom LLM and embedding model providers (lla 2025).

To this end, Document Chat uses LLMs available on SAP's Generative AI Hub. These currently include GPT models by OpenAI (e.g. GPT-4o), Google's Gemini models (e.g. Gemini-2.0-Flash), the Anthropic-Claude-3 family, and the open-source models Mistral-Large, Mistral-Small, and Llama3.1 (gen 2025). This means that all of these models are also available in Document Chat. However, open-source models are almost exclusively used in production deployments to comply with internal data protection and privacy policies.

For the embedding models, Document Chat uses locally run open-source models based on ONNX. ONNX (Open Neural Network Exchange) is an open platform and runtime for machine learning models that does not require libraries such as PyTorch or TensorFlow (onn 2019). Document Chat therefore supports virtually all embedding models available on HuggingFace.

Furthermore, Document Chat also has support for rerankers. To achieve the functionality described in Section 2.1, it integrates ONNX-based reranking models as a post-processor into LlamaIndex to optimize the ranking of the retrieved text passages. This functionality can be turned on or off and also allows for the configuration of various reranking models, just like for LLMs and embedding models.

If reranking is turned on, the initial retrieval step is always configured to return $k = 30$ chunks. These 30 chunks are then passed to the rerank model, which adjusts the ranking and returns the `top_k` chunks.

To keep track of data, the application uses a PostgreSQL database. This database is equipped with the pgvector extension, allowing it to store the embeddings of the documents. In Document Chat, documents are organized into self-contained contexts that are created by the users and are characterized by the documents that are uploaded into these contexts. Questions asked by the users always relate to exactly one context and the retrieval step therefore only considers the documents present in that context. Besides user-created private contexts, Document Chat also supports public contexts that can be created by administrators and are visible to everyone in the organization (though everyone has their private chat history), for example for HR documents. All contexts are identified by a context ID that is supplied with each prompt, executing the RAG pipeline with this prompt on the documents belonging to that context.

As mentioned before, Document Chat supports integrations with various external data sources. These data sources — examples include Confluence and SharePoint — can be connected to individual contexts, leading to the contents of the external system being regularly synchronized into the vector database. This means that these contexts can be used to ask questions regarding entire Confluence spaces or SharePoint sites, leading to potentially tens of thousands of candidate vectors for retrieval.

Table 2.4 provides a simplified representation of how chat messages and their associated metadata are stored in the database. For every prompt, information on the LLM used, tokens, timing, the underlying context and therefore the documents, and the results of the retrieval step are stored. Furthermore, specialized messages keep track of when the chat history was cleared and when documents were added or deleted from the context.

In summary, Document Chat is a production-grade RAG app that allows 250 employees to ask questions regarding web pages, personal documents, company-wide documentation, and external data sources. It keeps track of the asked queries, responses, and the retrieved text passages. The availability of this data offers a unique opportunity to evaluate a real-world RAG pipeline, in contrast to the synthetic benchmarks commonly used in related work.

Table 2.3.: Summary of generation evaluation metrics

Metric	**Type**	**Evaluation Dataset**	**Alignment with Human Judgement**
BLEU (Papineni et al. 2002)	N-gram-based ground truth	500 sentences	corr=0.99
ROUGE (Lin 2004)	N-gram-based ground truth	18,942 summaries	up to corr=0.99
BERTScore (Zhang et al. 2020)	Embedding-based ground truth	2,998 sentences & 80,000 images	up to corr=0.99 for translation, up to corr=0.92 for captioning
Reference-guided LLM judgement (Zheng et al. 2023)	LLM-based ground truth	-	unclear
Reference-free LLM judgement (Zheng et al. 2023)	LLM-based reference-free	6,000 pair-wise votes	up to agreement=0.85
Groundedness (Es et al. 2023; Saad-Falcon et al. 2023)	LLM-based reference-free	RAGAs: 50 answers (WikiEval), ARES: 1,217 answers	RAGAs: agreement=0.95, ARES: accuracy=0.72
Answer Relevance (Es et al. 2023)	LLM/embedding-based reference-free	50 system answers (WikiEval)	agreement=0.78
Answer Relevance (Saad-Falcon et al. 2023)	LLM-based reference-free	1,217 system answers	accuracy=0.88
RAG Triad (tru 2025)	LLM-based reference-free	-	unclear

Table 2.4.: Database entries representing user query in Document Chat

Type	**Message**	**UserID**	**ContextID**	**Time**	**Tokens**	**LLM**	**Retrieval result**
User LLM	Prompt Answer	userID	contextID	- response time	prompt completion	LLM used	- chunks retrieved

3. Experimental Setup

3.1. Data Extraction

3.1.1. sovanta Dataset

The first step in our evaluation is the extraction and preparation of the datasets. As previously discussed, data from the production RAG system Document Chat is used. This means that the starting point is the data from the chat message table described in Table 2.4.

Although data collection began in April 2024, we restrict our evaluation to interactions from January to April 2025. This timeframe was chosen to ensure consistency, as significant system changes were implemented in December 2024, rendering earlier data unsuitable for direct comparison.

In addition, follow-up questions are excluded from the dataset. Since Document Chat is a chatbot-like application, users often ask follow-up questions for clarification or additional details. To improve the quality of these interactions, Document Chat employs a technique known as condensed questions, where an LLM transforms the preceding chat history and the follow-up question into a standalone query which is then used for retrieval. While this approach improves retrieval quality by incorporating conversational context, it also makes these follow-up prompts unsuitable for standalone evaluation and they are therefore excluded.

As a result, we only consider prompts that were submitted after the chat history had been reset. Although this measure reduces the size of the dataset considerably, it leads to a cleaner evaluation dataset with more predictable outcomes. The resulting dataset consists of 111 standalone prompts, each paired with the corresponding system-generated answer and the associated source documents used to generate the response.

To use this dataset for ground-truth-based evaluations, gold answers and ground-truth retrieval results needed to be constructed. To do this, each provided answer was manually reviewed and corrected where necessary. Additionally, relevant text passages required to answer each question were identified and compiled into a new *relevant_text* column.

If the question is not answerable based on the context, we set *The context does not contain information on...* as the gold answer and an empty list as *relevant_text*. Out of the 111 prompts in the new dataset, 24 were determined to be unanswerable. We argue that this is a critical component, as one of the key promises of RAG systems is to reduce hallucinations by grounding answers in retrieved context. Including such cases allows evaluating the system's ability to abstain from answering when appropriate. Consequently, these questions are excluded from all retrieval evaluations and are only

considered in generation evaluations.

3.1.2. WikiEval Dataset

To enable comparisons with existing benchmarks, we also include the aforementioned publicly available WikiEval dataset from RAGAs in our evaluation. This dataset consists of 50 question-answer-context triplets derived from Wikipedia articles (wik 2023).

During initial tests with the dataset, it became clear that the context information given in the dataset often contains additional, unnecessary information not necessary to answer the actual question. This presents a problem, because if these additional passages are not retrieved, recall may indicate a lower value than desired. For this reason, we manually removed unnecessary information from the *context* column in the dataset.

This problem with the dataset does not affect the evaluations conducted by the authors of RAGAs, because they only use Context Relevance as a metric, but it certainly affects all recall-based metrics calculated on the dataset.

3.2. Preprocessing

3.2.1. Data Validation

The first step in preprocessing involves validating the quality and consistency of the datasets. This step ensures that all gold answers are fully supported by the listed relevant text passages and that these passages can be located within the original source documents. Reliable validation is essential for ensuring the integrity of downstream evaluation metrics.

For this purpose, we developed a Python script that iterates over the rows in the datasets and checks for each text passage in the *relevant_text* column whether it is actually contained in the underlying source documents. In addition to the automated check, we manually reviewed each gold answer to ensure that the provided response was adequately supported by the relevant passages.

Additionally, we set up a basic RAG Pipeline with a large context size (chunk_size=512, `top_k`=12) and executed each question in our datasets on it. For the WikiEval dataset, this resulted in an average retrieval recall of 0.997, confirming the consistency of relevant information within the dataset. On the sovanta dataset, the average recall on the validation set is lower at 0.859. Given that all relevant passages were verified to exist in the source documents, this result suggests that the sovanta dataset presents greater retrieval challenges — likely due to its broader document base and potentially more diverse phrasing. This highlights the increased complexity of the sovanta dataset for RAG systems.

3.2.2. Clustering

To analyze the performance of our RAG system on different use cases and to identify performance differences of different metrics across domains, we clustered the prompts in

the sovanta dataset into six categories:

- **HR**: Questions related to human resources and company policies
- **TECH**: Questions regarding technology and data science
- **SUMMARY**: Summaries, long texts, and creative writing
- **CONTRACTS**: Questions regarding contracts and projects
- **UNKNOWN**: The 24 unanswerable questions
- **OTHER**: All other questions

These clusters were manually identified from the dataset. The clustering of the prompts was then performed by prompting Llama3.1 to output a cluster for each row given prompt and gold answer. The used prompt can be found in the appendix.

The resulting cluster distribution is as follows: 30 prompts belong to HR, 24 are UNKNOWN, 18 belong to TECH, 16 to SUMMARY, 12 to OTHER, and 11 to CONTRACTS. We manually reviewed all the cluster labels generated by Llama3.1 to ensure label accuracy.

3.2.3. Final Datasets

After data extraction, validation, and clustering, we are left with our final datasets. As mentioned before, the sovanta dataset has 111 questions, of which 75 are in German and 36 in English. An excerpt is presented in Table 3.1. Each row in the dataset includes the assigned cluster label, the prompt, the gold answer, the relevant text passages, and metadata about the source materials used to generate the system's response.

As discussed in Section 2.3, the Document Chat RAG pipeline retrieves information from two types of sources: individual files uploaded by users and content synchronized from external knowledge systems. In the first case, the *sources* column lists file names (e.g. PDF or Word documents). In the second case, it indicates the external system involved, such as *SHAREPOINT* (referring to sovanta's internal SharePoint instance) or *CONFLUENCE-SDS* (referring to the Data Science Confluence space).

Of the 111 prompts, 31 reference six distinct Confluence spaces, 28 reference SharePoint documents, and 19 draw from the *Innovation Factory*, sovanta's internal project catalog. The remaining 33 prompts reference .pdf, .docx, or .pptx files. As such, the dataset spans a diverse range of source types, from small documents to large external knowledge bases comprising thousands of pages. Alongside the dataset of prompts, we store all files and the full contents of the referenced external systems at a specific point in time to ensure reproducibility.

Table 3.2 shows an example from the final WikiEval dataset. Like the sovanta dataset, it contains the prompt, gold answer, and relevant text passages. However, in contrast, WikiEval also provides a single column named *context*. This column is taken directly

Table 3.1.: Excerpt from the sovanta dataset

Cluster	Prompt	Gold Answer	Relevant Text	Sources
HR	I lost my key card for our office, what should I do?	You should report the loss of your card promptly to the Back Office by sending an email to backoffice@sovanta.com.	['If you lose your card, you must report this promptly to the Back Office by sending an e-mail to backoffice@sovanta.com.']	['SHAREPOINT']
CONTRACTS	Wie lange läuft der Vertrag für Brandmeldeanlagen?	Der Vertrag für Brandmeldeanlagen läuft für 4 Jahre.	['Die Vertragsdauer beträgt: 4 Jahre']	['Wartungsvertrag Brandmeldeanlagen.pdf']

from the original dataset and contains the information from the original Wikipedia page needed to answer the question as well as some extra information that is not necessary.

It is important to note that while the *context* column in WikiEval contains the relevant background information, it does not include the full Wikipedia page. Moreover, due to the passage of time and the dynamic nature of Wikipedia, many of the original pages have changed, making retrieval of the exact original content no longer feasible. Using the full original Wikipedia pages instead of the excerpts would have made for a more realistic retrieval task, which is a challenge we address in the next section.

3.3. RAG Pipeline Execution

To evaluate the Document Chat RAG pipeline on the two datasets, it is necessary to execute the ingestion, retrieval, and generation steps with different configurations of the hyperparameters explored in Section 2.1. This results in a final dataset with retrieved passages and LLM answers for the prompts in the datasets on different configurations, enabling evaluations across these configurations.

All experiments are run locally on an Apple M2 Pro MacBook Pro with 32 gigabytes of RAM. We use Python 3.12 and a PostgreSQL database with pgvector running in a local Docker container as the vector database. As described before, all embeddings are computed locally with ONNX and LLM capabilities are provided by SAP's Generative AI Hub.

Table 3.2.: Excerpt from the WikiEval dataset

Prompt	Gold Answer	Relevant Text	Context
When is the scheduled launch date and time for the PSLV-C56 mission, and where will it be launched from?	The PSLV-C56 mission is scheduled to be launched on Sunday, 30 July 2023 at 06:30 IST / 01:00 UTC. It will be launched from the Satish Dhawan Space Centre, Sriharikota, Andhra Pradesh, India.	['The PSLV-C56 is the 58th mission of Indian Space Research Organisation's Polar Satellite Launch Vehicle (PSLV) and the 17th flight of the PSLV-CA variant, and will be launched from Satish Dhawan Space Centre First Launch Pad (FLP).', 'It is Scheduled to get launched on Sunday, 30 July 2023 at 06:30 IST / 01:00 UTC from Satish Dhawan Space Centre, Sriharikota, Andhra Pradesh, India.']	*Content of the PSLV-C56 Wikipedia page*

3.3.1. Ingestion

During ingestion, the source documents are chunked, embedded, and stored in the vector database. The two hyperparameters that affect this stage are the embedding model and the chunk size.

For the embedding models, we use BGE-M3 and Multilingual-E5-Large, two state-of-the-art multilingual embedding models with 1,024 embedding dimensions (Chen et al. 2024; Wang et al. 2024). In the Massive Text Embedding Benchmark (MTEB), both are among the highest performing open-source models, with scores of 59.56 and 58.55, respectively (Enevoldsen et al. 2025). For the chunk size, we use 64, 128, 256, and 512 tokens. Larger chunk sizes are not possible because Multilingual-E5-Large has a maximum input size of 512 tokens (Wang et al. 2024).

For each dataset and each combination of embedding model and chunk size, a separate vector table is created in the vector database, yielding eight tables per dataset.

We start with the sovanta dataset, where the following steps are repeated over the eight hyperparameter combinations: Initially, a context is created for each external system and the data is loaded, chunked, and embedded using the corresponding chunk size and embedding model. For all prompts referencing the specific external system, the ID of the context is stored in a new column named after the hyperparameter combination, for example *eval_context_id_128_BAAI/bge-m3*.

Additionally, for all prompts that reference single files, a separate context containing the respective documents is created and their context IDs are stored accordingly. This allows referencing the correct hyperparameter configuration later when executing the retrieval and generation pipelines.

A similar process is applied to the WikiEval dataset. As discussed before, the dataset suffers from limited retrievable context per query, leading to a potentially trivial retrieval task. For this reason, we aggregate the context from all queries in the dataset and embed

Table 3.3.: Impact of chunk size on the number of vectors in each dataset

Chunk Size	# sovanta Chunks/Vectors	# WikiEval Chunks/Vectors
64	520,190	1,437
128	264,507	468
256	77,850	196
512	42,665	91

it in a single context for each hyperparameter combination. The IDs of the eight resulting contexts are stored in the same column as in the sovanta dataset — although here the ID is the same for each prompt per hyperparameter combination.

Table 3.3 shows how chunk size impacts the number of vectors generated for both datasets. The sovanta dataset has a considerably higher amount of chunks, resulting in a significantly more challenging retrieval process.

This means that this new dataset, which is based on real-world enterprise data, already represents a meaningful contribution of this work, as it undoubtedly extends beyond the comparatively simpler WikiEval dataset that was used to validate many of the LLM-based metrics discussed in Section 2.2.2.

These indexed embeddings serve as the foundation for the retrieval and generation stages, discussed next.

3.3.2. Retrieval

Since the retrieval step in RAG does not require the use of LLMs, it is usually considerably faster and less costly than the generation phase. This is also the case in Document Chat. For this reason, we start by executing only the retrieval phase on a large hyperparameter set. The preliminary results of this evaluation can then be used to select the hyperparameter set for the generation phase, leading to increased evaluation efficiency and reduced costs.

In retrieval, the relevant hyperparameters are the embedding model, the chunk size, the number of chunks selected per query (`top_k`), and the optional rerank model. For embedding model and chunk size, we use the same hyperparameter set that was already used in the ingestion phase.

For `top_k`, we use the values 2, 4, 6, 8, 10, 12 and 14. These values are selected based on empirical observations from practical use and internal benchmarking during Document Chat development. For example, `top_k`=2 is the default value used by LlamaIndex (lla 2025), though we observed that using only two chunks is frequently insufficient when working with larger datasets and complex queries. On the other hand, we restrict values to a maximum of 14, because in combination with a chunk size of 512, this already results in up to 7,168 context tokens. These large context sizes not only negatively affect the answer generation speed, thereby impacting user experience, but together with chat history and system prompt tokens also come close to filling the context windows of certain older LLMs.

Table 3.4.: Retrieval hyperparameter grid summary

Hyperparameter	**Values**
Embedding Model	BGE-M3, Multilingual-E5-Large
Chunk Size (tokens)	64, 128, 256, 512
`top_k`	2, 4, 6, 8, 10, 12, 14 (WikiEval: up to 12)
Rerank Model	None, BGE-Reranker-V2-M3, Mixedbread-Base, Mixedbread-XSmall

For the rerank model, we use four variants: *BGE-Reranker-V2-M3*, *Mixedbread-Rerank-Base-V1*, *Mixedbread-Rerank-XSmall-V1*, and None. *BGE-Reranker-V2-M3* was already mentioned in Section 2.1 and is an open-source multilingual reranker based on the BGE-M3 embedding model with 568M parameters (Li et al. 2025). *Mixedbread-Rerank-Base-V1* and *Mixedbread-Rerank-XSmall-V1* are two smaller open-source rerank models with 184M and 70M parameters, respectively. Both models are trained exclusively on English text and may not generalize well to multilingual input, unlike *BGE-Reranker-V2-M3* (Shakir et al. 2024).

Larger rerank models are not used, as even the 568M parameter model already increases retrieval time by an average of 12 seconds on our data and hardware. Larger models would exacerbate this latency, further diminishing usability. Detailed timing results are presented in Section 4.1.5.

In total, the two embedding models, four chunk sizes, seven `top_k` values, and four rerank models lead to 224 unique hyperparameter combinations. This grid size represents a balance between thoroughness and computational feasibility on the available hardware. As discussed before, the retrieval evaluation on the sovanta dataset considers the 87 answerable questions, leading to a total of **19,488 evaluated data points for sovanta retrieval**.

For WikiEval, we exclude `top_k`=14 from the evaluation, because the initial evaluations in Section 3.2.1 already indicate that `top_k`=12 is absolutely sufficient to achieve maximum recall. The remaining 192 hyperparameter combinations together with the 50 questions in the dataset therefore result in **9,600 data points** for retrieval.

Technically, the retrieval-only RAG pipeline is realized by supplying LlamaIndex with a *MockLLM* class that always outputs an empty string. A grid search is then performed on the hyperparameter space, and the RAG pipeline is executed on each row in the dataset with the respective parameters and against the context of which the ID is stored in the previously explained column.

In the retrieval-only case, the RAG pipeline returns the chunks found and the timings for query embedding and retrieval. For each hyperparameter combination and each prompt, the RAG pipeline output and the hyperparameter combination as well as the prompt, relevant text, and cluster from the original dataset are stored as a row in a new dataset. We call this dataset the prediction dataset.

After running the prediction, an additional script reruns any failed predictions, making sure that there are no errors in the prediction dataset. The final prediction dataset then serves as the basis for all retrieval evaluations. The following is an example row from the sovanta dataset:

embed_model: BAAI/bge-m3; **chunk_size**: 64; **top_k**: 2; **rerank_model**: hooman650/bge-reranker-v2-m3-onnx-o4; **is_error**: False;
prompt: *I lost my key card for our office, what should I do?*;
runtime: 5,110.139 ms;
retrieval_time: 5,080.568 ms; **embedding_time**: 27.153 ms;
nodes: *[PNode(nodeID='6ae02157-f6f5-4de8-b38e-6855c866b018', documentID=43517, documentName='Policy-Access Card Handling-en1.pdf', text='If you lose your card, you must report this promptly to the Back Office by sending an e-mail to', score=-0.3450, source='SHAREPOINT'), PNode(nodeID='d2a26626-646d-4705-8b7a-42be4b0fa322', documentID=43517, documentName='Policy-Access Card Handling-en1.pdf', text='you must report this promptly to the Back Office by sending an e-mail to backoffice@sovanta.com.', score=-1.7069, source='SHAREPOINT')]*;
relevant_text: *['If you lose your card, you must report this promptly to the Back Office by sending an e-mail to backoffice@sovanta.com.']*

3.3.3. Generation

As previously outlined, the preliminary retrieval evaluation results inform the selection of the hyperparameter space for the generation phase. The full results are presented in Section 4.1. The results indicate that BGE-M3 outperforms Multilingual-E5-Large, with the performance gap being particularly pronounced on the sovanta dataset. Given its consistently superior performance, BGE-M3 is selected as the sole embedding model for the generation experiments.

As anticipated, retrieval recall improves with larger context sizes, defined as the product of chunk size and `top_k`. This increase is more drastic on the sovanta dataset. Notably, configurations with `top_k`=2 and chunk sizes of 64 tokens yield substantially lower recall and are therefore excluded from subsequent generation evaluations.

In contrast, the performance of the reranking models shows more variability across datasets. On both datasets, using *BGE-Reranker-V2-M3* leads to the highest average recall. On the sovanta dataset, using no rerank model is the second best option, closely followed by *Mixedbread-Rerank-Base-V1*. *Mixedbread-Rerank-XSmall-V1* performs significantly worse than all other configurations. On the WikiEval dataset, these three reranking configurations exhibit comparable performance, with *Mixedbread-Rerank-Base-V1* showing a slight advantage over both its smaller variant and the baseline with no reranker.

As described before, rerank models always incur a performance penalty and make retrieval pipelines more complex. Therefore, the option of omitting reranking is retained within the hyperparameter space. This allows assessing whether rerankers influence not only retrieval recall but also the quality of the final generated answers. As the two Mixedbread rerankers do not demonstrate significant gains over using no reranking, they are excluded from the generation experiments.

In addition to the aforementioned hyperparameters, the choice of LLM plays a crucial role in the generation phase, as it substantially affects the system's ability to produce

Table 3.5.: Generation hyperparameter grid summary

Hyperparameter	**sovanta Dataset**	**WikiEval Dataset**
Embedding Model	BGE-M3	
Chunk Size (tokens)	128, 256, 512	
`top_k`	4, 6, 8, 10, 12, 14	4, 6, 8, 10, 12
Rerank Model	None, BGE-Reranker-V2-M3	
LLM	Llama3.1, Mistral-Large	Llama3.1, Mistral-Large, GPT-4o, Gemini-2.0-Flash

coherent and contextually appropriate responses. The choice of LLMs varies depending on the dataset.

Because the sovanta dataset contains sensitive information that cannot be shared with external providers such as Microsoft, OpenAI, or Google, the selection of LLMs is restricted to the two open-source models *Llama3.1-70b-Instruct* and *Mistral-Large-Instruct-2407*, both of which are hosted within SAP's data centers.

For the WikiEval dataset, *GPT-4o-2024-11-20* and *Gemini-2.0-Flash-001* are evaluated alongside the open-source models. The Llama3.1 version used in this evaluation has 70 billion parameters (Dubey et al. 2024), while Mistral-Large has 123 billion parameters (mis 2024). The parameter sizes of GPT-4o and Gemini-2 remain undisclosed, though it has been estimated that GPT-4o has more than 200 billion parameters (Abacha et al. 2024). This selection therefore enables comparative evaluation across a diverse set of LLMs with varying model scales and capabilities.

Table 3.5 shows the final hyperparameter grid for both datasets. It results in 72 combinations for the sovanta dataset. With the 111 prompts in the sovanta dataset, **7,992 data points** are evaluated in total. For the WikiEval dataset, 120 combinations across 50 prompts yield **6,000 data points** in total.

The execution of the full RAG pipeline for generation evaluation builds upon the previously described retrieval-only setup. The primary difference is the replacement of the *MockLLM* class with a class that performs actual inference using the LLM specified in the hyperparameter grid. The resulting prediction dataset contains all columns that are also stored for retrieval evaluation, but also includes additional fields such as the selected LLM, the gold answer, the runtime of the generation step, the used input and completion tokens, and the final LLM-generated answer, which is also the output of the RAG pipeline.

Similarly to the retrieval prediction, an additional script was used to rerun any failed predictions to ensure that there are no errors in the dataset. This step is more relevant here because in our experience, the generation component of a RAG pipeline is the most susceptible to errors, for example due to connection problems, rate limits, and prompt filters. The following is an example row from the sovanta dataset:

> **embed_model:** BAAI/bge-m3; **LLM:** Mistral-Large-Instruct; **chunk_size:** 128; **top_k:** 4; **rerank_model:** None; **is_error:** False;
> **prompt:** *I lost my key card for our office, what should I do?*;
> **runtime:** 1,287.681 ms; **retrieval_time:** 297.29 ms;

embedding_time: 43.562 ms; **generation_time:** 946.832 ms;
input_tokens: 847; **completion_tokens:** 38;
answer: *If you lose your key card for the office, you must report this promptly to the Back Office by sending an e-mail to backoffice@sovanta.com.*;
nodes: *[PNode(...), [PNode(...), [PNode(...), [PNode(...)]*;
gold_answer: *You should report the loss of your card promptly to the Back Office by sending an email to backoffice@sovanta.com.*;
relevant_text: *[If you lose your card, you must report this promptly to the Back Office by sending an e-mail to backoffice@sovanta.com.]*

3.4. User Feedback Collection

As outlined in Section 2.2.2, human preference evaluations remain the gold standard for assessing LLM-based systems such as RAG. For this reason, they are also incorporated in this evaluation. Both static and live preference collection methods are used: The former is based on the previously described datasets, while the latter is introduced in this section.

Having access to the development and deployment environment of Document Chat enables us to extend the system with a live human feedback mechanism inspired by LMArena. The idea is to periodically present users with two alternative answers to their query, each generated using a different RAG configuration. Users are then asked to select the response they deem better. These preferences can then be used to directly compare system variants.

Another advantage of this approach is that follow-up questions that had to be excluded from the other evaluations are also covered here, because feedback can be collected on any user prompt.

A key challenge in implementing such a system lies in selecting which hyperparameters to vary. As mentioned before, platforms like LMArena solely vary the LLM used for response generation and the approach has not been applied to RAG yet. However, RAG systems have a much larger hyperparameter space.

Since this feedback mechanism is integrated into a production system, inference latency is a primary concern, as response times must remain within acceptable limits. For this reason, only `top_k` and the LLM used are selected as tunable hyperparameters. Both of these parameters only influence the inference stage (when a user sends a prompt) and in LlamaIndex, both are passed to the framework at inference time anyway, so they can be easily changed (lla 2025).

For various reasons, the other hyperparameters used in this work are not suitable for a production live feedback approach. Varying embedding model and chunk size would require storing differently chunked and embedded vectors of each document, which would not only multiply storage space requirements, but also significantly increase the upload time for new documents. Theoretically, the relevant documents could also be reembedded and rechunked with the respective varying hyperparameters at inference

time, but this would inevitably dramatically increase the response time, especially for larger documents. The effect on user experience was therefore deemed too high to make varying these hyperparameters feasible.

Similar reasoning holds for reranking. Although reranking is performed at inference time and can therefore easily be modified when necessary, most reranking models introduce a significant performance penalty on the available hardware. As detailed in Section 4.1.5, the three reranking models tested incur average delays of 12, 5, and 2 seconds, respectively. On the other hand, it was already established that only the largest model (which is also the slowest) offers a clear benefit in retrieval quality. However, its latency was judged unacceptable for a production environment. Therefore, reranking is also excluded from live feedback evaluation.

We subsequently define a parameter grid over the two tunable hyperparameters. For the LLM, the same external provider restrictions also apply here, so the open-source LLMs *Mistral-Large-Instruct-2407*, *Mistral-Small-Instruct-2503* with 24 billion parameters (mis 2025), and *Llama3.1-70b-Instruct* are used. For `top_k`, we reuse the six values previously evaluated on the sovanta dataset (4 through 14), resulting in a total of 18 unique combinations.

This reduced search space has the added benefit of requiring fewer data points to yield statistically meaningful comparisons, as user feedback is concentrated on fewer configurations.

During feedback collection, all other RAG hyperparameters in the production system are held constant: The chunk size is fixed at 256, the embedding model is BGE-M3, and no reranking is applied.

The following part describes the implementation of the feature in Document Chat, which we refer to as side-by-side feedback. Every time any user submits a query, side-by-side feedback is triggered with a certain probability. This probability is configurable and defaults to 10%. If feedback is to be collected, the next step is the sampling of the two hyperparameter combinations used for feedback.

For the sampling process, we initially considered using Bayesian sampling or a dueling bandits approach to guide the optimization process. This way, confident results could be achieved faster by focusing the evaluation on areas of the hyperparameter grid with high uncertainty. However, there is a surprising lack of suitable Python libraries that would allow the application of such algorithms to our case, where sampling and reward updates occur in different execution contexts and at varying points in time.

For this reason, we apply uniform sampling to our hyperparameters. Although LMArena uses a custom non-uniform sampling rule, its authors show that random sampling still yields competitive outcomes given sufficient data (Chiang et al. 2024). Since our search space is rather limited, we expect uniform sampling to be effective in practice.

In detail, we sample two combinations of LLM and `top_k` such that the two values for LLM or `top_k` may be the same, but the overall combinations are distinct. These combinations are subsequently used to simultaneously execute the two respective versions of the RAG pipeline. Results are passed to the frontend and displayed to the user immediately upon completion, meaning that users are able to see which response is generated faster. This introduces a potential bias in user choices, as generation latency

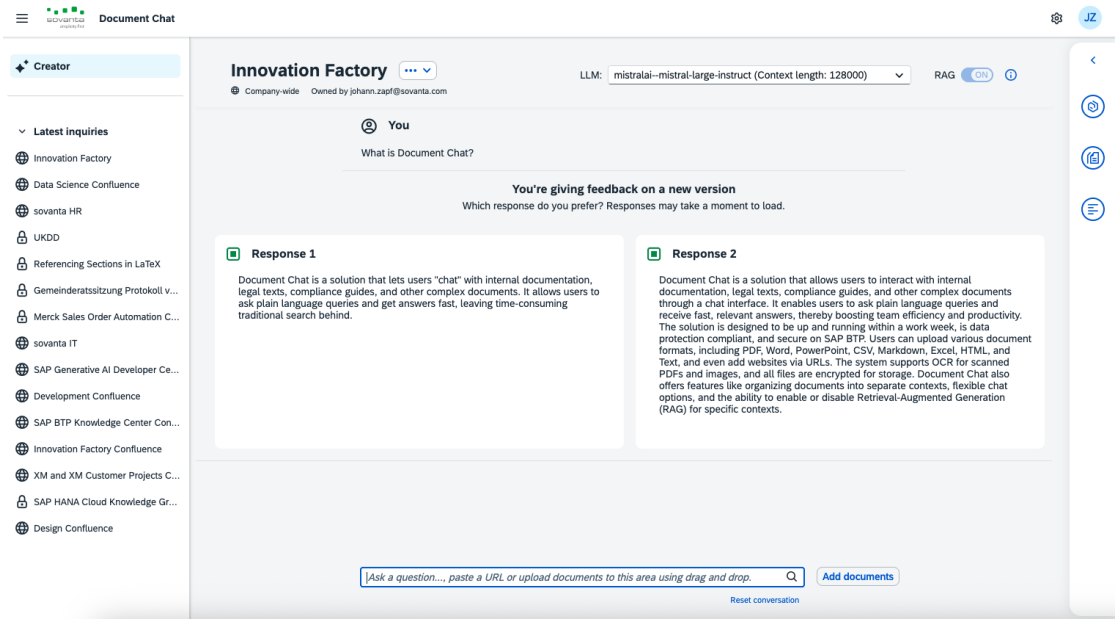

Figure 3.1.: Side-by-side feedback in Document Chat. In this example, the left answer is generated by *Llama3.1-70b-Instruct* with `top_k`=4 and the right answer is generated by *Mistral-Small-Instruct-2503* with `top_k`=12.

differs across models and increases slightly with larger `top_k` values (see Section 4.2.8). Users may therefore inadvertently prefer faster responses.

After both answers have been generated, the user is able to select the answer that is deemed better by clicking on it. Upon doing so, the interface reverts to the standard single-answer view. Furthermore, the two choices, their underlying hyperparameter combinations, as well as the choice made by the user are stored as a new row in a special feedback database table.

An example of side-by-side feedback in Document Chat is shown in Figure 3.1. In this case, the answer on the right is clearly better, because it elaborates more on the question while still providing only correct information. In LMArena, the used models *Llama3.1-70b-Instruct* and *Mistral-Small-Instruct-2503* currently have scores of 1,294 and 1,349, respectively (lma 2025). Therefore, the stronger LLM and access to the triple amount of context likely lead to the better and more elaborate answer.

Side-by-side feedback in Document Chat was rolled out to the production system as part of release 1.2 on March 21st, 2025. The feature has been continuously active and collecting data since. Its results are presented in Section 4.2.9.

4. Results

In the following sections, we present the results of our evaluations, starting with retrieval and then proceeding to generation. Both sections follow a similar structure: We first present the selected ground-truth baseline metric and its results. Then, we describe which metrics are selected for further evaluations and how they perform against each other and with different variants. We then apply these metrics to both datasets and evaluate how they correlate with the ground-truth baseline. Subsequently, we employ an LLM-based approach to analyze the errors incurred by the metrics. Finally, we report auxiliary evaluation metrics, including processing time and token consumption.

To perform these evaluations, we integrate the TruLens evaluation library into Document Chat to make it compatible with the application's outputs. We choose TruLens because it supports most of the metrics required for our analysis and, unlike frameworks such as RAGAs, offers extensive configurability, allowing for further optimization (tru 2025; Es et al. 2023).

For the computation of LLM-based metrics, the same external LLM provider restrictions mentioned before also hold here. Consequently, we only use *Mistral-Large-Instruct-2407* and *Llama3.1-70b-Instruct* to compute metrics on the sovanta dataset. On WikiEval, we additionally use *GPT-4o-2024-11-20*, *GPT-4o-mini-2024-07-18*, and *Claude-3.7-Sonnet-2025-02-19*. This enables us to analyze the performance differences of LLM-based metrics when applied to different models.

4.1. Retrieval Evaluation

4.1.1. Baseline: Recall

The baseline retrieval evaluation method is recall, computed using the previously constructed *relevant_text* columns. As discussed in Section 2.2.1, recall is arguably the most meaningful non-rank-based ground-truth metric. We do not use any rank-based metrics here for two reasons: First, it remains unclear whether the order of retrieved chunks significantly impacts RAG system performance. Moreover, current rank-based metrics may not accurately capture this effect. Second, our comparison with LLM-based Context Relevance — also a non-rank-based metric — further justifies the choice of a rank-agnostic baseline.

For recall, the previously discussed question of how to handle results that partially contain the relevant text remains. To investigate this effect, we extract large context portions from both datasets (chunk size=512, `top_k`=12) and compute three recall variants. The first variant, standard recall, counts only exact matches between relevant and retrieved texts. The other two versions, which we call *recall_0.5* and *recall_0.3*, also

Algorithm 4.1 Calculate Recall

Require: Retrieved content *pred*, List of strings *relevant_text*, Threshold *lcs_threshold*
Ensure: Retrieval Recall

```
total ← 0
for all y ∈ relevant_text do
    if y in pred then
        total ← total + 1
    else
        node_score ← LCS_Length(pred,y) / min(len(pred),len(y))
        if node_score > lcs_threshold then
            total ← total + node_score
        end if
    end if
end for
return total
```

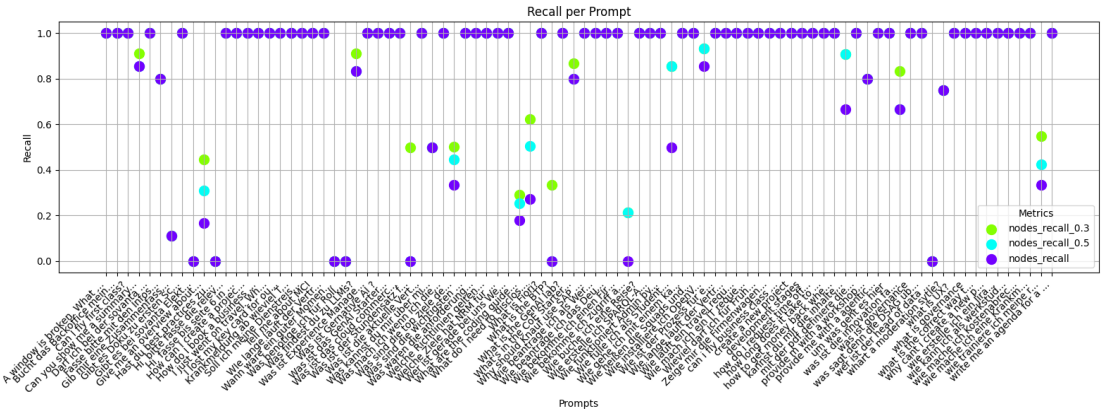

Figure 4.1.: Different flavors of recall per prompt on the sovanta dataset

consider substring matches that cover at least 50% or 30% of the relevant string length. We calculate these versions using Algorithm 4.1. Note that if *lcs_threshold* = 1, the algorithm simply returns standard recall.

Figure 4.1 presents the differences of the metrics per prompt with the used subset of the sovanta dataset. It shows that for most prompts, changing the recall calculation does not make a difference, but that the consideration of substrings increases recall in certain cases.

For example, the *relevant_text* column for a question on VPN access contains the string *Type URL "https://openvpn.sovanta.com" and click NEXT*. The retrieved text, however, only includes the URL portion, yet it still contributes meaningfully to the answer. It therefore makes sense to respect this in the final score. On the other hand, the average recall on the portion of the sovanta dataset only increases by 2.1% when

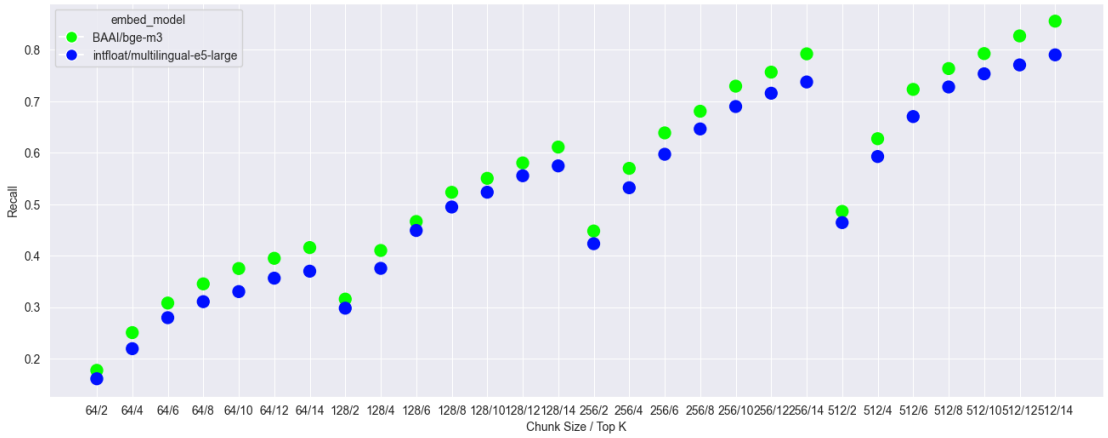

Figure 4.2.: Retrieval recall per embedding model on the sovanta dataset

using *recall_0.5*. This is important, because the approach has a risk of introducing false positives. However, the minor increase in average recall shows that using *recall_0.5* achieves a good balance between including partially relevant content and preventing false positives.

On the WikiEval dataset, the variation of the recall metric only affects 3 of 50 prompts with an increase in average recall of 0.9%. Based on these findings, we use *recall_0.5* as the metric going forward.

Results

On the sovanta dataset, BGE-M3 is the more effective embedding model with an average recall of 0.551, compared to Multilingual-E5-Large's 0.515. As seen in Figure 4.2, this advantage of BGE-M3 holds for all combinations of chunk size and `top_k`. Among the reranking models, BGE-Reranker achieves an average recall of 0.580. In comparison, not using a reranker results in 0.542 and Mixedbread-Base and Mixedbread-XSmall yield 0.540 and 0.468, respectively. Figure 4.3 illustrates that this difference is especially pronounced for larger context sizes and that the models perform relatively similarly on smaller contexts.

Both Figures also demonstrate a significant improvement in recall as the context size increases. This is expected, because larger contexts incur a higher probability of finding the relevant content and because recall does not penalize the retrieval of irrelevant information. Table 4.1 shows the three best overall combinations on the sovanta dataset. As expected, these combinations correspond to the largest context sizes.

Table 4.1 also shows that even the best hyperparameter combination only achieves an average recall of 0.89. Our analysis here shows that for eight questions from the dataset, none of the retrieval pipelines achieve a recall of 1, though all achieve at least 0.3.

Of these eight questions, five are summaries with extensive relevant text that needs to be found, and three are challenging questions where the answer is contained in a very

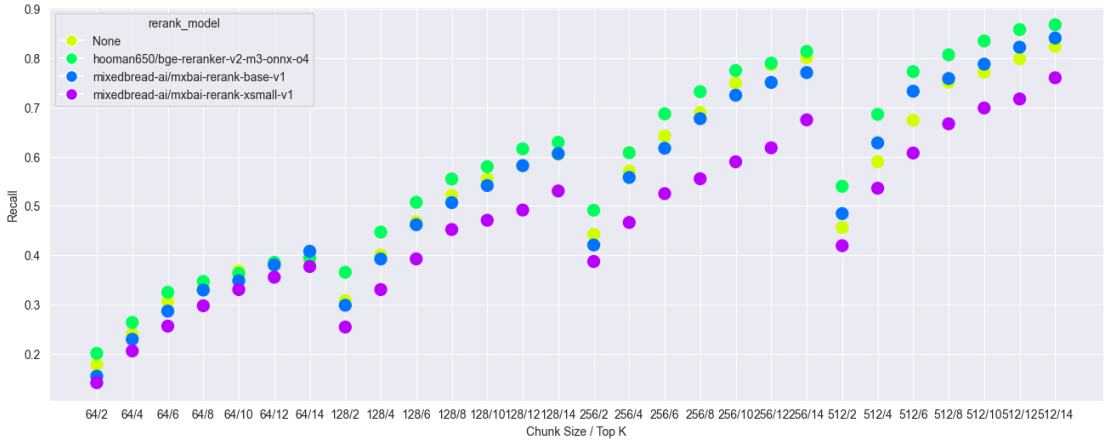

Figure 4.3.: Retrieval recall per reranking model on the sovanta dataset

Table 4.1.: The three hyperparameter combinations with the highest recall on the sovanta dataset

Embedding Model	**Rerank Model**	**Chunk Size**	**top_k**	**Recall**
BGE-M3	BGE-Reranker-V2-M3	512	14	0.889852
BGE-M3	None	512	14	0.886457
BGE-M3	BGE-Reranker-V2-M3	512	12	0.874651

large context with lots of similar content. These factors increase the difficulty for the retriever; however, as previously discussed, all relevant content is available within the documents. This characteristic may even be considered an advantage of the dataset, as it provides challenging cases that effectively test retrieval performance.

In addition to these scores, Figure 4.4 illustrates the retrieval recall scores per cluster. It shows that summaries have the lowest average recall and benefit the most from increasing context sizes, which is expected because these types of questions usually require many text passages to be retrieved.

Questions on contracts and technology generally perform best. This makes sense, because on average, they only require 2.8 and 3.6 text passages to be found, compared to the 12.2 required by summaries. However, HR questions only require 3.3 text passages to be found on average, and still perform worse than contracts and technology. These results indicate that although the number of text chunks to be retrieved has an influence on the retrieval results, the characteristics of the different question classes also have an effect.

As seen in Figure 4.5, BGE-M3 (recall=0.868) also outperforms Multilingual-E5-Large (recall=0.864) on the WikiEval dataset, though the difference is less pronounced than on the sovanta dataset. Similarly to the sovanta dataset, the BGE-Reranker is the best reranking model with an average recall of 0.880. In contrast, both Mixedbread-Base (recall=0.870) and Mixedbread-XSmall (recall=0.860) outperform the no-reranker

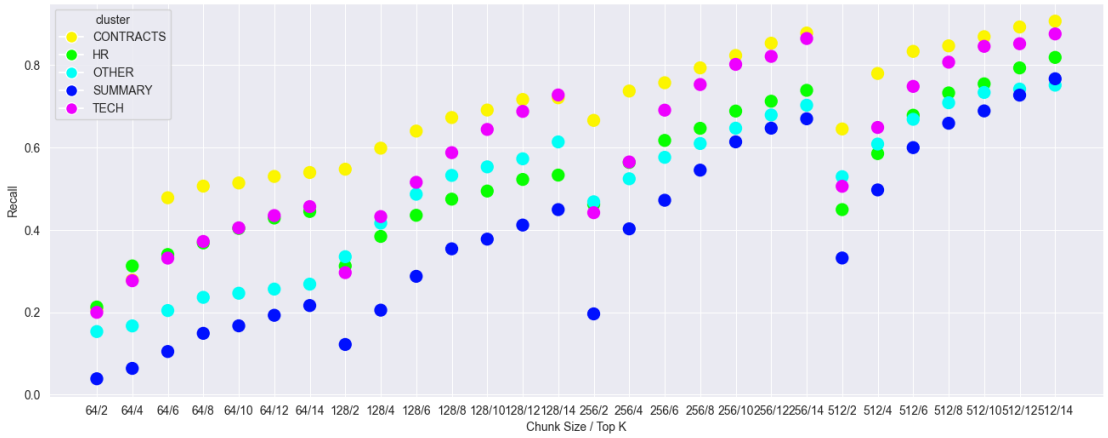

Figure 4.4.: Retrieval recall per cluster on the sovanta dataset

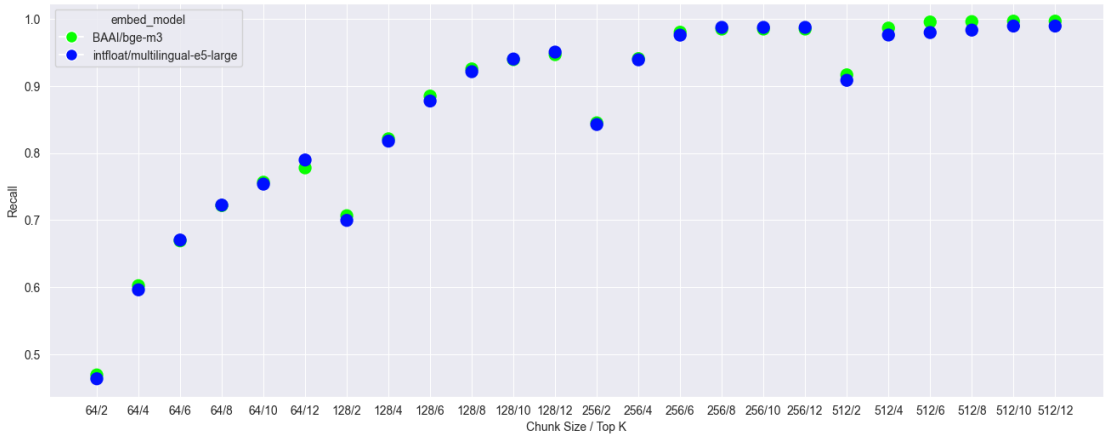

Figure 4.5.: Retrieval recall per embedding model on the WikiEval dataset

configuration (recall=0.854) on the WikiEval dataset.

These values and Figure 4.6 show that the gap between the models is significantly smaller on the WikiEval dataset. As mentioned before, the Mixedbread models were trained exclusively on English text. To evaluate whether the performance gap and the better performance of the Mixedbread models on WikiEval is due to the fact that WikiEval only contains English queries, while 68% of the sovanta dataset is in German, we evaluate the sovanta dataset on its English part only.

The result is that the relative performance of the Mixedbread-Base model does not change, but that the Mixedbread-XSmall model now achieves a recall within one percentage point of not using a reranker, compared to eight percentage points before. In this setting, the performance gap between the best and the worst model also reduces from twelve percentage points to the three percentage points that are also observed on

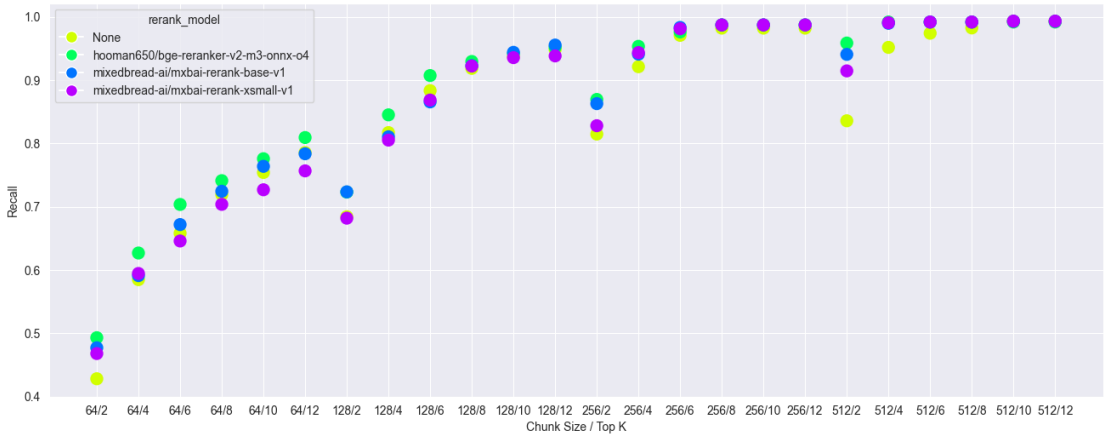

Figure 4.6.: Retrieval recall per reranking model on the WikiEval dataset

Table 4.2.: The three hyperparameter combinations with the highest recall on the WikiEval dataset

Embedding Model	**Rerank Model**	**Chunk Size**	**top_k**	**Recall**
BGE-M3	Mixedbread-Rerank-XSmall-V1	512	12	0.996803
BGE-M3	Mixedbread-Rerank-XSmall-V1	512	10	0.996803
BGE-M3	Mixedbread-Rerank-Base-V1	512	12	0.996791

the WikiEval dataset. These results suggest that the smaller reranker's performance is particularly impacted by non-English input and that the benefits of using rerankers depend on the language set used.

Similarly to the sovanta dataset, Figures 4.5 and 4.6 show that more context generally leads to better results. In contrast to the sovanta dataset, WikiEval converges to the optimal recall of 1.0, as shown in Table 4.2. Furthermore, this convergence is already achieved with moderate context sizes, as chunk size=512 and `top_k`=4 already result in a recall of 0.99. Also, 60 of 9,000 combinations achieve a recall of at least 0.98, while only 20 of 19,488 sovanta combinations exceed a recall of 0.8.

To investigate these differences, we look at the relevant text to be retrieved in both datasets. On the sovanta dataset, the average row has 3.9 relevant sentences with 99.7 characters each. For WikiEval, the average row has 4.1 relevant sentences with 167.8 characters each. This means that although the retriever needs to find, on average, almost double the content per query, the average recall on the WikiEval dataset is substantially higher.

To rule out the language differences as the cause for the lower recall on the sovanta dataset, we look at the average recall of the English prompts. It is given as 0.489 for BGE-M3, which is lower than the 0.551 achieved on the full dataset. This means that the German language does not negatively affect retrieval. The likely remaining reason for the difference is the vastly larger number of vectors present in the search space for

the sovanta dataset, confirming the intuition that larger vector spaces lead to a more challenging retrieval process.

In summary, the baseline evaluation highlights the strong overall performance of BGE-M3 and the BGE-Reranker across both datasets, while also underscoring the influence of context size, language, and dataset characteristics on retrieval effectiveness. It also shows that smaller reranking models are usually not more effective than using no reranking at all.

4.1.2. Metric Selection

This ground-truth evaluation serves as the baseline for further retrieval metrics. As discussed in Section 2.2.1, Context Relevance is currently the only LLM-based metric for retrieval, with three main implementation variants. We select TruLens and its variants for this evaluation, as the RAGAs implementation emphasizes precision — contrasting with our recall-based baseline — and the ARES version uses a fine-tuned LLM that we do not have access to. As mentioned before, TruLens also has the benefit of providing several versions of the metric that can be compared with each other.

In detail, the Context Relevance metric offers two primary configuration options. The first is that Chain-of-Thought reasoning can be enabled or disabled. If CoT is enabled, the Context Relevance system prompt instructs the LLM to reason step by step about the user query and search results and to provide reasoning for the final rating. This approach increases token usage and response times, but CoT is commonly expected to improve LLM outputs on complex queries (Wei et al. 2022).

The second variation of the metric is that the scale for the LLM's output can be modified. By default, Context Relevance uses a four-point Likert scale. However, this can be changed to any other scale, for example 0 to 10. Additionally, the choice of LLM also has an influence on the output.

Based on these configuration options, we evaluate three metric variants: (1) standard Context Relevance, (2) Context Relevance with CoT reasoning, and (3) Context Relevance with a 0 to 10 scoring scale. As described before, we use two LLMs on the sovanta dataset and five on WikiEval, resulting in six combinations for the sovanta dataset and 15 for WikiEval. The system prompts of all variants can be found in the TruLens implementation (tru 2025).

To reduce computational effort, we evaluate all metric variants on representative dataset subsets. These subsets are created by randomly sampling 15 prompts from the sovanta dataset and 10 prompts from the WikiEval dataset, representing approximately 20% of the answerable queries in each dataset. On the sovanta dataset, the 15 prompts are sampled uniformly across clusters.

For each prompt, we use a reduced hyperparameter set consisting of $embed\_model \in$ {BGE-M3, Multilingual-E5-Large}, $chunk\_size \in \{64, 128, 256, 512\}$, $top\_k \in \{2, 12\}$, and $rerank\_model \in$ {BGE-Reranker, None}. This subset is designed to cover a wide range of context sizes while ensuring comparability across hyperparameter configurations by holding prompts constant. This would not be given by computing a fully random sample. Consequently, the sovanta sample contains 480 data points and the WikiEval

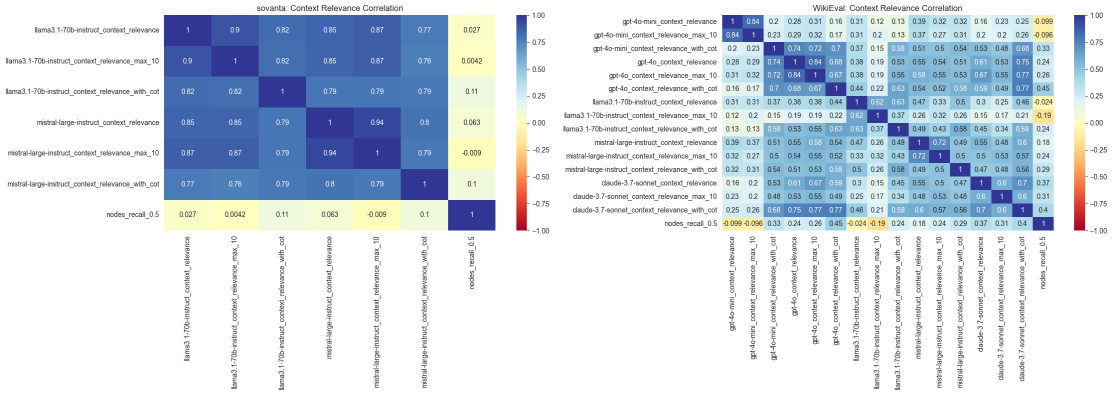

Figure 4.7.: Correlation matrices of Context Relevance metrics on both datasets

sample 320 data points.

After preparing the samples, we run the aforementioned variants of Context Relevance on the respective prompts. All metric variants executed successfully without runtime errors. Figure 4.7 illustrates the correlations of the different metrics with each other and with retrieval recall.

On the sovanta sample, Llama3.1 with CoT has the highest correlation with recall at 0.11. Mistral-Large with CoT has a similar correlation, and all other metrics have correlations that are very close to zero. However, across both LLMs, the standard four-point Likert scale slightly outperforms the 0 to 10 scale. These results indicate that the Context Relevance metric has limited effectiveness in identifying relevant content on the sovanta dataset, but that using CoT at least improves the results slightly.

On the WikiEval dataset, GPT-4o with CoT has the highest correlation at 0.45. In fact, using CoT improves the results for all models, while the different scoring scale does not have an effect on the results. The improved results on the WikiEval dataset cannot be attributed solely to the strength of GPT-4o, as Mistral-Large and Llama3.1 also achieve better results here, though they are not as effective. Another interesting observation is that for the most part, the correlations between different metric versions used by the same LLM are higher than the correlations between the same metric versions between LLMs, indicating that the choice of LLM has a greater impact than the specific version of the metric.

In summary, Llama3.1 performs best on the sovanta dataset and GPT-4o best on WikiEval. In both cases, using CoT reasoning improves the results. However, the metrics are significantly higher correlated with recall on the WikiEval dataset, indicating a better ability of the LLMs to determine whether the retrieved content is relevant to the query. The highest-performing metrics are now computed on the full datasets to analyze the full results.

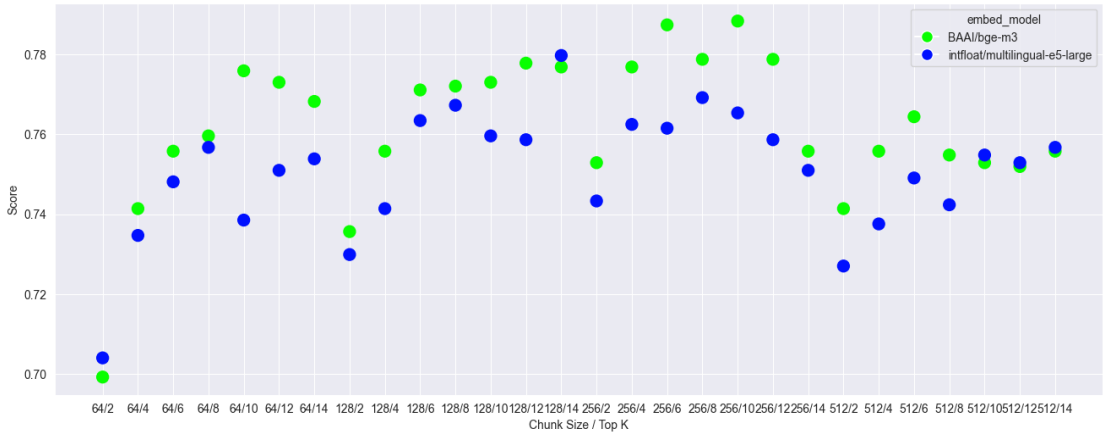

Figure 4.8.: Context Relevance (Llama3.1-70b, with CoT) per embedding model on the sovanta dataset

4.1.3. Context Relevance

sovanta Dataset

The result of the full Context Relevance evaluation in terms of embedding models on the sovanta dataset is presented in Figure 4.8. It shows that Context Relevance rates BGE-M3 as more effective than Multilingual-E5-Large, which is consistent with the baseline. Specifically, BGE-M3 has an average Context Relevance of 0.782, while Multilingual-E5-Large has 0.751. This indicates that Context Relevance reflects the same performance ranking, albeit with a slightly reduced margin compared to the baseline.

Figure 4.9 shows these results grouped by rerank models. BGE-Reranker (cr=0.780) outperforms both the no reranker setting (cr=0.761) and the Mixedbread models — Base (cr=0.760) and XSmall (cr=0.724). This is the same order that is also given by the baseline, though again, the performance differences are smaller.

Unlike the baseline, both Figures indicate no clear positive correlation between longer context sizes and higher Context Relevance scores. In fact, a chunk size of 512 performs worse than 256 and 128 and almost equally to 64. This suggests that Context Relevance underestimates the effectiveness of larger context sizes.

One possible explanation for this phenomenon might be the *lost in the middle* effect discussed earlier (Liu et al. 2023), because with higher context sizes, the probability of relevant content being placed in the middle of the context and being surrounded by irrelevant content increases. Alternatively, the LLM may inadvertently penalize retrieved content that contains irrelevant text, even if the correct information is also present. The TruLens system prompt for Context Relevance tries to mitigate this by explicitly stating that long search results should score equally well as short results (tru 2025), but inevitably, the amount of irrelevant content in the search result increases with longer contexts.

Figure 4.10 presents the Context Relevance results by cluster. Similarly to embedding

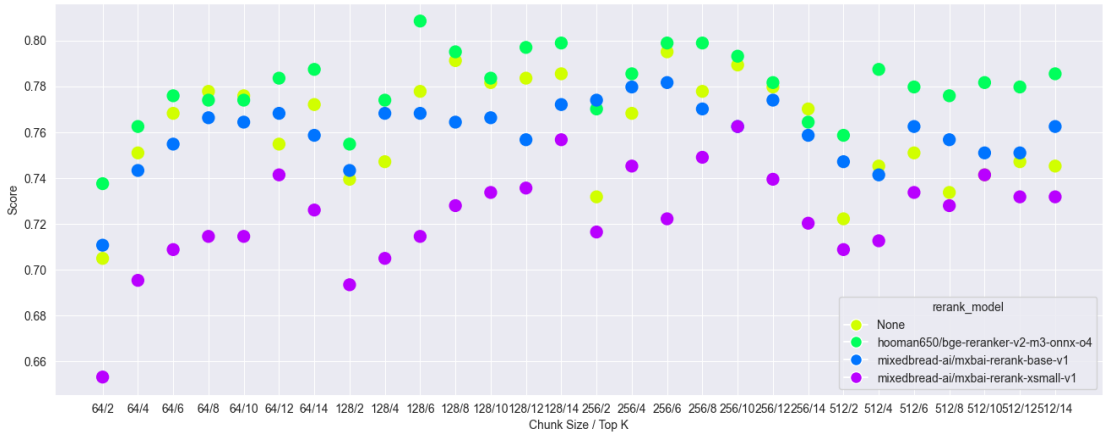

Figure 4.9.: Context Relevance (Llama3.1-70b, with CoT) per rerank model on the sovanta dataset

and rerank models, the cluster performance order is the same as in the baseline, but with smaller margins.

The highest Context Relevance score (cr=0.82) is achieved with the configuration BGE-M3 and BGE-Reranker using six chunks of 128 tokens.

In total, the correlation between Context Relevance and recall on the sovanta dataset is **0.231**. The correlation is as high as 0.386 for cluster HR and as low as -0.154 for summaries. This is expected, as summaries often require comprehensive coverage, making it challenging for any metric to assess completeness without access to ground-truth.

While Context Relevance struggles to accurately assess the impact of the context size, it effectively captures relative performance differences across embedding models, rerankers, and clusters.

WikiEval Dataset

Figures 4.11 and 4.12 reveal the Context Relevance results on the WikiEval dataset. In terms of the embedding model, Multilingual-E5-Large slightly outperforms BGE-M3 with a marginal improvement of 0.2%. On the baseline, the results are reversed and the models have a performance gap of 0.4%. Therefore, both models demonstrate nearly identical performance across both metrics, which is why the results can still be deemed accurate. Regarding rerank models, Context Relevance outputs the same model performance order as the baseline with similar margins.

Both Figures also show again that the scores generally increase with higher context sizes, but that there is a ceiling to this performance improvement. As with the baseline, performance plateaus at a context size of approximately 1,000 tokens. However, similar to the sovanta evaluation, this evaluation also shows a slight decline in performance beyond this threshold, as chunk size 512 has a lower average Context Relevance than 256.

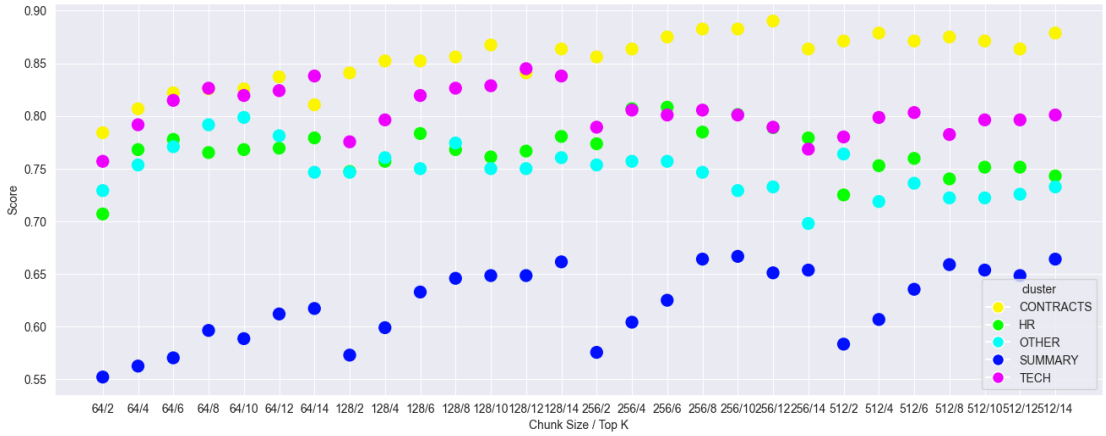

Figure 4.10.: Context Relevance (Llama3.1-70b, with CoT) per cluster on the sovanta dataset

Overall, 24 hyperparameter combinations achieve the optimal Context Relevance of 1. This confirms the results from the baseline that perfect retrieval accuracy can be achieved with Document Chat's RAG pipeline on the WikiEval dataset.

The overall correlation between Context Relevance and recall is **0.465** on the WikiEval dataset. Although this is only a moderate correlation, the results confirm that Context Relevance is an appropriate metric for determining embedding model and rerank model performance. In contrast to the sovanta dataset, Context Relevance on WikiEval also more reliably reflects the impact of varying context sizes, despite some minor inaccuracies.

4.1.4. Error Analysis

In the following section, we conduct a detailed error analysis of the Context Relevance metric. The goal of this analysis is to identify the types of queries that lead to incorrect Context Relevance scores, to define specific error categories, and to assess their distribution across the dataset.

There has been some work done on evaluating outputs of LLM-based metrics using LLMs, for example by Kulkarni et al. on LLM-based hallucination detection (Kulkarni et al. 2025). However, to our knowledge, such an approach has not been applied to RAG-specific evaluation metrics and there is therefore no notion of specific error classes for these metrics in the literature. This highlights a current gap in the literature on LLM-based evaluation of RAG systems.

The same approach is applied to the generation metrics in Section 4.2.7.

Approach

For each dataset, we identify all data points with an absolute deviation greater than 0.5 between Context Relevance and recall. This returns the data points with a strong

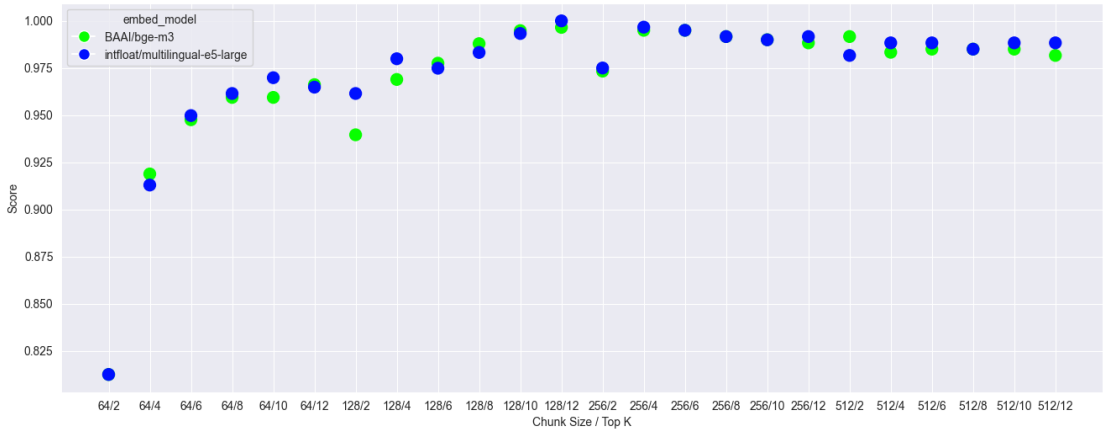

Figure 4.11.: Context Relevance (GPT-4o, with CoT) per embedding model on the WikiEval dataset

disagreement between the two metrics. We then manually examine 30 randomly selected example cases. During this examination, we define a number of distinct error classes based on the observations and assign exactly one error class to each prompt in the sample.

Since manually labeling all erroneous cases is impractical, yet we still want to get an approximate estimation of the total error class distribution, we apply an LLM-based heuristic. Specifically, we ask various LLMs with different system prompts to make the same classification on the 30 example cases. The LLMs always have the option to output *OTHER* as the error class in case the example does not match any of the provided classes. We can then examine which combination of LLM and prompt leads to the highest prediction accuracy compared to the human labels. We then apply the best-performing model and prompt configuration to the full set of error cases to estimate the overall error class distribution.

The final prompt used for error classification is given in the appendix.

Context Relevance Errors

On the sovanta dataset, 6,823 of the 19,488 evaluated retrieval data points incur an absolute difference of more than 0.5 between recall and Context Relevance. 67% of these errors have recall values smaller than 0.2 and 13% have a recall of 1. For Context Relevance, 44.3% have a value of 1, 41.2% have 0.67, 9.9% have 0.33 and 4.6% have 0.

This indicates that most errors result from low recall values, with the LLM incorrectly classifying irrelevant content as relevant. The errors are equally distributed across prompts and hyperparameters, albeit they appear more frequently on smaller context sizes. This makes sense as smaller context sizes inherently have lower recall.

On the WikiEval dataset, 794 of the 9,600 data points fall under the error classification. Here, 87% of the errors have a Context Relevance of 1, but a recall below 0.5. However,

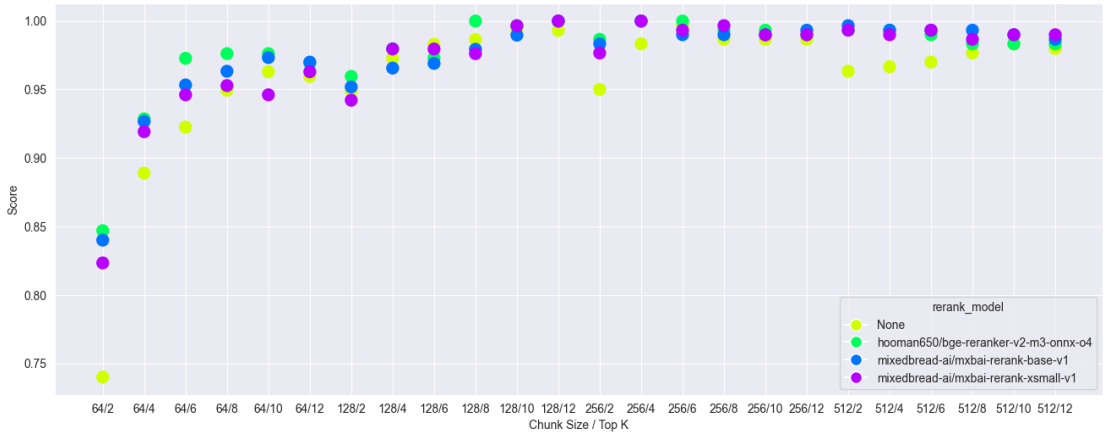

Figure 4.12.: Context Relevance (GPT-4o, with CoT) per rerank model on the WikiEval dataset

the recall values in these cases are not as close to 0 as in the sovanta dataset, which aligns with the overall higher retrieval recall on the dataset.

Error cases are similarly distributed across prompts, embedding models, and rerank models, but almost universally appear with smaller context sizes. This is expected, because as established before, longer context sizes result in very high recall on this dataset, eliminating the problem of too large Context Relevance scores.

Here is an example case from the WikiEval dataset that illustrates the common problem of overestimating Context Relevance:

> **Prompt:** What was the estimated timeline for fully restoring power in Moore County after the shooting attack on the electrical distribution substations?
> **Relevant Text:** Initial estimates were that up to four days could be required to fully restore power in the area.
> **Retrieved Text:** On December 3, 2022, a shooting attack was carried out on two electrical distribution substations located in Moore County, North Carolina, United States. [...] As of December 6, it was estimated that about 35,000 Moore County residents were still without power, and the timeline for completing repairs and restoring power county-wide was revised from December 8 to midnight December 7. By the morning of December 7, the number of affected residents without power was down to about 23,000.
> **Recall (Baseline):** 0
> **Context Relevance:** 1

Although the retrieved text is very close to being able to sufficiently answer the question, it does not actually contain the information about the initial estimate for the timeline. This case illustrates how Context Relevance may assign a perfect score despite the absence of a direct answer due to semantic proximity and contextual overlap. On

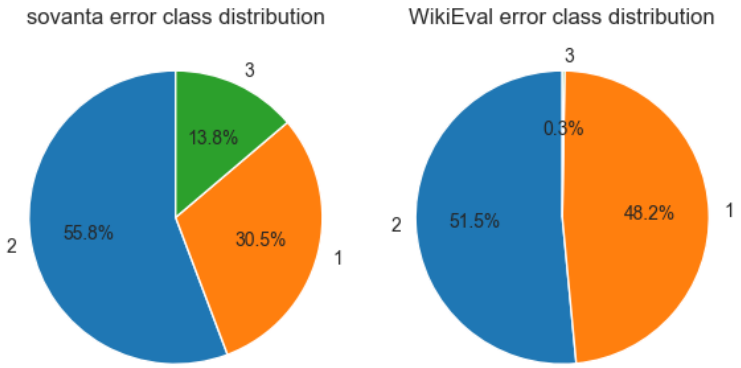

Figure 4.13.: Context Relevance error class distribution

the other hand, there also exist cases where the retrieved text does contain the necessary information with different phrasing, therefore falsely leading to a low recall.

Overall, we identify the following three error classes across both datasets:

1. The retrieved text contains the needed information, but the relevant text contains extra or broader content that is technically unnecessary to answer the query or describes the information differently. The essential answer is still present.

2. The retrieved text omits important details that are needed to answer the query properly or entirely lacks the necessary information to answer the question.

3. The retrieved text contains the relevant text and therefore has high recall, but Context Relevance fails to classify it as relevant.

Since class three can be automatically identified based on high recall and a low Context Relevance score, this leaves only the other two cases and the option *OTHER* to the LLM.

On the sovanta dataset, both Mistral-Large and Llama3.1 achieve an accuracy compared to human labels of 0.73. We use Mistral-Large on the entire dataset, because it is generally the stronger LLM (lma 2025). On the WikiEval dataset, Claude-3.7-Sonnet achieves the highest maximum accuracy (0.73) of all LLMs. As mentioned before, humans often also align with each other with accuracies of only 0.81 (Zheng et al. 2023), so these scores can be considered sufficient. We therefore use these two LLMs to get an estimate of the error distribution on the full datasets. Figure 4.13 shows the distribution of error classes on both datasets.

Notably, error class one represents 31% and 48% of all errors, respectively. Since the retrieved text does contain the relevant information in this case, just in a different form than the relevant text, it means that these cases do not point to actual errors in the Context Relevance metric. To incorporate this into the final result, we manually set recall to 1 for all rows where the error class is one, indicating that the retrieved text contains the relevant information.

Through this, the correlation between Context Relevance and recall is increased from **0.465 to 0.525** for WikiEval and from **0.231 to 0.368** for the sovanta dataset. This demonstrates that the effectiveness of the Context Relevance metric is higher than initially estimated. Additionally, it reveals problems with the ground-truth-based approach, as the labels apparently do not cover all text that contains the relevant information, especially on the WikiEval dataset. This highlights that ground-truth labels for retrieval need to be written very precisely to capture all possible expressions of the relevant content.

However, the analysis also highlights systematic limitations of Context Relevance. On both datasets, around half of the error cases stem from Context Relevance overestimating the relevance of the retrieved text. This is expected, as LLM-based metrics without access to ground truth cannot reliably assess the completeness of retrieved content.

Furthermore, 14% of the errors on the sovanta dataset are induced by Context Relevance underestimating the relevance of text that actually contains the relevant information. An example of this is the prompt *Can I fly first class?*, where the retrieved text is *Standard class for flights is coach (economy) class.* This is the relevant text that the system is looking for, however, Context Relevance does not recognize it as sufficient for answering the question, assigning a score of only 0.33.

Error class three is almost not present on the WikiEval dataset. This is in line with the earlier observation that the Context Relevance is 1 in 87% of cases and therefore, the problem of recall being too high does not occur. We suspect that the reason for this is the cleaner and simpler dataset, where the content is more semantically separated.

In summary, the error analysis of the Context Relevance metric shows that its effectiveness is higher than initially estimated, with a significant portion of error cases being attributed to syntactic differences between the retrieved text and the labels. Nevertheless, it also shows that a large portion of the errors is due to the fact that Context Relevance cannot accurately capture the completeness of the retrieved information. Furthermore, our approach demonstrates that LLMs can be used to estimate the error distribution of LLM-based metrics with sufficient accuracy.

4.1.5. Runtime Analysis

While the primary focus of this evaluation lies in assessing retrieval and generation quality in RAG systems, we also report on auxiliary performance metrics such as response times, which are byproducts of the evaluation process. In this section, we present the timing implications of the different components in the retrieval step. It is important to note that all reported response times are specific to the hardware setup described earlier and may vary on different systems. We present these timings as unified results over all data points in both datasets.

We begin by examining the average query embedding time for both models. BGE-M3 demonstrates slightly faster performance (33.6ms) compared to Multilingual-E5-Large (38.3ms). However, both models are highly efficient, and the query embedding step constitutes only a small fraction of the overall execution time in a RAG pipeline.

Figure 4.14 illustrates the average retrieval time per rerank model. The fastest con-

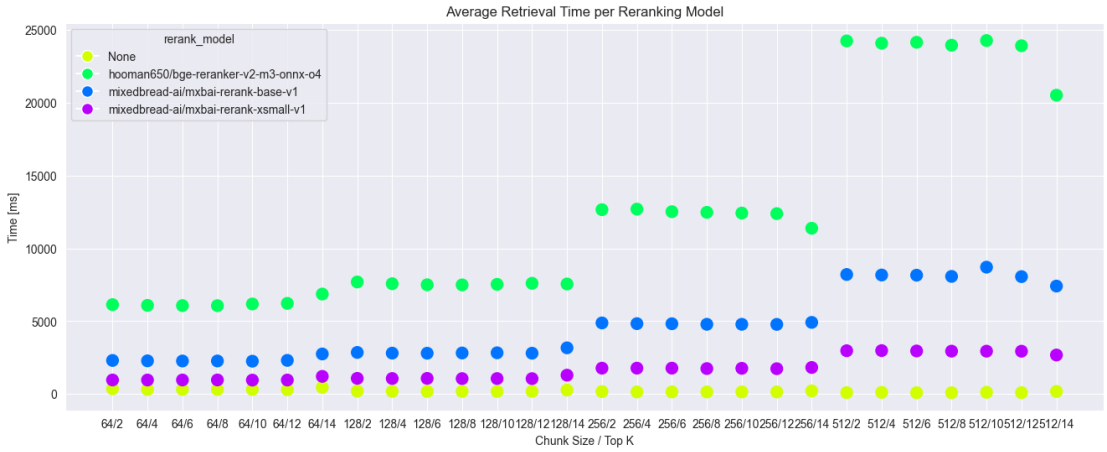

Figure 4.14.: Average retrieval time per rerank model

figuration is using no reranking (189.4ms), followed by Mixedbread-XSmall (1,686.2ms), Mixedbread-Base (4,533.1ms), and BGE-Reranker (12,436.6ms). In most practical applications, delays exceeding a few seconds are undesirable, particularly given that the LLM generation step introduces additional latency. Consequently, the two larger rerank models are impractical on our hardware due to excessive latency, while the smallest model (Mixedbread-XSmall) offers limited improvement in retrieval quality, as shown before. Substantial hardware resources are thus required to deploy reranking approaches that offer meaningful benefits to end users without adding too much latency.

Figure 4.15 reports average retrieval times per embedding model, excluding the reranking step to isolate the performance of the initial vector retrieval process. Again, BGE-M3 (186.4ms) slightly outperforms Multilingual-E5-Large (192.5ms). In contrast to the rerank models, increasing the context size leads to reduced retrieval time. This is expected, as fewer, larger chunks result in fewer cosine similarity computations during the retrieval process. Conversely, when rerank models are applied, larger context sizes lead to increased retrieval times due to longer inputs to these models.

In summary, both embedding models offer fast performance, with BGE-M3 holding a slight advantage. However, the use of larger rerank models increases response time by up to 25 seconds in our setup and thus requires high-performance hardware with GPU acceleration. The benefits in retrieval quality provided by such models therefore need to be carefully weighed against the associated latency.

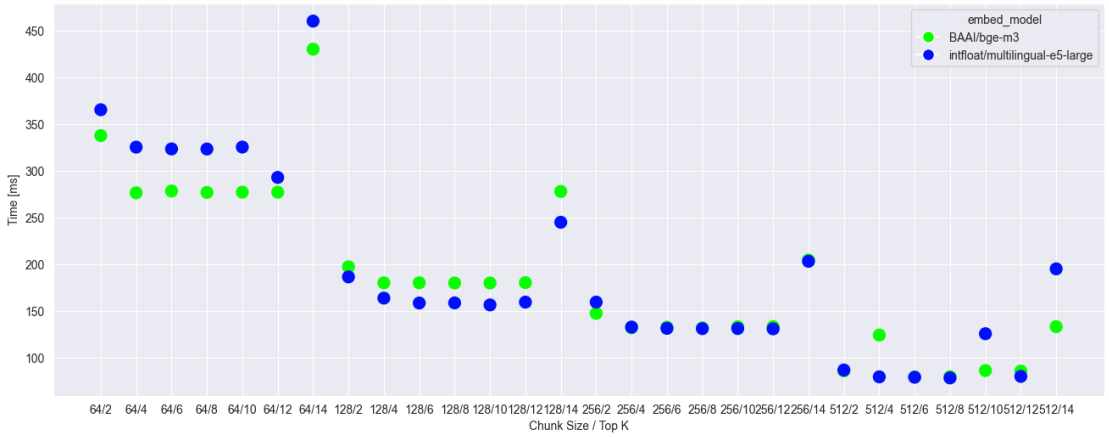

Figure 4.15.: Average retrieval time (without reranking) per embedding model

4.2. Generation Evaluation

After evaluating the retrieval results, we now proceed to generation, following a similar outline.

4.2.1. Baseline: LLM Judgement with Ground Truth

A key challenge in generation evaluation is defining a reliable and meaningful baseline. Given the primary objective of assessing LLM-based metrics that do not rely on reference answers, we employ a reference-based metric as the baseline to benchmark their performance.

As covered in Section 2.2.2, human judgement based on ground truth is still the gold standard for the evaluation of LLM outputs. However, human annotations are costly to obtain, especially at the scale of the datasets used in this work. Therefore, we employ a two-step approach: First, we manually label the outputs on subsets of our datasets. Then, we compare these labels with the results from various automatic metrics. Subsequently, the best-performing metric can be used as a baseline on the full datasets.

For the sovanta dataset, we randomly sampled up to 10 prompts from each of the six clusters, yielding a total of 51 prompts. For each prompt, two hyperparameter combinations were randomly selected, resulting in 102 examples. Similarly, for WikiEval, two hyperparameter configurations per prompt were sampled, leading to 100 examples in total. This strategy ensures a balanced distribution across prompts, configurations, and, in the sovanta case, clusters.

Each sampled output was manually annotated to assess its quality relative to the corresponding reference answer, using the following five-point Likert scale:

1. Bad: Completely wrong or misleading

2. Poor: Significant issues; only partially correct or unclear
3. Acceptable: Reasonable but some flaws; not as complete or precise
4. Good: Minor deviations from gold answer; still valid, useful, and accurate
5. Perfect: Output matches the gold answer exactly or improves upon it with no errors

The labeled datasets now provide the basis for the evaluation of several LLM-based metrics. Specifically, we use four LLM judgement system prompts based on those proposed by Zheng et al. (Zheng et al. 2023):

- Reference-guided LLM judgement without CoT
- Reference-guided LLM judgement with CoT
- Reference-guided LLM judgement with CoT and explicit Likert scale
- Reference-free LLM judgement (similar to single answer grading in Zheng et al. 2023)

This selection of prompts covers both reference-guided and reference-free setups, as well as configurations with and without CoT reasoning. We do not use a pairwise grading approach, because the LLM-based metrics without ground truth that we examine later also grade single answers. The LLMs used for evaluation are the same models used for retrieval evaluation. To best match the human annotations, all LLM judges are instructed to output scores between 1 and 5. All prompts are documented in the appendix.

To determine the quality of the different LLM judges' outputs, we measure their correlation with the human judgement labels. As described before, Zheng et al. find that the agreement between humans is only 0.81 (Zheng et al. 2023), so this is the benchmark for all LLM judges.

Figure 4.16 shows the correlation between each LLM judge and human annotations on the sovanta dataset, as well as correlations between the judges. In general, using Chain-of-Thought reasoning leads to worse results than not using it. Explicitly providing the Likert scale for judgement yields the same result for Llama3.1 and small improvements for Mistral-Large. It is also visible that using LLM judges without reference answers substantially diminishes the correlations with human preferences. This suggests that, at least on this subset, reference-free judging lacks the grounding necessary to reliably assess the quality of the outputs.

Overall, reference-guided judgement with Llama3.1 and without CoT has the highest correlation with human judgement at 0.73. We argue that this correlation is sufficiently high to serve as a baseline for all following metrics.

Nevertheless, we also want to give an example of one of the few cases where there is a large deviation between human and LLM judgement. The prompt *Is there a project called MCI at sovanta?* with the gold answer *No, there is no project called MCI at sovanta*

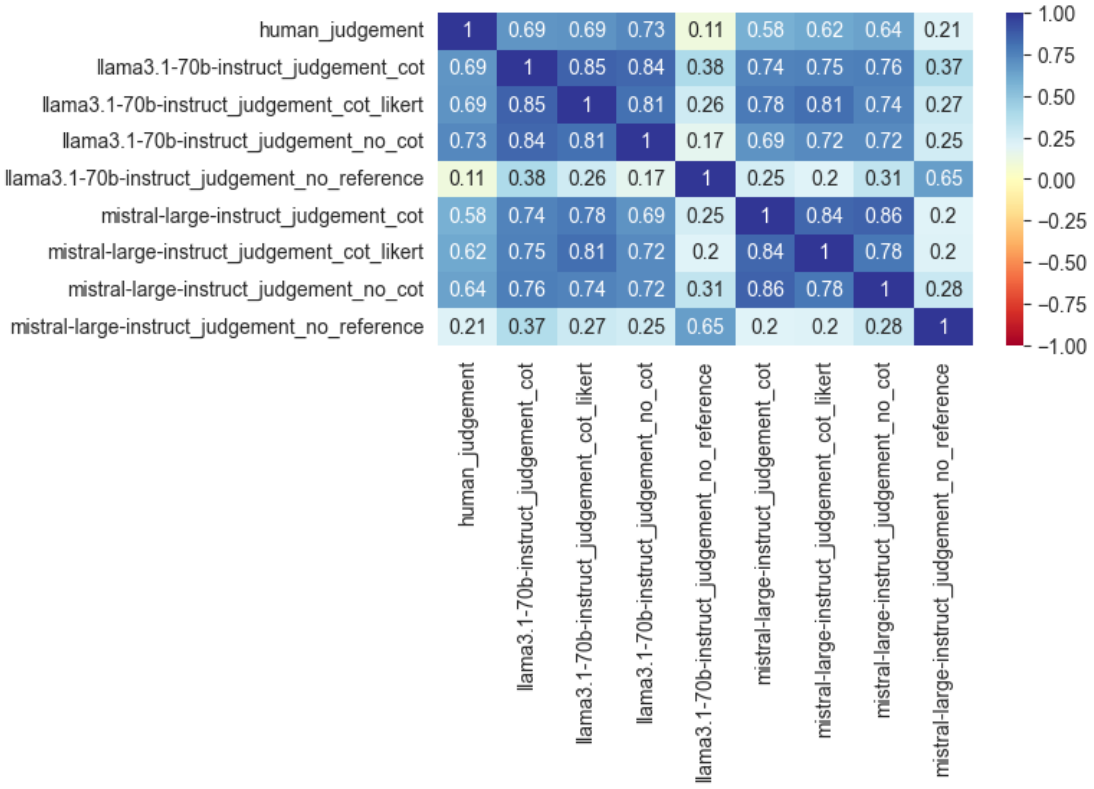

Figure 4.16.: Correlation of LLM judges with human judgement on the sovanta sample

produces the system answer *No information is available in the provided context about a project called MCI at sovanta.* In the context of RAG, those two answers essentially mean the same, which is why the human judgement was set to 5 (perfect). However, Llama3.1 only gives a score of 2 (poor).

Refining the prompt to account for such cases does lead to an increase of the LLM-based score to 4 (good) on this case, but also results in a lower overall correlation. Therefore, this case further illustrates the necessity to precisely design system prompts for LLM-based judges.

The results on the WikiEval dataset that are shown in Figure 4.17 are similar. For all tested LLMs except Llama3.1, using CoT gives worse results than not using it, for example with a correlation that drops from 0.77 to 0.71 for GPT-4o. Providing the explicit Likert scale slightly reduces the correlations for all LLMs except Claude-3.7-Sonnet.

Additionally, reference-free judgement again results in significantly lower correlations, although they are higher than on the sovanta dataset. GPT-4o without references even yields a correlation with human judgement of 0.57. This indicates that evaluating the answer quality on the WikiEval dataset without a reference is easier than on the sovanta dataset.

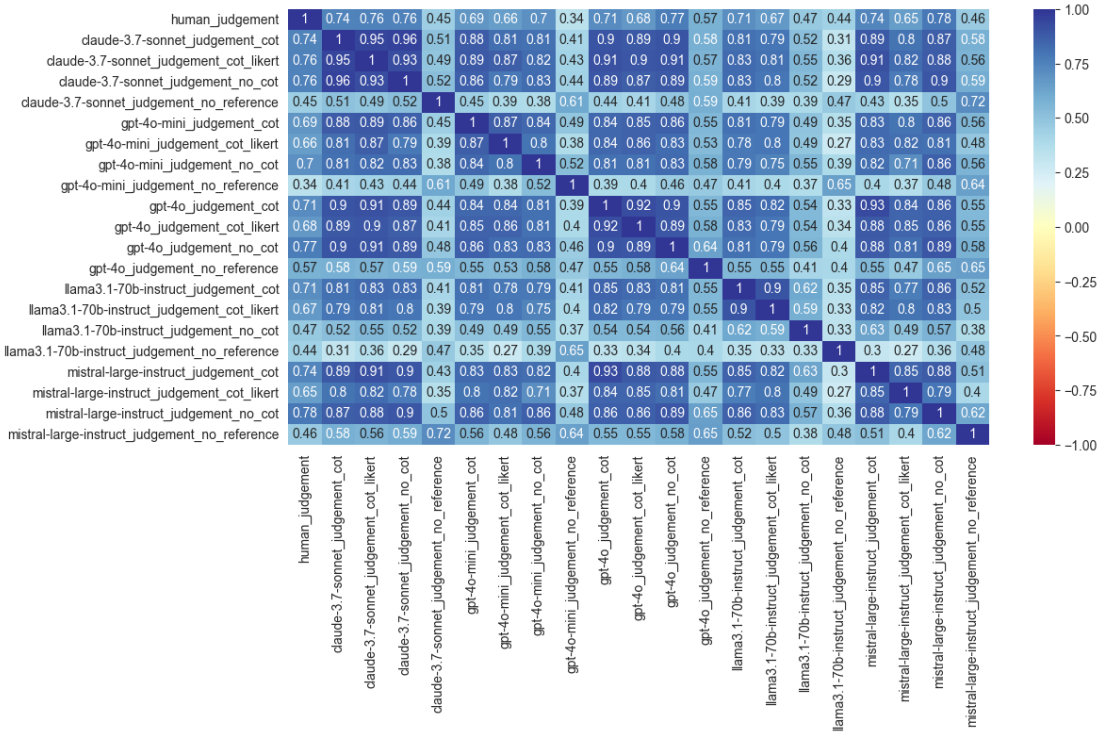

Figure 4.17.: Correlation of LLM judges with human judgement on the WikiEval sample

In total, reference-guided judgement with Mistral-Large and without CoT has the highest correlation with human judgement (0.78) and is close to the benchmark of 0.81.

To further validate using LLM judgement as a baseline that substitutes human judgement, we compare the correlation of the best respective LLM judges with the correlations that other metrics have with the human judgement on our dataset samples. Specifically, we evaluate BLEU, ROUGE-1, ROUGE-2, ROUGE-L, BERTScore (Precision, Recall, and F1) and retrieval recall. For the calculation of BERTScore, we use the standard models proposed in the original work, i.e., RoBERTa-Large for English queries and BERT-cased-multilingual for German queries (Zhang et al. 2020). The results are given in Figure 4.18.

Both LLM judges are the best-performing metrics in both cases, which means that they are most effective at substituting human preferences. However, ROUGE-1 achieves a correlation with human judgement that is not significantly lower, at 0.68 and 0.70, respectively. On the WikiEval dataset, BERTScore-Recall even produces 0.75, coming very close to the score of the LLM judge.

Given earlier observations, several LLM-based judges underperform in comparison with ROUGE, BLEU, and BERTScore in correlation with human annotations. These results highlight that only the most effective LLM judges are able to improve on these metrics, and that LLM judges are not always better metrics.

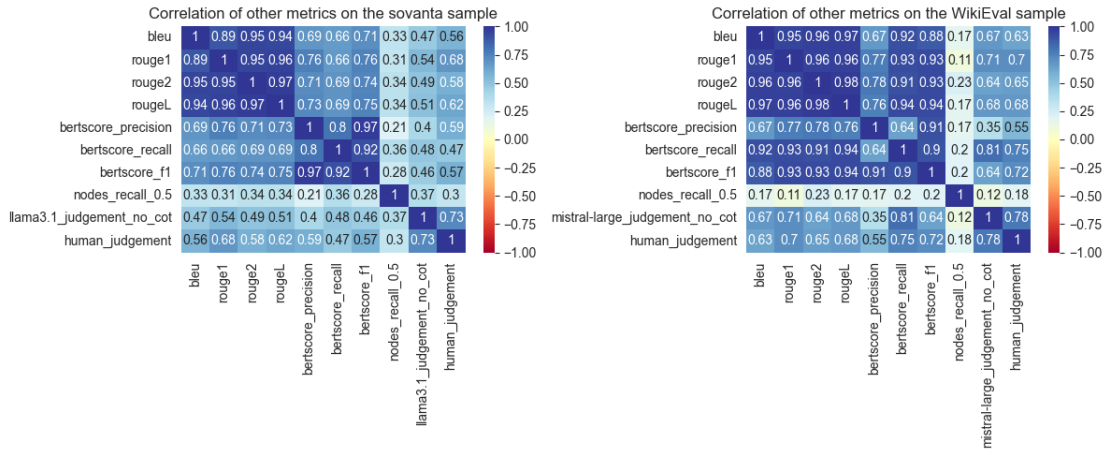

Figure 4.18.: Correlation of other metrics with LLM judgement and human judgement

Interestingly, recall only has a weak correlation with the human-annotated answer quality. Our investigation shows that in 41% of the cases on the sovanta sample, recall is higher than the human judgement, while it is lower in 25% of cases. This indicates that high-quality retrieval does not necessarily lead to high-quality answers and that lower quality retrieval can still lead to good answers.

In summary, reference-guided LLM judges without CoT consistently achieve the highest correlation with human annotations. Llama3.1 performs best on the sovanta data (r=0.73), while Mistral-Large achieves the strongest alignment on WikiEval (r=0.78). Using CoT and providing explicit grading scales generally leads to worse results. Although BERTScore and ROUGE also demonstrate moderate to strong correlations, they fall short of the top-performing LLM judges.

We therefore adopt the strongest-performing LLM judges as our evaluation baselines for the full datasets, providing a robust reference point for subsequent metric comparisons.

Results

This section analyzes the impact of different hyperparameter combinations on generation quality, based on the baseline LLM judges. These results identify the optimal configuration of the RAG pipeline for maximum answer quality. Subsequent evaluations focus on how other metrics align with the baseline and therefore how accurately they can reproduce these results.

Figure 4.19 shows the results on the sovanta dataset grouped by the two LLMs for generation. Mistral-Large achieves higher scores than Llama3.1 across all context sizes, with an average LLM judgement of 0.737 compared to 0.669.

Answer quality increases with larger context sizes, supporting the earlier retrieval findings. However, this improvement is less pronounced than for recall, suggesting that high-quality answers can still be generated when only parts of the relevant context are

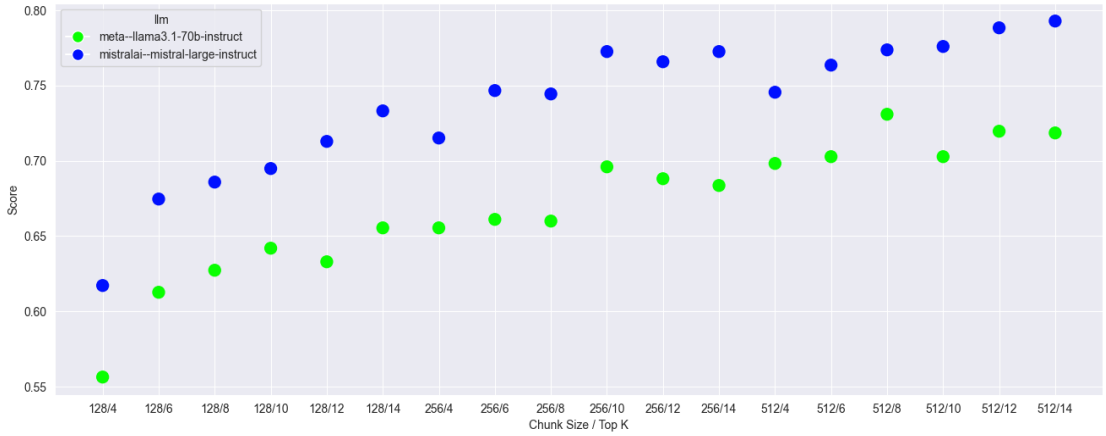

Figure 4.19.: LLM judgement (Llama3.1) per LLM on the sovanta dataset

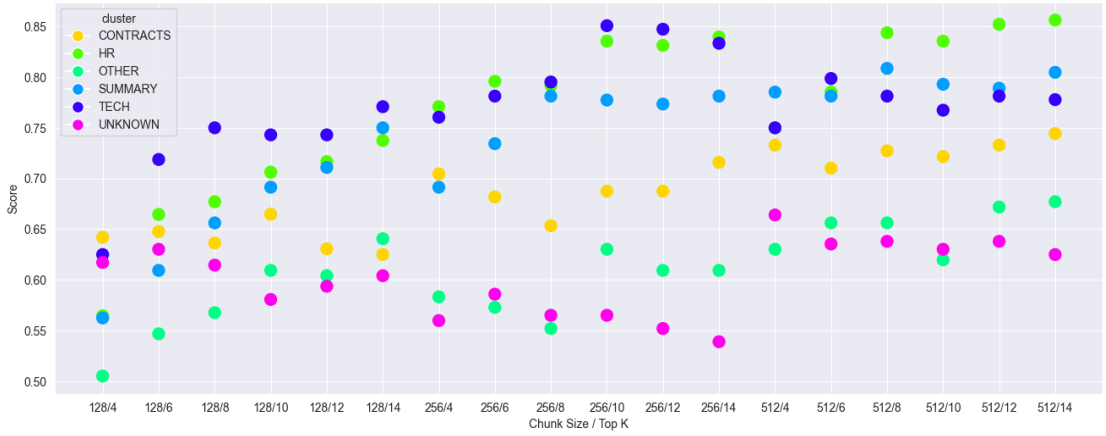

Figure 4.20.: LLM judgement (Llama3.1) per cluster on the sovanta dataset

retrieved. This is in line with the previous observation of low correlation between recall and human answer quality annotations.

The use of the BGE-Reranker leads to a moderate improvement in answer quality, increasing the average score from 0.695 to 0.712 compared to the setting without reranking. This shows that reranking also directly improves answer quality and not only retrieval, though the performance improvement is smaller.

The highest overall judgement score (0.797) is obtained with Mistral-Large, using 14 chunks of 512 tokens and the BGE-Reranker.

In Figure 4.20, the same results are presented by clusters. Among the clusters, TECH achieves the highest answer quality (judgement=0.771), followed by HR (judgement=0.770), SUMMARY (judgement=0.738), CONTRACTS (judgement=0.686), and OTHER (judgement=0.608). The lowest performance is observed for unanswerable questions (judgement=0.602), which indicates that the system is susceptible to hallucinating

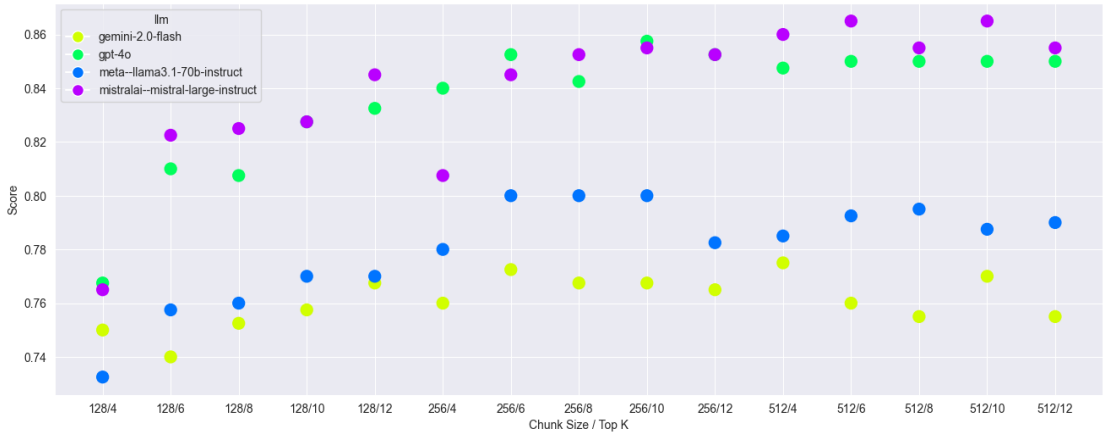

Figure 4.21.: LLM judgement (Mistral-Large) per LLM on the WikiEval dataset

answers instead of stating that the information asked for cannot be found. Research shows that using RAG significantly reduces hallucinations (Huang et al. 2025), but the results presented here reinforce the observation that queries lacking contextual grounding remain a challenge for RAG systems, as they are the worst-performing type of question here.

The results also confirm that summaries benefit the most from increasing context sizes, while questions on technology and HR already yield high scores with lower context sizes.

Figure 4.21 presents the results on the WikiEval dataset. Mistral-Large achieves the highest average score (judgement=0.840), slightly surpassing GPT-4o (judgement=0.836), with both significantly outperforming Llama3.1 (judgement=0.780) and Gemini-2.0-Flash (judgement=0.761). Again, larger context sizes generally lead to better results, though this correlation is not as pronounced as on the sovanta data, just as shown by the retrieval evaluation.

In terms of reranking models, BGE-Reranker (judgement=0.807) again slightly outperforms using no reranker (judgement=0.801), but again, the difference is smaller than in the retrieval evaluation.

Since the correlation between context sizes and scores is lower on the WikiEval dataset, the top-performing hyperparameter configurations exhibit greater diversity than those observed on the sovanta dataset. Table 4.3 shows the 10 best configurations.

Among all configurations, Mistral-Large with 12 chunks of 512 tokens and without reranking has the highest score, followed by six configurations that yield exactly the same result. Additionally, the judgement scores are higher than on the sovanta dataset, indicating that the general answer quality is higher on WikiEval. This is in line with the higher retrieval quality measured before.

To sum up, Mistral-Large is the best-performing LLM on both datasets, though GPT-4o yields comparable results on the WikiEval dataset. The BGE-Reranker consistently enhances the quality of outputs, albeit to a lesser extent than in retrieval. Furthermore,

Table 4.3.: The 10 hyperparameter combinations with the highest LLM judgement (Mistral-Large) on the WikiEval dataset

LLM	Rerank Model	Chunk Size	top_k	LLM Judgement
Mistral-Large	None	512	12	0.870
GPT-4o	BGE-Reranker	256	12	0.865
Mistral-Large	BGE-Reranker	512	4	0.865
Mistral-Large	BGE-Reranker	512	6	0.865
Mistral-Large	None	512	6	0.865
Mistral-Large	BGE-Reranker	512	10	0.865
Mistral-Large	None	512	10	0.865
GPT-4o	BGE-Reranker	256	10	0.860
GPT-4o	BGE-Reranker	512	12	0.860
Mistral-Large	BGE-Reranker	512	8	0.860

larger context sizes generally improve output quality, particularly for the sovanta dataset. Cluster-level analysis also highlights underperformance on unanswerable queries, indicating potential for further improvements to mitigate hallucinations.

4.2.2. Metric Selection

As discussed in Section 2.2.2, a wide range of metrics exists for the evaluation of LLM outputs, both with and without reference answers. The previous section established that LLM-based metrics with ground truth exhibit the strongest correlation with human judgement. In this section, we analyze how modern LLM-based metrics without ground truth perform in comparison.

Additionally, we include traditional metrics such as BLEU, ROUGE, and BERTScore in our comparison, as recent evaluation methods often omit direct benchmarking against these established baselines.

Given their computational efficiency, BLEU, ROUGE, and BERTScore are calculated across the full datasets and directly compared to reference-based LLM judgements. We also apply the reference-free LLM judge to the entire datasets to assess how this simpler method performs relative to more sophisticated reference-free metrics.

For the remaining LLM-based metrics, we first evaluate various configurations on representative dataset subsets, following the approach for retrieval presented in Section 4.1.2. Specifically, we again sample 10 random prompts from the WikiEval dataset. For the sovanta dataset, we sample 18 prompts instead of 15 to accommodate the additional cluster UNKNOWN that is present for generation evaluations.

Each prompt in the sample is evaluated under a reduced hyperparameter configuration, with $chunk\_size \in \{128, 256, 512\}$ and $top\_k \in \{6, 12\}$. The set of LLMs is the respective full set used for generation on each dataset.

Embedding and reranking models are held constant, as preliminary analysis and experience suggest their impact on generation quality is comparatively limited relative to chunk size, `top_k`, and LLM choice. This approach yields a sample size of 216 for sovanta and 240 for WikiEval. These samples are used to evaluate different configurations of two

primary metrics: Groundedness and Answer Relevance. As in earlier experiments, we employ the TruLens library due to its flexibility and extensive configuration support. Again, the system prompts of all variants can be found in the TruLens implementation (tru 2025).

In the following sections, we first evaluate various configurations of Groundedness and Answer Relevance on the sampled subsets and then apply the best-performing configurations to the full datasets. These two metrics are subsequently combined with Context Relevance to produce the RAG Triad scores.

Finally, we report the results of traditional metrics and the reference-free LLM judge, comparing their performance to both the RAG Triad and the reference-based LLM-Judgment baseline.

4.2.3. Groundedness

As outlined in Section 2.2.2, the TruLens library offers a variety of configuration options for Groundedness. Accordingly, we first evaluate several metric variants on the described dataset subsets before identifying the best-performing configurations and applying them to the full datasets.

For each evaluator LLM, we compute the following four variants of the Groundedness metric:

- Groundedness on all statements
- Groundedness without trivial statements
- Groundedness with answerability on all statements
- Groundedness with answerability and without trivial statements

Each system-generated answer is first segmented into individual statements using the NLTK Punkt sentence tokenizer. Although this task can also be performed by an LLM in the TruLens library, we argue that the marginal benefits of using an LLM do not justify the additional computational cost and latency. In variants that exclude trivial statements, the LLM is prompted to remove stylistic, trivial, or insubstantial sentences from the list of statements.

For variants incorporating answerability, the LLM also assesses each statement for abstention such as *I don't know* and evaluates whether the question can be answered based on the given context. As described before, answerable abstentions receive a score of 0 and unanswerable abstentions a score of 1. These variants are only included in the sovanta dataset evaluation, because the WikiEval dataset does not contain any unanswerable questions to start with. If the system produces an abstention for an answerable question, standard Groundedness will assign a low score to it anyway.

Each remaining statement is then scored by the LLM on a scale of 0 to 3, based on the extent to which it is supported by the context. These scores are normalized, and the average of all statements is taken to yield a final Groundedness value between 0 and 1.

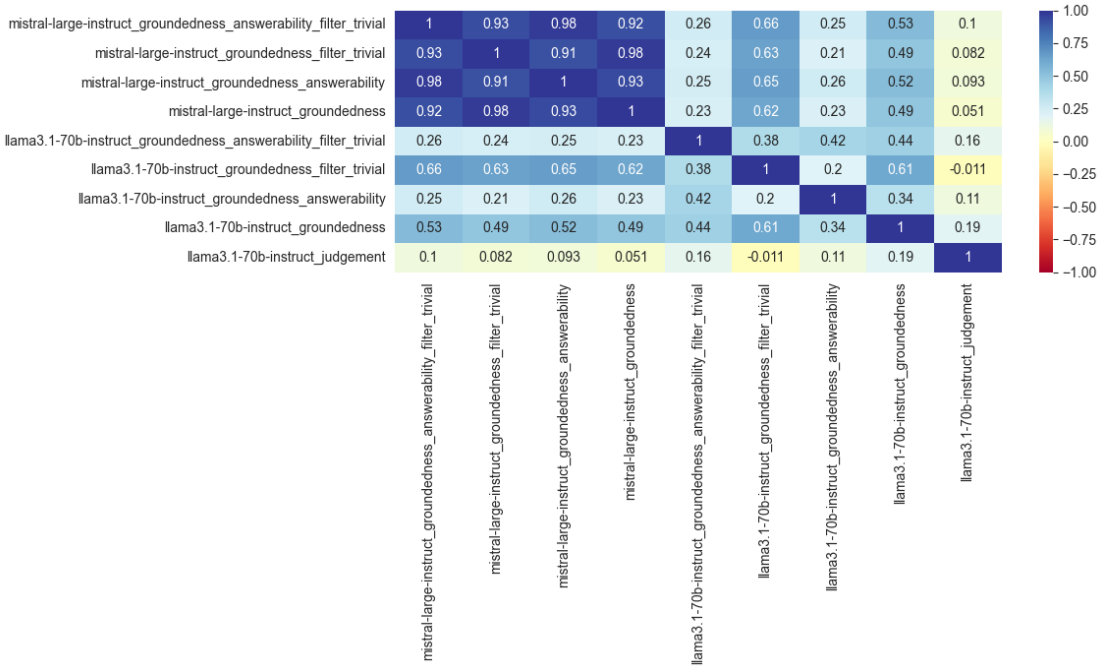

Figure 4.22.: Correlations of Groundedness variants on the sovanta dataset

As a result, four Groundedness variants are computed with two LLMs on the sovanta dataset and two variants using five LLMs are applied to the WikiEval dataset. Figure 4.22 shows the correlation of the different variants with each other and with reference-guided LLM judgement on the sovanta dataset. All Groundedness variants exhibit only weak correlations with the baseline.

For Mistral-Large in particular, the scores of all variants are very strongly correlated with each other, more so than with those produced by Llama3.1. This pattern, consistent with the Context Relevance findings, indicates that the choice of evaluator LLM has a greater influence on metric output than the specific Groundedness variant.

A strong correlation with the baseline is not necessarily expected, given that Groundedness reflects only one aspect of answer quality, whereas LLM judgement assesses holistic performance. Nevertheless, a higher correlation would have been desirable, given the central role of factual consistency in overall answer quality.

Among the evaluated configurations, the standard Llama3.1 Groundedness without answerability and without filtering trivial statements demonstrates the highest correlation with the baseline. Considering answerability and filtering out trivial statements even reduces the results when using Llama3.1, while it very slightly increases the correlation for the variants that use Mistral-Large. This suggests that these extensions of the metric do not have a clear positive performance impact.

Figure 4.23 presents the results on the WikiEval dataset. In this case, none of the Groundedness variants exhibits a positive correlation with LLM judgement. In fact,

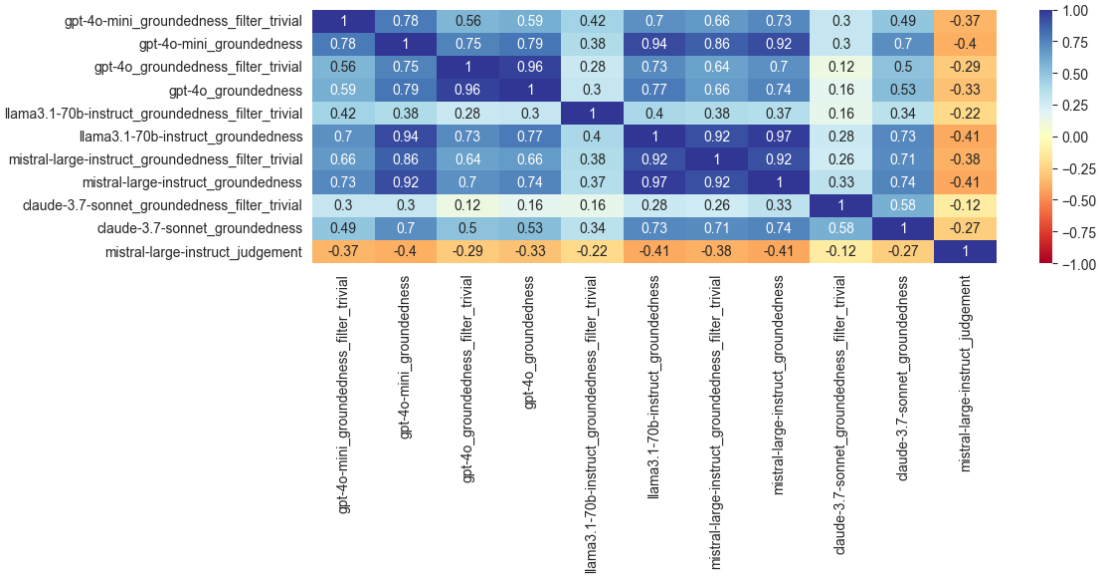

Figure 4.23.: Correlations of Groundedness variants on the WikiEval dataset

some variants even display a moderately negative correlation. The highest, though still negative correlation is observed with Claude-3.7-Sonnet (with trivial statement filtering) at -0.12.

An investigation into this behavior shows that the average Groundedness values on WikiEval are above 0.96 for all metric variants and even reach as high as 0.99, while they are between 0.7 and 0.9 on the sovanta dataset. These values are substantially higher than the values for LLM judgement, indicating that the LLMs consistently either overestimate the Groundedness of the statements in the absence of reference answers, or that almost all answers are grounded in the context, but exhibit other problems such as lacking relevance, though this seems unlikely. The results on answer relevance will provide more clarity here.

The Groundedness variants with the highest positive correlations with LLM judgement are subsequently applied to the full datasets. However, it should be noted that for WikiEval, the highest correlation is still negative.

On the full sovanta dataset, Llama3.1 Groundedness has a **correlation of 0.074 with the baseline**. When unanswerable questions are excluded, the correlation increases to 0.12. On the full WikiEval dataset, Claude-3.7-Sonnet Groundedness with trivial statements filtered out has a **correlation of 0.061 with the baseline**.

These results show that Groundedness does not correlate with the reference-guided LLM judgement baseline on either dataset. However, since the metric is designed to cover only one aspect of answer quality, its utility may be more evident when used in conjunction with other components of the RAG Triad.

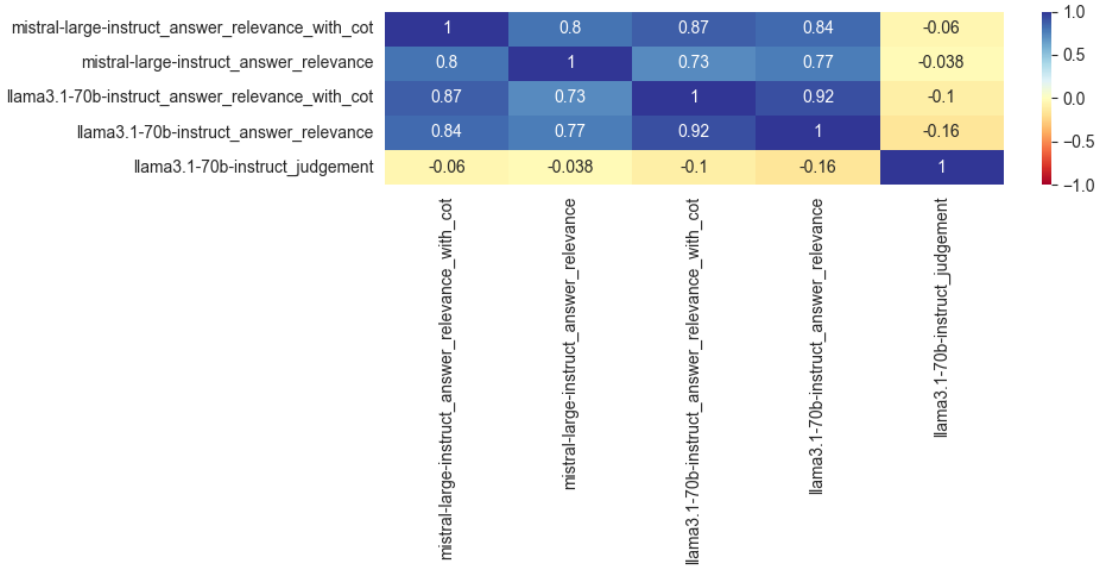

Figure 4.24.: Correlations of Answer Relevance variants on the sovanta dataset

4.2.4. Answer Relevance

This section applies the same evaluation methodology to the Answer Relevance metric. Unlike Groundedness, only two variants are tested: one incorporating CoT reasoning and one without it.

Aside from differences in system prompting based on CoT inclusion, Answer Relevance is computed in the following way: Given a user query and a system-generated answer, an LLM rates the relevance of the answer to the prompt on a scale of 0 to 3. The resulting score is then normalized to fall between 0 and 1.

A consequence of this approach is that abstentions inherently receive a minimum Answer Relevance score. This is a relevant consideration for the sovanta dataset, as it contains cases where an abstention is the expected response.

Results for the sovanta dataset are given in Figure 4.24. All metric variants exhibit correlations with the baseline that are near zero. When unanswerable questions are excluded from the evaluation, the correlations increase slightly, reaching a maximum of 0.095 with Mistral-Large (without CoT).

Incorporating CoT does not lead to a substantial change in metric scores. Given the associated increase in latency and token usage, its application is justifiable only if it results in a significant performance gain.

Figure 4.25 shows the results for the WikiEval dataset. Here, many correlations are missing due to the corresponding variants having average Answer Relevance scores of 1, thereby making correlation calculations impossible.

In total, 6 of the 10 metric variants report average relevance values of 1, indicating that they assess every answer in the dataset sample as fully relevant to its respective question.

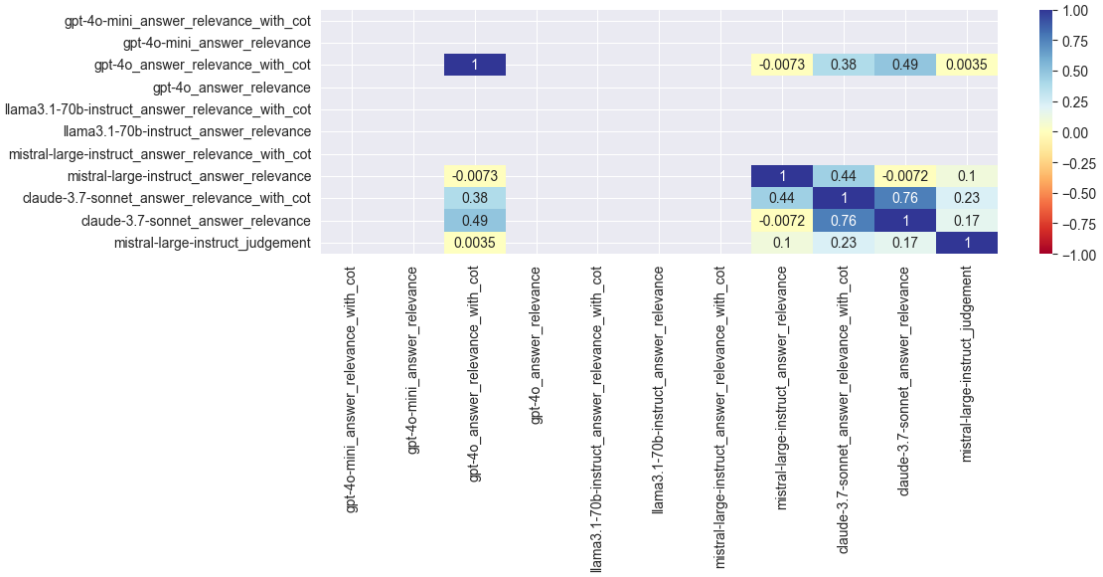

Figure 4.25.: Correlations of Answer Relevance variants on the WikiEval dataset

While this suggests that Document Chat's RAG pipeline reliably generates relevant responses, it also complicates evaluation, as variability across responses is essential for meaningful comparisons between configurations. This also confirms the earlier assumption that the LLMs overestimate Groundedness and that the respective low correlation is not due to lacking relevance of the answers.

Of the remaining variants, Claude-3.7-Sonnet with CoT has the highest correlation with the reference-guided LLM judgement baseline at 0.23. As with Answer Relevance, these best-performing configurations are then applied to the full datasets.

On the sovanta dataset, Mistral-Large without CoT yields a **correlation of 0.16 with the baseline**. This correlation increases to 0.26 when only answerable questions are considered, confirming that evaluating unanswerable questions is an inherent problem of the Answer Relevance metric. On the WikiEval dataset, Claude-3.7-Sonnet with CoT achieves a **correlation of 0.32 with the baseline**.

As with Groundedness, the Answer Relevance metric only captures a single aspect of overall answer quality and therefore, strong correlations with the holistic LLM judgement baseline are not expected. Nonetheless, the results indicate that Answer Relevance exhibits stronger alignment with the baseline compared to Groundedness.

In the next section, we analyze how the metric performs when it is integrated with Groundedness and Context Relevance to form the RAG Triad.

4.2.5. RAG Triad

As previously discussed in Section 2.2.2, although the RAG Triad is a concept that has now been frequently mentioned in the literature, there is a notable absence of standard-

ized implementation guidelines. Furthermore, there is a complete lack of evaluations of the metric against other evaluation metrics, especially against those based on ground truth.

To our knowledge, no existing description of the RAG Triad specifies how its individual components should be aggregated into a unified metric. For this reason, we perform such an evaluation on our data.

This evaluation is based on the previously computed variants of Groundedness and Answer Relevance on the sampled datasets, as well as the best-performing variants applied to the full datasets. For Context Relevance, only the best-performing variant from Section 4.1.3 is used, as its computation is based on a different dataset subset designed for retrieval evaluation and is therefore not directly comparable to the other samples.

A grid search is conducted to evaluate the performance of various configurations of the RAG Triad, varying both the metric variants and their respective weightings. Weighting schemes include equal distribution, disproportionate emphasis (e.g., 50% on one component and 25% on the others), and exclusion of individual components. The objective is to identify the configuration that exhibits the highest correlation with the reference-guided LLM judgement baseline.

sovanta Dataset

On the sovanta dataset, unanswerable questions pose two specific challenges for the RAG Triad: First, Answer Relevance inherently assigns low scores to the abstentions that are expected for these questions. Second, Context Relevance is irrelevant for these metrics, as the retrieved context is never relevant to the query. However, unanswerable questions must be treated identically in the metric calculation to preserve the reference-free nature of the metric.

For this reason, we analyze the results on both the full set of questions, as well as only on answerable questions. Table 4.4 shows the correlations of the five best-performing configurations on the full subset of the sovanta data and Table 4.5 shows the same data on answerable questions only.

The data shows that the best-performing variants of Groundedness and Answer Relevance identified earlier are also the ones used to produce the best RAG Triad scores and that the same variants perform best on both selections of questions. Notably, the highest correlations are achieved when Context Relevance is excluded from the calculation.

Furthermore, the data shows much higher correlations of the RAG Triad with the reference-based baseline when only answerable questions are considered. This underlines the earlier assumption that the RAG Triad is not suited for unanswerable questions. In fact, no configuration of the metric leads to a positive correlation when only unanswerable questions are considered.

Since these results validate the selection of variants for Answer Relevance and Groundedness, a similar grid search is conducted on the full sovanta dataset to determine the final optimal weighting and the overall correlation of the RAG Triad with the baseline. In this grid search, the highest-performing metric is **50% Mistral-Large Answer Rel-**

Table 4.4.: The five RAG Triad combinations with the highest correlation with the baseline on the sovanta dataset subset for all types of questions

Answer Relevance	Groundedness	Context Relevance	Weights	r
Mistral-Large	Llama3.1	Llama3.1 (with CoT)	(0.25, 0.75, 0)	0.169
Mistral-Large (with CoT)	Llama3.1	Llama3.1 (with CoT)	(0.25, 0.75, 0)	0.164
Mistral-Large	Llama3.1 (Answerability + Trivial Filtering)	Llama3.1 (with CoT)	(0.25, 0.75, 0)	0.153
Mistral-Large (with CoT)	Llama3.1 (Answerability + Trivial Filtering)	Llama3.1 (with CoT)	(0.25, 0.75, 0)	0.152
Llama3.1 (with CoT)	Llama3.1	Llama3.1 (with CoT)	(0.25, 0.75, 0)	0.143

Table 4.5.: The five RAG Triad combinations with the highest correlation with the baseline on the sovanta dataset subset for answerable questions only

Answer Relevance	Groundedness	Context Relevance	Weights	r
Mistral-Large	Llama3.1	Llama3.1 (with CoT)	(0.25, 0.75, 0)	0.312
Mistral-Large	Llama3.1	Llama3.1 (with CoT)	(0.5, 0.5, 0)	0.305
Mistral-Large (with CoT)	Llama3.1	Llama3.1 (with CoT)	(0.25, 0.75, 0)	0.301
Llama3.1 (with CoT)	Llama3.1	Llama3.1 (with CoT)	(0.25, 0.75, 0)	0.281
Llama3.1	Llama3.1	Llama3.1 (with CoT)	(0.25, 0.75, 0)	0.277

evance, 25% Llama3.1 Groundedness, and 25% Llama3.1 Context Relevance (with CoT), yielding a correlation with the baseline of 0.185. When unanswerable questions are ignored, this metric configuration is still the best-performing with an **increased correlation of 0.294.**

These results indicate that combining the individual metrics into the RAG Triad leads to improved alignment with the reference-guided baseline, thereby supporting the overall validity of the composite approach. Nonetheless, the results also confirm the earlier observation that the RAG Triad is unsuited for questions that are not answerable based on the context.

WikiEval Dataset

The evaluation of the RAG Triad metric on the WikiEval data is more straightforward, since the dataset contains no unanswerable questions. Table 4.6 presents the five best-performing configurations on the subset of the WikiEval data. Again, the individual metrics that have the highest correlation with the baseline on their own are also used to compute the highest-correlating RAG Triad scores.

Subsequently, we again perform the grid search for optimal weights on the entire dataset. The highest correlation is achieved at **0.2613 with 50% Claude-3.7-Sonnet Answer Relevance (with CoT), 25% Claude-3.7-Sonnet Groundedness (Triv-**

Table 4.6.: The five RAG Triad combinations with the highest correlation with the baseline on the WikiEval dataset subset

Answer Relevance	Groundedness	Context Relevance	Weights	r
Claude-3.7-Sonnet (with CoT)	Claude-3.7-Sonnet (Trivial Filtering)	GPT-4o (with CoT)	(0.5, 0.25, 0.25)	0.290
Claude-3.7-Sonnet (with CoT)	Claude-3.7-Sonnet (Trivial Filtering)	GPT-4o (with CoT)	(0.33, 0.33, 0.33)	0.282
Claude-3.7-Sonnet (with CoT)	Claude-3.7-Sonnet (Trivial Filtering)	GPT-4o (with CoT)	(0.25, 0.25, 0.5)	0.270
Claude-3.7-Sonnet	Claude-3.7-Sonnet (Trivial Filtering)	GPT-4o (with CoT)	(0.25, 0.25, 0.5)	0.265
Claude-3.7-Sonnet	Claude-3.7-Sonnet (Trivial Filtering)	GPT-4o (with CoT)	(0.33, 0.33, 0.33)	0.241

ial Filtering), and 25% GPT-4o Context Relevance (with CoT). This means that the best-performing RAG Triad metric has a lower correlation with reference-guided judgement than the standalone Answer Relevance metric.

Overall, the evaluation shows that the optimal configuration of the RAG Triad metric and its effectiveness depend on the characteristics of the dataset, particularly the presence or absence of unanswerable questions. While the RAG Triad outperforms individual components on the sovanta dataset, it does not surpass the performance of the standalone Answer Relevance metric on WikiEval. The optimal weighting of the metrics' components is the same on both datasets at 50% Answer Relevance, 25% Groundedness and 25% Context Relevance.

4.2.6. Overall Results

This section presents the overall performance of all evaluated metrics on both datasets, with a particular focus on their alignment with the reference-guided LLM judgement baseline. Specifically, the results of BLEU, ROUGE, BERTScore, and the reference-free LLM judge are also presented here, as these metrics require no hyperparameter tuning and are applied directly to the full datasets.

As before, we use the standard models proposed in the original work to calculate BERTScore, i.e., RoBERTa-Large for English queries and BERT-cased-multilingual for German queries (Zhang et al. 2020).

sovanta Dataset

Figure 4.26 presents the correlation matrix for all metrics on the full sovanta dataset. ROUGE-1 demonstrates the highest correlation with the baseline at 0.53, followed by ROUGE-L (r=0.49), ROUGE-2 (r=0.47), BLEU (r=0.45), and BERTScore-Recall (r=0.4). This means that these more traditional metrics all perform relatively similarly,

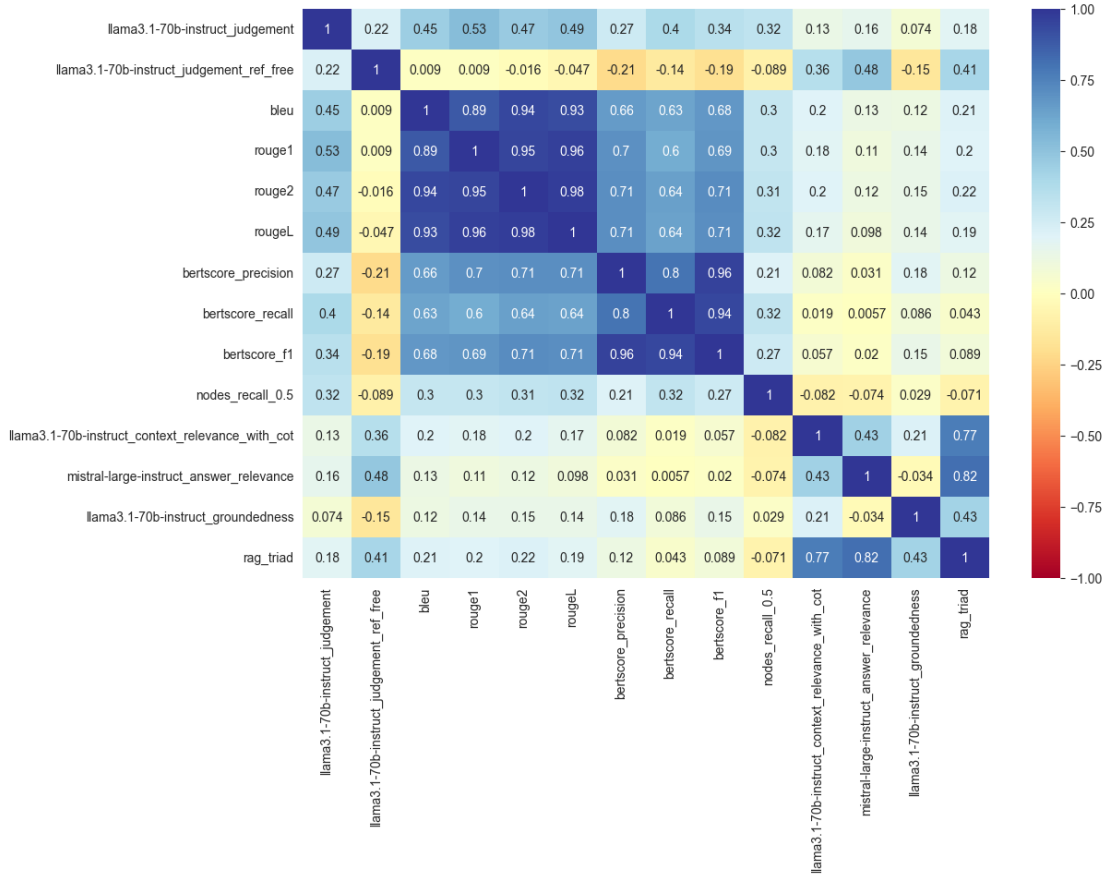

Figure 4.26.: Correlations of all metrics with each other on the sovanta dataset

which is further reflected in their strong inter-metric correlations, particularly between BLEU and ROUGE. This is expected, given that both are n-gram-based.

For metrics without ground truth, reference-free LLM judgement has the highest correlation with the baseline (r=0.22), followed by the RAG Triad (r=0.18), which has higher correlations than its individual components.

As discussed before, unanswerable questions present a significant challenge in this dataset. Therefore, Figure 4.27 presents the same results on answerable questions only. In this setting, the correlation of the RAG Triad with the baseline increases to 0.29, while the correlations for all other metrics remain very similar. This suggests that the RAG Triad in particular fails to effectively evaluate unanswerable instances.

Notably, retrieval recall achieves a moderate correlation of 0.49 with the baseline, which is second only to ROUGE-1. This means that for answerable questions, recall is a more reliable proxy for their quality than the other metrics.

Since all metrics only give moderate correlations with the baseline, we want to check whether the results that these metrics give in terms of optimal hyperparameter configu-

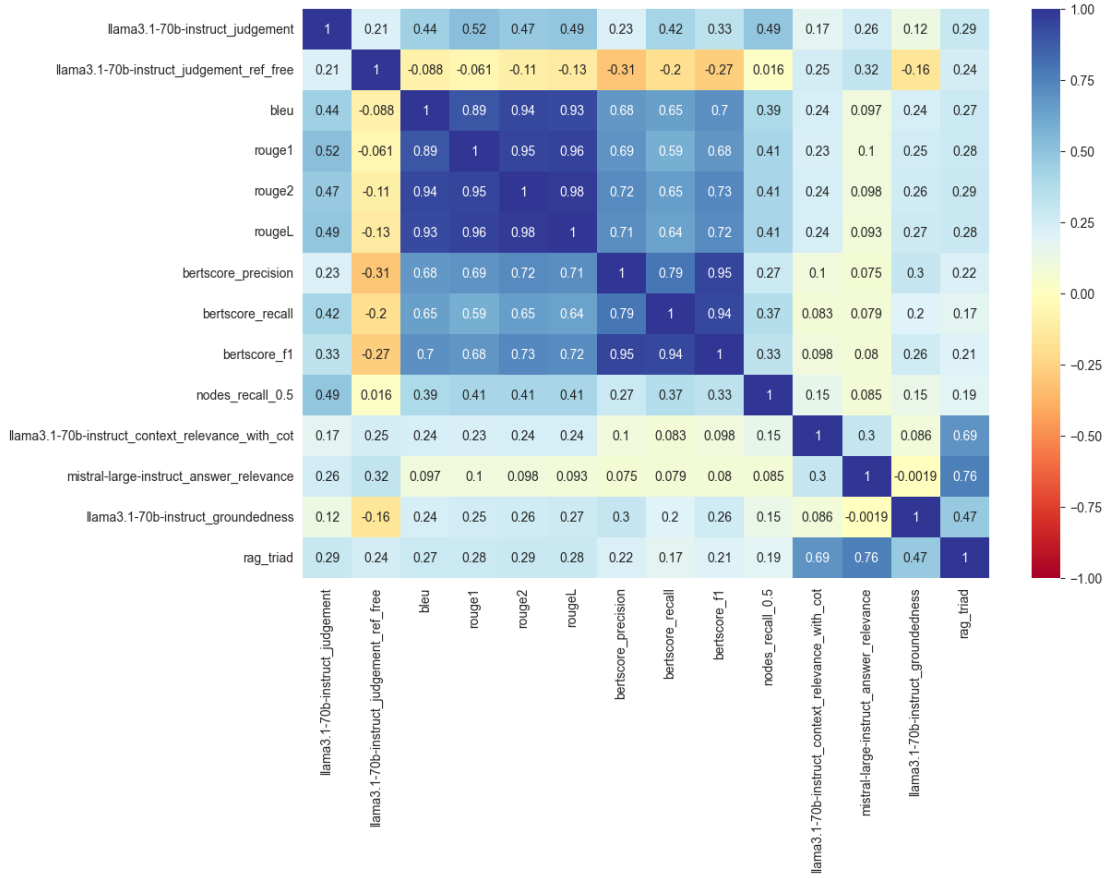

Figure 4.27.: Correlations of all metrics with each other on the sovanta dataset with unanswerable questions excluded

rations are still similar to the baseline by analyzing the highest-performing reference-free and reference-guided metrics, respectively.

ROUGE-1 is the best-performing reference-based metric and Figure 4.28 shows its results per LLM. Similar to the baseline results given in Section 4.2.1, Mistral-Large (rouge1=0.428) outperforms Llama3.1 (rouge1=0.414), albeit with a smaller margin. Also, the correlation between context size and answer quality is less pronounced in this case, indicating that ROUGE fails to effectively capture correct answers for longer context sizes. Section 4.2.8 shows that especially Mistral-Large tends to generate longer responses for longer input context, potentially negatively impacting n-gram overlap measurements used by ROUGE.

Similarly to the baseline, ROUGE-1 also observes a slight increase in answer quality when using the BGE-Reranker (0.422 versus 0.420). The ranking of the answer quality across clusters is different to the baseline, though CONTRACTS, HR, and TECH perform better than UNKNOWN, OTHER, and SUMMARY, which is similar to the baseline.

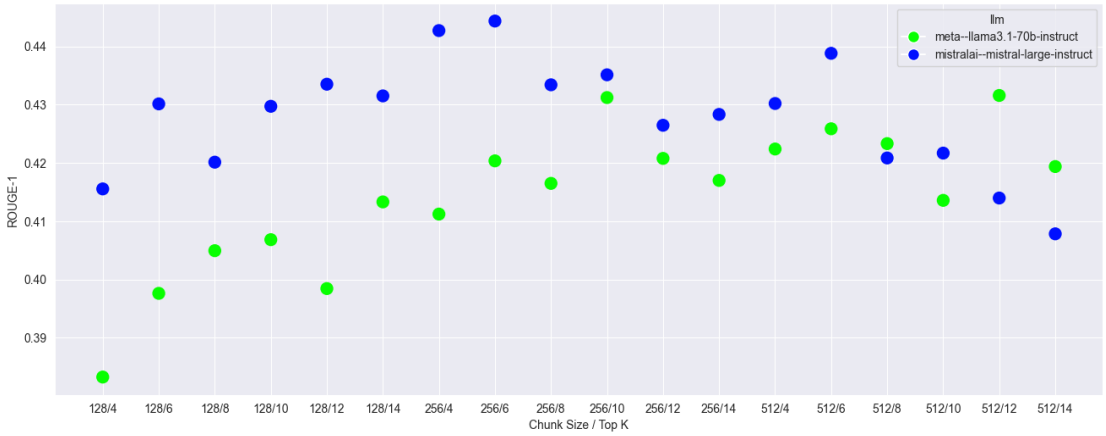

Figure 4.28.: Average ROUGE-1 score per LLM on the sovanta dataset

The RAG Triad on answerable questions only is the best-performing reference-free metric on this dataset. However, the metric assigns a higher average score to Llama3.1 (rag_triad=0.887) than to Mistral-Large (rag_triad=0.885), which contradicts the baseline findings. Also, no relationship between longer context sizes and better answers is visible. Nonetheless, the RAG Triad also shows a slight improvement when using the BGE-Reranker (0.887 versus 0.885) and the ranking of the clusters is similar to the baseline.

WikiEval Dataset

Figure 4.29 shows the correlation matrix for all metrics on the full WikiEval dataset.

BERTScore-Recall is the best-performing metric, achieving a correlation of 0.79 with the baseline. This is followed by ROUGE-1 (r=0.71), BLEU (r=0.68), and the other variants of BERTScore and ROUGE. As with the sovanta dataset, these metrics exhibit strong correlations with each other.

Examining the results of BERTScore-Recall in detail in Figure 4.30, the strong alignment with the baseline given in Section 4.2.1 is further evidenced by the similarity in model rankings. Again, GPT-4o and Mistral-Large perform very similarly, with both outperforming Llama3.1 and Gemini-2.0-Flash.

Larger context sizes generally result in slightly better results and the BGE-Reranker (bert=0.9275) slightly outperforms using no reranking (bert=0.9266).

In terms of metrics without ground truth, reference-free LLM judgement has the highest correlation with the baseline at 0.53, followed by Answer Relevance (r=0.32) and the RAG Triad (r=0.26). Figure 4.31 shows the results of reference-free LLM judgement in detail. It produces the same ranking of the four LLMs as observed in the baseline, though the relationship between context size and answer quality is not visible here. Still, the performance gap between the two reranking options is identified by the metric in the same way as in the baseline.

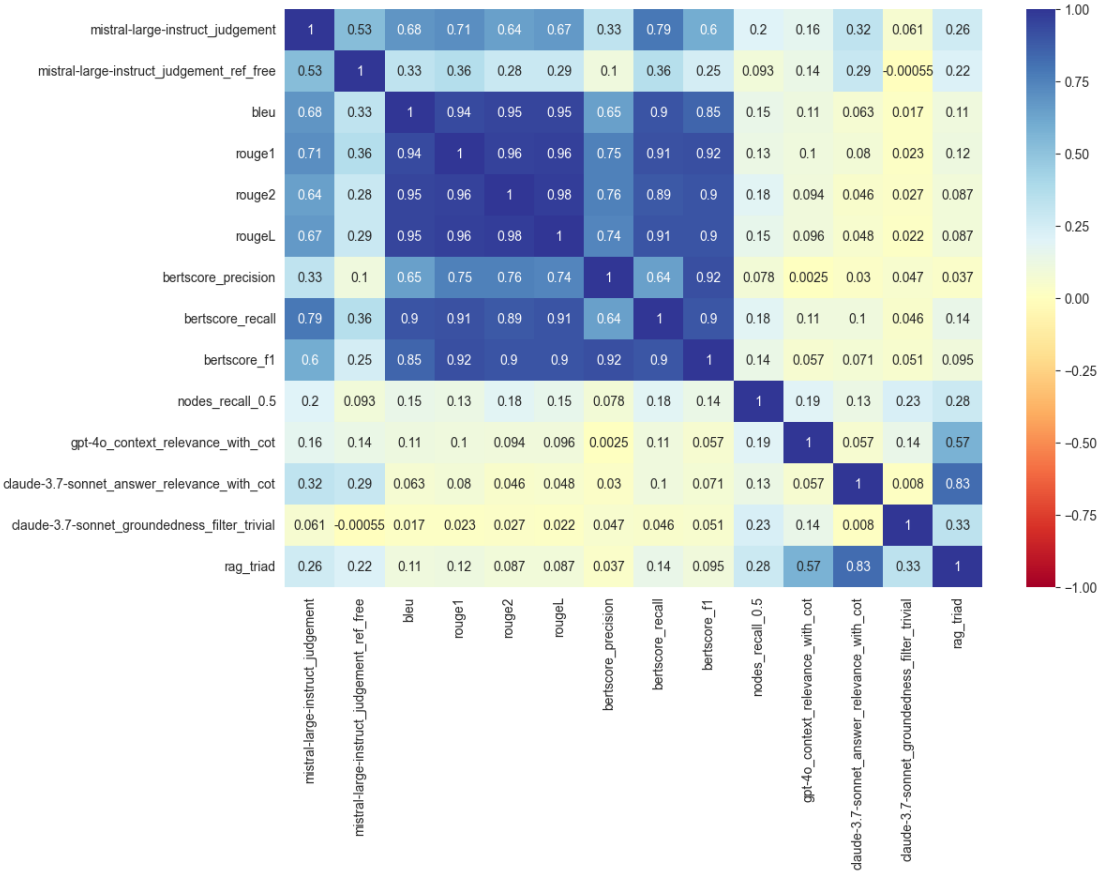

Figure 4.29.: Correlations of all metrics with each other on the WikiEval dataset

Summary

Overall, the ability of the different metrics to replicate the reference-guided LLM judgement baseline depends on the dataset. In the case of the sovanta dataset, traditional metrics like BLEU, ROUGE, and BERTScore, which are reference-based but not LLM-based, achieve moderate correlations of around 0.5. ROUGE in particular gives similar results as the baseline in terms of which LLMs and rerank models perform best in the RAG pipeline. However, no metric reliably evaluates the differences in answer quality based on different context sizes.

In contrast, all reference-free metrics only achieve weak correlations with the baseline. The RAG Triad is still the best of these metrics, but it produces rankings for LLMs and context sizes that are in contrast to the baseline. Furthermore, the RAG Triad does not work on unanswerable questions.

On the WikiEval dataset, BLEU, ROUGE, and BERTScore achieve strong correlations with the baseline. Especially BERTScore-Recall accurately replicates the results from the baseline, identifying the same optimal hyperparameter combinations. These results

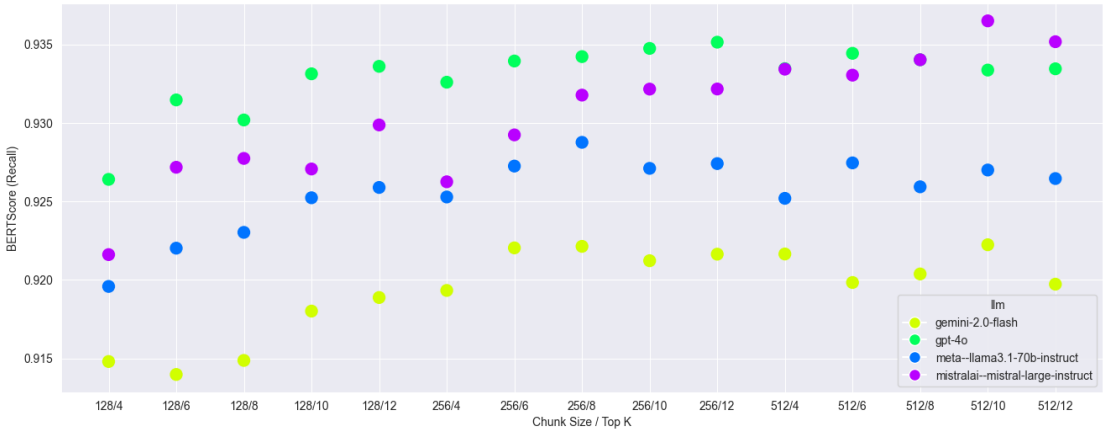

Figure 4.30.: Average BERTScore-Recall per LLM on the WikiEval dataset

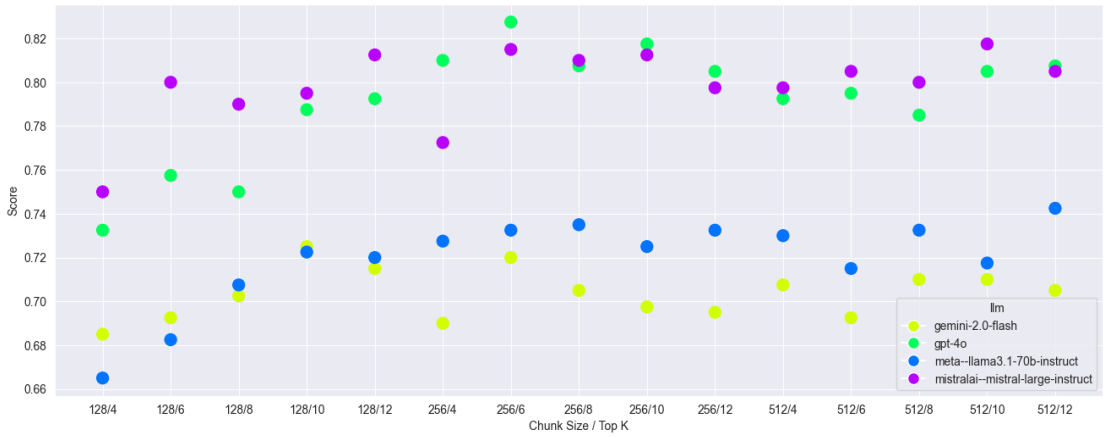

Figure 4.31.: Average reference-free LLM judgement per LLM on the WikiEval dataset

suggest that BERTScore could serve as an effective alternative to the computationally intensive LLM-based baseline in this case.

Also, the reference-free LLM judge still achieves a moderate correlation with the baseline on this dataset and outputs the same classification for the LLMs and reranking options. The improved performance of the metric on this dataset may be attributable to the less complex nature of the questions that has been discussed before, allowing a more accurate evaluation of the answers without a baseline. However, the RAG Triad still only yields a weak correlation with the baseline on this dataset.

4.2.7. Error Analysis

This section provides a detailed error analysis of the evaluated generation metrics. The analysis follows the same approach outlined in Section 4.1.4 for retrieval.

Specifically, the errors incurred by the traditional metrics such as BLEU and ROUGE, the reference-free LLM judge, and the RAG Triad are analyzed separately in terms of error classes and their distributions. Again, the final prompts used for each classification are given in the appendix.

BLEU, ROUGE, BERTScore

Since the scores for BLEU, ROUGE, and BERTScore are highly correlated with each other on both datasets, we analyze the errors on the average of the metrics for each row to streamline the analysis. Specifically, the average of BLEU, ROUGE-1, and BERTScore-Recall is taken, as these are the best-performing variants.

On the sovanta dataset, 1,346 of the 7,992 evaluated generation data points incur an absolute difference of more than 0.5 between LLM judgement and the average of the metrics. In nearly all of these instances, the LLM judgement assigns a score of 1, meaning that the evaluated metrics underestimate the accuracy of the answer. A similar pattern is observed on the WikiEval dataset, where 115 of 6,000 cases fall into the error classification.

On both datasets, the errors appear throughout context sizes, but are slightly more prevalent on larger context sizes. Notably, 80% of the errors on the WikiEval dataset are created by just two distinct prompts. Here is an example of such an error:

> **Prompt:** What are some measures for pandemic prevention?
> **Gold Answer:** Some measures for pandemic prevention include reducing the causes of new infectious diseases, preventing outbreaks and epidemics from becoming pandemics, and implementing active surveillance for early detection of cases. Additionally, robust public health systems, global infrastructure, and effective treatment measures are important in preventing the spread of pandemics. Biosafety technologies and biotechnology regulation, such as laboratory containment technologies and strengthening laboratory and field work protocols, are also crucial in preventing the emergence and spread of infectious diseases.
> **Actual Answer:** Some measures for pandemic prevention include: * Robust, collaborating public health systems that have the capacity for active surveillance for early detection of cases and to mobilize their health care coordination capacity. * A good global infrastructure, consequent information exchange, minimal delays due to bureaucracy and effective, targeted treatment measures. * Considering pandemic prevention as an aspect of international development in terms of health-care infrastructure and changes to the pathogen-related dynamics between humans and their environment including animals. * Public health systems "in each country" need to be capable of detecting contagion early, diagnosing it accurately, implementing effective disease control measures, and fully collaborating with the relevant international authorities at each stage. * Potential policies that support global biosafety could make use of various technologies, including but not limited

to laboratory containment technologies. * Limiting research on highly contagious biological agents to only trained researchers in well-protected environments and advanced biological safety systems and disposal of biohazards. * Improving physical security and educating scientists about the misuse potentials * Review processes that ensure risks are justified and minimized, such as preventing certain gain-of-function studies. * Monitoring and strengthening laboratory protocols around the world. * Monitoring and strengthening field work protocols around the world (such as viral sampling).
LLM Judgement (Baseline): 1
Average of BLEU, ROUGE-1, and BERTScore-Recall: 0.44

Although the answer contains all of the information from the gold answer, it is much more elaborate and includes additional information not found in the gold answer. This poses a particular challenge for n-gram-based metrics. In fact, early analysis already shows that cases like this represent the vast majority of errors, especially on the WikiEval data. Overall, we identify one more type of error, leading to these two error classes:

1. Over-Elaboration: The model answer includes extensive elaboration, giving a lot more information than the gold answer, but with the information from the gold answer still present.

2. Lexical Variation (same meaning, different words): The model answer expresses the same meaning as the gold answer using different words, phrasing, or sentence structure. The metrics fail to capture the semantic equivalence.

As in the retrieval error analysis, the LLM may also output *OTHER* as a classification category.

On the sovanta dataset, both Mistral-Large and Llama3.1 achieve an accuracy of 0.66 when compared to human annotations. Again, we use Mistral-Large for the final estimation. On the WikiEval dataset, three LLMs achieve the same accuracy of 0.63, we select GPT-4o. Figure 4.32 illustrates the distribution of error classes across both datasets.

As expected, almost all errors in the WikiEval dataset belong to error class one. This is in line with the fact that only two prompts are responsible for 80% of errors and that there is only a low number of errors present in the dataset for these metrics. It also explains why the errors are slightly more prevalent for higher context sizes, as these tend to create longer answers (see Section 4.2.8).

In contrast, the sovanta dataset contains a significant number of prompts where lexical variation leads to an underestimation of the answer accuracy. For example, the prompt *Wie heißen die Squads und wer sind die Leads?* with gold answer *Die Squads heißen "Automation & Analytics", "Core AI & GenAI" und "Integration & Data". Die Leads sind Larissa Haas für "Automation & Analytics", Niklas Fruehauf für "Core AI & GenAI" und Philip Weißmann für "Integration & Data".* and system answer *Die Squads sind: 1. Automation & Analytics - Squad Lead: Larissa Haas; 2. Core AI & GenAI -*

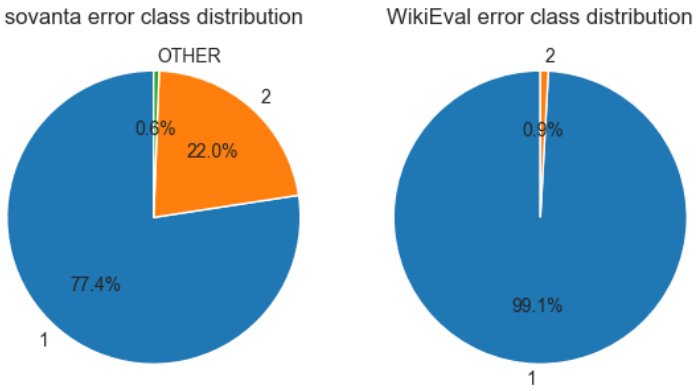

Figure 4.32.: BLEU / ROUGE / BERTScore error class distribution

Squad Lead: Niklas Fruehauf; 3. Integration & Data - Squad Lead: Philip Weißmann shows how differently phrased answers lead to lower scores for these metrics, although their meaning is exactly the same.

This analysis confirms that, as anticipated, metrics such as BLEU and ROUGE struggle to accurately evaluate responses that exhibit different phrasing or contain additional information. This reflects an inherent limitation of these metrics. However, it has to be noted that the accuracy of the LLMs evaluating the errors is relatively modest, showing a somewhat limited ability to accurately identify the error classes in this case. Since the accuracy is already limited, splitting up the existing error classes into more granular classes would reduce the accuracy even further and is therefore not feasible here.

Reference-free LLM judgement

This section applies the same error analysis methodology to the reference-free LLM judgement metric. On the sovanta dataset, 824 out of 7,992 evaluated queries exhibit an absolute score difference greater than 0.5 between the baseline and the reference-free LLM judgement. As in the previous analysis, these cases typically feature a baseline score of 1, while the reference-free metric assigns a lower score, thereby underestimating the answer quality. The errors are distributed equally across different hyperparameter configurations. All of these characteristics also hold for the WikiEval dataset, where 149 of 6,000 queries are above the threshold.

However, on the sovanta dataset, 40% of errors belong to the cluster UNKNOWN. This is expected, as the reference-free metric has no notion of a correct answer in this case and inherently assigns low values.

Notably, two distinct prompts again account for 72% of the errors on the WikiEval dataset. Here is an example of an error with respect to one of these prompts:

> **Prompt:** When was the 5th Separate Guards Tatsin Red Banner Order of Suvorov Tank Brigade formed, and what is its military unit number?

Gold Answer: The 5th Separate Guards Tatsin Red Banner Order of Suvorov Tank Brigade was formed in 2009. Its military unit number is 46108.
Actual Answer: The 5th Separate Guards Tatsin Red Banner Order of Suvorov Tank Brigade was formed in 2009 and its military unit number is 46108.
LLM Judgement (Baseline): 1
Reference-free LLM Judgement: 0.5

The example demonstrates that, despite the system answer being nearly identical to the gold reference, the reference-free metric assigns an unjustifiably low score. A possible explanation is that the LLM deems the answer too brief, implicitly expecting additional elaboration, even though the response fully satisfies the prompt.

In other cases, the language of the system-generated answer diverges from the prompt language, which leads to a low reference-free-based score, while the reference-based judgement still gives a high score due to semantic similarity with the gold answer. Overall, we identify the following four error classes:

1. Language Diversion: The model answer is semantically correct, but it uses a different language than the gold answer.
2. Overly Short Answer: The model answer closely resembles the gold answer, but it is very short or minimal, leading the reference-free method to underestimate its correctness due to lack of elaboration or context.
3. Missing Information: The answer partially answers the question, but certain information from the gold answer is missing.
4. "I Don't Know" is the correct answer: The gold answer indicates that the information is not present in the context. Reference-free scoring fails to recognize this as correct due to lack of supporting context.

Mistral-Large achieves the highest agreement with human annotations on the sovanta dataset, with an accuracy of 0.63. The same accuracy is achieved by GPT-4o on the WikiEval dataset. Figure 4.33 shows the error class distribution on both datasets.

As expected, the primary source of error in the sovanta dataset arises from unanswerable questions. This is followed by overly short answers, which is also the dominant error class on the WikiEval dataset. Missing information is a more frequent error class on the sovanta dataset, which makes sense because as described before, the retrieval accuracy is generally lower on this dataset. Language diversion makes up the rest of the cases across both datasets. Notably, 13% of classifications incurred an error on the WikiEval dataset due to content filters of GPT-4o.

The analysis reveals that the reference-free LLM judgement especially struggles with short yet correct answers, which it frequently misclassifies as incomplete. Furthermore, the metric exhibits limitations in handling unanswerable prompts and partially correct responses. As with the previous error analysis, the accuracy of the LLM-based error classification remains an area for improvement.

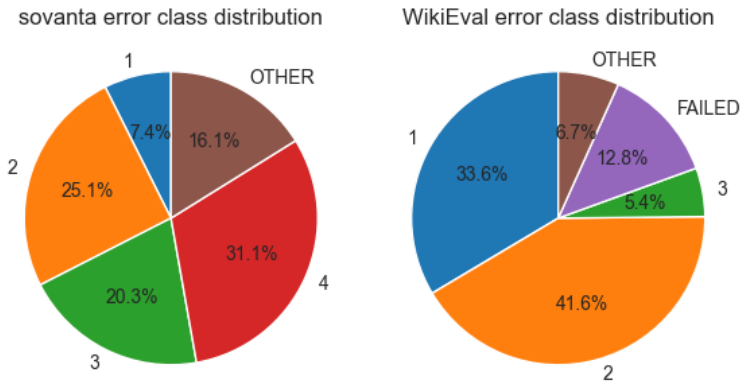

Figure 4.33.: Reference-free LLM judgement error class distribution

RAG Triad

This section concludes the error analysis by examining the RAG Triad metric. On the sovanta dataset, 1,058 of the 7,992 evaluated queries exceed the error threshold. Errors are distributed across context sizes, but occur slightly more frequently with smaller context sizes. A similar, but more pronounced pattern is observed on the WikiEval dataset. Of the 78 WikiEval queries exceeding the error threshold, 50 originate from the smallest chunk size.

Across both datasets, most errors occur when the RAG Triad assigns a score close to 1, while the baseline score is significantly lower, indicating that, unlike the other metrics, the RAG Triad systematically overestimates answer quality. Here is an example from the WikiEval dataset:

> **Prompt:** When was the Roanoke and Tar River Railroad fully merged into the Seaboard Air Line Railway network, and what happened to the line after that?
> **Gold Answer:** The Roanoke and Tar River Railroad was fully merged into the Seaboard Air Line Railway network in 1911. After the merger, the line operated as the Seaboard Air Line's Lewiston Subdivision. In 1967, the Seaboard Air Line merged with the Atlantic Coast Line Railroad to form the Seaboard Coast Line Railroad. The Lewiston Subdivision connected with an ex-ACL line in Kelford. In 1980, the Seaboard Coast Line's parent company merged with the Chessie System to create the CSX Corporation. The CSX Corporation initially operated the Chessie and Seaboard Systems separately until 1986, when they were merged into CSX Transportation. The southernmost five miles of the line from Kelford to Lewiston was abandoned sometime after 1986. The remaining line from Boykins to Kelford was sold to the North Carolina and Virginia Railroad in 1987, who operates the line today.

Actual Answer: The Roanoke and Tar River Railroad was fully merged into the Seaboard Air Line in 1911. The line would operate as the Seaboard Air Line's Lewiston Subdivision.
LLM Judgement (Baseline): 0.25
RAG Triad: 1

This example illustrates that when the gold answer is elaborate and highly detailed, a shorter yet technically correct response receives a substantially lower score under the baseline metric than under the RAG Triad. This issue is particularly pronounced on the WikiEval dataset, which contains numerous gold answers that are significantly more elaborate than the responses typically produced by LLMs. This highlights an area for further improvement on the dataset.

Another common error type involves answers that are topically related to the prompt but omit the specific information required by the gold answer. This phenomenon is particularly prevalent on the sovanta dataset, where key terminology frequently appears in unrelated contexts, leading to misleading but plausible responses. For instance, the prompt *What is Experience Management?* has a gold answer describing it in terms of what kind of discipline it is. However, a system response stating that *Experience Management is an area headed by...* receives a high RAG Triad score, despite lacking the core definitional content. In total, we observe the following three error classes:

1. Partially Correct Answer: The answer contains parts of the information from the gold answer, but certain information from the gold answer is missing.

2. Related Answer: The answer is topically related to the query and may seem plausible, but does not contain any of the information from the gold answer.

3. "I Don't Know" is the correct answer or question unclear: The gold answer indicates that the information is not present in the context or asks a follow-up question for clarification. Reference-free scoring fails to recognize this as correct due to lack of supporting context.

Mistral-Large yields the highest agreement with human annotations on the sovanta dataset at 0.6. Claude-3.7-Sonnet achieves an accuracy of 0.97 on the WikiEval dataset. Figure 4.34 presents the error class distribution on both datasets.

As anticipated, the majority of errors in the WikiEval dataset stem from partially correct answers. Notably, 66% of these errors are linked to only two prompts, both associated with unusually detailed gold answers.

Answers that are only related to the question but do not cover the information from the gold answer are exclusively present in the sovanta data. This is expected, as the WikiEval dataset has a much cleaner separation of topics, while the sovanta dataset has more overlaps.

Both of these error classes explain why the errors are more prevalent for short context sizes, as the likelihood of missing information is higher when less content is retrieved.

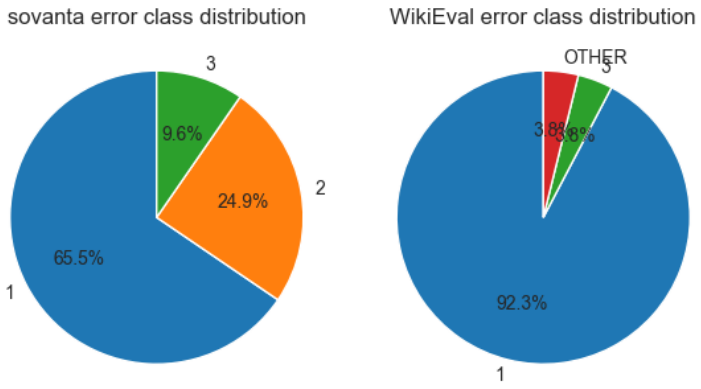

Figure 4.34.: RAG Triad error class distribution

Again, unanswerable questions are a source of errors in the sovanta dataset, though they represent a lower fraction than for previous metrics. The reason for this is likely that the Groundedness metric partially covers unanswerable questions, leading to fewer errors.

Table 4.7 provides a summary of the most important generation metrics and their error analysis.

4.2.8. Runtime and Token Usage Analysis

As with the retrieval evaluation, we again report on auxiliary performance metrics that result from the generation evaluation. Specifically, we analyze the inference time of the different LLMs as well as their token usages. Again, the values are presented as unified results across both datasets.

Figure 4.35 shows the average inference time per LLM, referring exclusively to the duration of the LLM inference step within the RAG pipeline.

Naturally, LLM inference time is influenced by the underlying hardware infrastructure. In the case of the SAP Generative AI Hub, which is used as the LLM provider in Document Chat, Mistral-Large and Llama3.1 are hosted in SAP data centers, while SAP acts as a proxy for GPT and Gemini models and forwards the requests to the respective providers (gen 2025).

The fastest LLM in our setup is Gemini-2.0-Flash with an average response time of only 1.3 seconds. This is expected, given that Gemini-2.0-Flash is the smallest model in the evaluated set. At an average response time of 1.9 seconds, GPT-4o is the second-fastest LLM. This is impressive, because GPT-4o is the largest LLM in the lineup.

Of the remaining two models, Mistral-Large (4.6s) is faster than Llama3.1-70b (6.4s). This is counterintuitive, as Mistral-Large is almost double the size of Llama3.1 (mis 2024; Dubey et al. 2024), but may be due to differences in resource allocation.

The figure also shows that for all models, the response time increases with increasing

Table 4.7.: Correlations of metrics with the baseline on sovanta and WikiEval datasets, along with error types and reasons. On the sovanta dataset, the two values represent the correlations on the full dataset and on answerable questions only.

Metric	Correlation with Baseline		Error Type	Error Reasons
	sovanta	WikiEval		
BLEU	0.45 / 0.44	0.68	Metric underestimates answer quality	Phrasing differences & additional information
ROUGE-1	0.53 / 0.52	0.71		
BERTScore-Recall	0.4 / 0.42	0.79		
Reference-free LLM judgement	0.22 / 0.21	0.53	Metric mostly underestimates answer quality	brief answers classified as incomplete
RAG Triad	0.18 / 0.29	0.26	Metric overestimates answer quality	Partially correct answers classified as complete

context size, though this increase is significantly more pronounced for Llama3.1 and Mistral-Large. To better understand the relationship between response time and context size, we examine the associated token usage.

Figure 4.36 shows the average prompt tokens per LLM. As expected, the prompt tokens are directly correlated with the context sizes, because each prompt is composed of system prompt, context, and user prompt.

However, there are slight differences between the LLMs. These differences are explained by the fact that the different LLMs use different tokenizers, which leads to a slight variation in the number of tokens. Nonetheless, the differences between the models are marginal.

In contrast, Figure 4.37 shows significant differences between the LLMs regarding the number of completion tokens. Specifically, Mistral-Large generates at least twice as many completion tokens as the other models. This also means that the number of tokens generated per second by Mistral-Large is similar to GPT-4o and Gemini-2.0-Flash.

In addition, the number of output tokens increases with the number of input tokens for all LLMs except Gemini-2.0-Flash and especially for Mistral-Large. This also explains the longer response times with increasing context sizes mentioned before.

In summary, Gemini-2.0-Flash and GPT-4o achieve the shortest average inference times, followed by Mistral-Large and Llama3.1. While most models produce responses of comparable length, Mistral-Large consistently generates approximately twice as many completion tokens per query.

We argue that in almost all RAG applications, the main determinant for the choice of LLM remains the answer quality. However, if LLMs have similar answer quality, response time and token usage can be guiding decision factors.

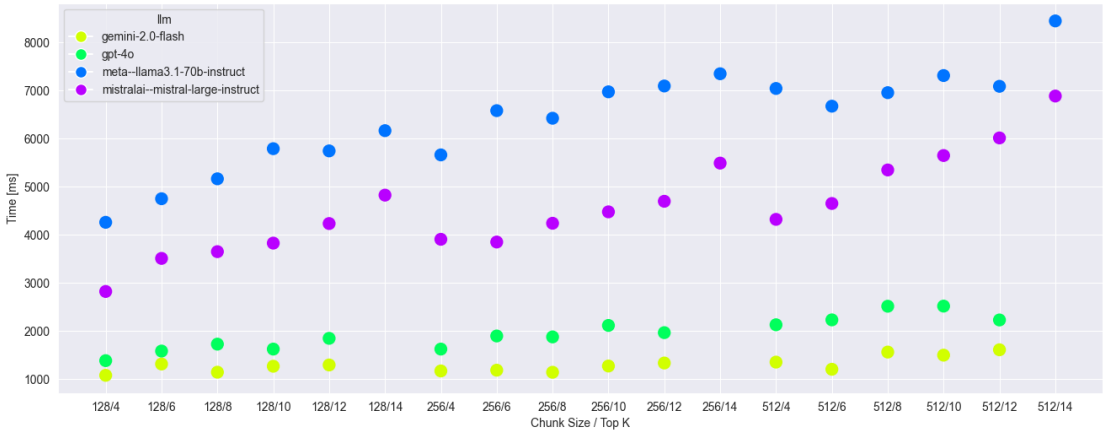

Figure 4.35.: Average inference time per LLM

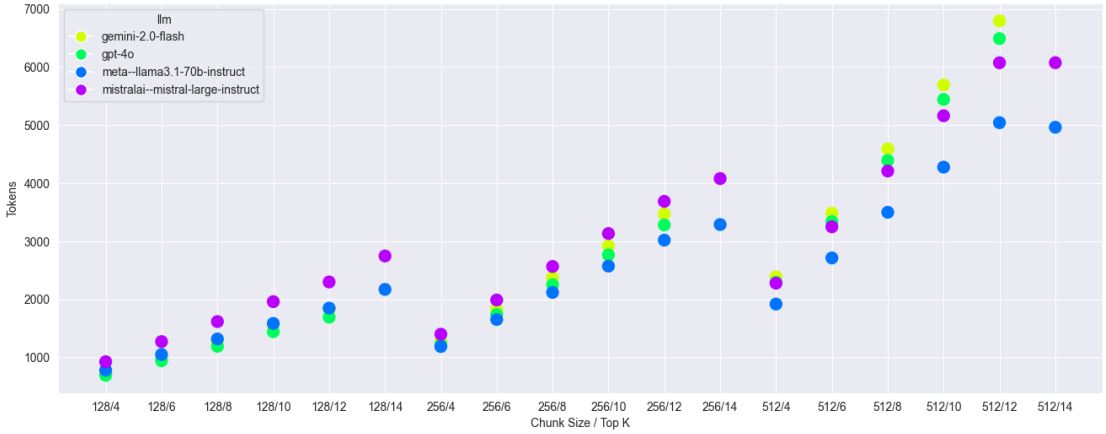

Figure 4.36.: Average prompt tokens per LLM

4.2.9. Results from User Feedback

This section concludes the results by presenting the user feedback collected through the side-by-side feedback feature in Document Chat. In total, 150 votes were collected from 23 unique users across six months. As described before, the testable parameters were the LLM used and `top_k`.

Two evaluation metrics are used to derive rankings for the tested parameters: win rate and Elo score. Win rate describes the proportion of cases where a given hyperparameter option was selected when it was among the options. We also compute confidence intervals for the win rate.

Elo was introduced in 1967 as a rating system for chess players (Elo 1967), but it has now also been applied to ranking LLMs (Boubdir et al. 2024). The idea of Elo is to update the ratings of players (or LLMs) based on the outcome of a duel and the rating of

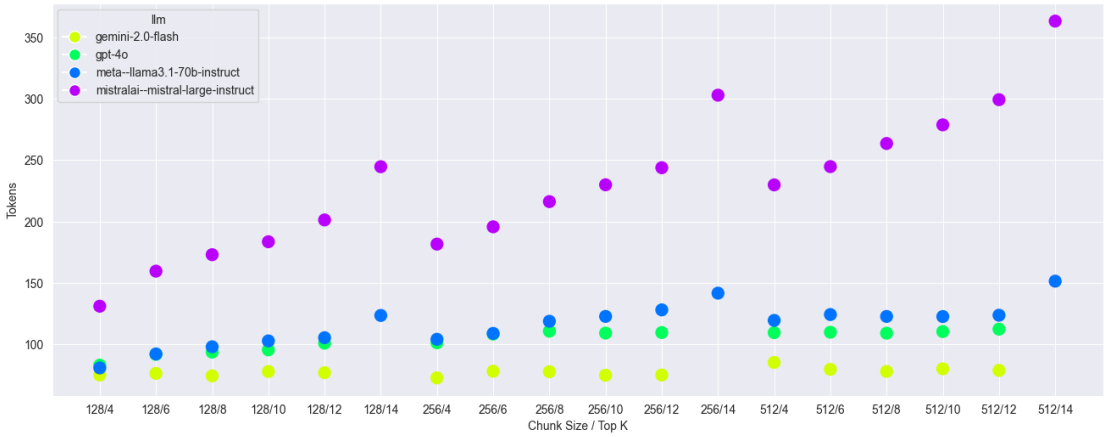

Figure 4.37.: Average completion tokens per LLM

Table 4.8.: Feedback results for LLMs

LLM	Win Rate (90% Confidence Interval)	Elo
Mistral-Small-Instruct	0.57 (0.45, 0.68)	1527
Mistral-Large-Instruct	0.54 (0.45, 0.63)	1512
Llama3.1-70b-Instruct	0.40 (0.30, 0.50)	1460

the opponent (Elo 1967). In our evaluation, we use an initial Elo of 1,500 and a K-factor of 32, which is standard for such evaluations (Boubdir et al. 2024). Elo ranking is also used in LMArena (Chiang et al. 2024).

Table 4.8 presents the results of the different LLMs. Mistral-Small slightly outperforms Mistral-Large in terms of win rate and Elo, while both models perform significantly better than Llama3.1.

These results are consistent with current Elo scores reported in LMArena, where Mistral-Small, Mistral-Large, and Llama3.1-70b have ratings of 1,353, 1,313, and 1,294, respectively (lma 2025). Although the approach would certainly benefit from more votes and therefore smaller confidence intervals, this alignment supports the validity of our evaluation framework. Moreover, the observed ranking of Mistral-Large ahead of Llama3.1 aligns with the performance trends reported by our LLM judgement evaluations in Section 4.2.1.

It is important to note that user preferences may be influenced not only by intrinsic answer quality but also by additional factors such as latency. For example, although Mistral-Small is a much newer model than Mistral-Large, it is also much smaller (mis 2025) and is very slightly outperformed by Mistral-Large in the MMLU-Pro benchmark (Wang et al. 2024). Its comparatively better performance in both LMArena and our own study may therefore be due to its smaller size and therefore faster inference time.

Table 4.9 presents analogous results for the `top_k` parameter. As expected, higher context sizes generally lead to better answers. However, `top_k`=10 outperforms the

Table 4.9.: Feedback results for `top_k`

top_k	Win Rate (90% Confidence Interval)	Elo
10	0.71 (0.56, 0.83)	1592
14	0.54 (0.41, 0.66)	1582
12	0.51 (0.37, 0.67)	1493
8	0.50 (0.36, 0.64)	1477
6	0.37 (0.25, 0.50)	1431
4	0.41 (0.28, 0.55)	1424

values 12 and 14, indicating that increasing `top_k` does not always monotonically improve performance. This may be due to similar reasons outlined before: Users inadvertently or deliberately prefer answers that are faster, and Section 4.2.8 shows that higher values of `top_k` lead to longer inference times.

Overall, these findings reinforce our retrieval evaluation results, which indicate that increasing `top_k` generally enhances response quality, albeit with diminishing returns beyond a certain threshold.

5. Discussion

This section discusses the metrics and datasets used in this work and the results they produce. Specifically, it focuses on comparing these findings with the original information on metrics and datasets provided in the respective literature, constructing a comprehensive view of RAG evaluation.

Our work introduces a new dataset for RAG evaluation, derived from real-world user queries in a production RAG system. The dataset contains a wide variety of queries, such as specific questions on projects, summarization requests, and questions that cannot be answered based on the retrieved context. This is a valuable addition, as reduced hallucinations for questions that do not have answers are a commonly cited advantage of RAG systems, yet most evaluation datasets fail to account for this factor. Furthermore, the dataset covers everything from smaller to larger files in various formats to large-scale external knowledge bases, such as complete Confluence spaces.

Our dataset stands in contrast to many datasets that are used in RAG evaluation literature. The survey by Yu et al. shows that most RAG benchmarks use synthetic data (Yu et al. 2024). Many even use LLM-generated questions and gold answers, for example the RGB benchmark, where an LLM was used to create question-answer-context triplets based on news articles (Chen et al. 2024). Another example of this is the WikiEval dataset, which is also used in this work to compare results across two distinct datasets. It was also created by using an LLM to generate questions and answers based on selected Wikipedia articles (Es et al. 2023). There are also manually created datasets without LLMs, such as MT-Bench, but it lacks the ground-truth labels necessary for meaningful evaluations (Zheng et al. 2023).

These characteristics of commonly used datasets lead to certain problems when conducting evaluations, which we especially observe on the WikiEval dataset. As shown in our error analysis, it frequently contains gold answers that exceed typical LLM output lengths and contexts that are broader than necessary for retrieval. Furthermore, the dataset does not store the original Wikipedia pages used to generate the answers, which reduces the reproducibility of the results. For this reason, further considerations are required to design a meaningful retrieval task for the dataset, which we do in our evaluation.

These limitations are addressed in our dataset, as the data stems from real-world usage of a production RAG application. Furthermore, we store snapshots of all relevant files and external system data used as the basis for retrieval to ensure reliable reproducibility.

The importance of good dataset design is further underlined by our results, as we come to different conclusions for the different datasets. In terms of optimal hyperparameters, we show that the best-performing embedding models and LLMs are the same across datasets. However, the optimal choice for chunk size, `top_k`, and rerank model

depends on dataset characteristics such as language and vector index size. The sovanta dataset even shows that the optimal chunking size depends on the type of question, with summaries, for example, usually requiring larger chunk sizes. This means that general recommendations for the setup of RAG systems can be derived with generic datasets, but specialized datasets are key to defining optimal system configurations tailored to specific use cases.

The primary objective of most RAG evaluations is to identify the optimal pipeline configuration, producing a ranking of hyperparameter choices that influence system performance. However, most RAG benchmarks in the literature evaluate a single RAG system rather than comparing the output quality of multiple systems. This is particularly important for evaluating new metrics, as it is paramount that a metric produces similar recommendations in terms of RAG system design as its baseline.

Of the metrics that we discuss in this work, ARES is the only one that is used by its original authors to evaluate rankings of different RAG systems (Saad-Falcon et al. 2023). Zheng et al. use their metrics to compare different LLMs (Zheng et al. 2023), but all other metrics are evaluated against their baseline on a singular RAG system. In contrast, we use the two datasets in our setup to create diverse RAG systems that are based on five different hyperparameters that can be used to benchmark against each other. This setup yields 29,088 data points for retrieval and 13,992 for generation.

Another core contribution of this work is the detailed evaluation of various metrics for retrieval and generation on the two datasets, resulting in unified and comparable results that reveal the true performance of each metric. Before discussing these results in detail in the following two sections, we give some general observations here.

As mentioned before, many of the new LLM-based metrics that have been introduced recently lack comprehensive evaluations of different variants and comparisons with meaningful baseline metrics. This is especially true for the choice of LLM in these metrics. Our evaluation shows that the LLM used has a greater impact on the results of a metric than the metric's design itself. This is reflected in strong inter-model correlations across different metric types and is especially present on LLM judgement, Groundedness, and Context Relevance. The work by Zheng et al. does evaluate different LLMs in terms of judgement, but primarily presents results for GPT-4, with the other LLM options being relatively older models.

Notably, newer and larger LLMs do not always outperform smaller and older models when used as evaluators. Although both MMLU-Pro and LMArena rate Llama3.1 as substantially weaker than Mistral-Large (Wang et al. 2024; lma 2025), it outperforms Mistral-Large for judgement and Context Relevance on the sovanta dataset. These results highlight the necessity of carefully selecting the LLM when computing such metrics. Nonetheless, stronger LLMs still lead to better results in most cases, as shown by the fact that GPT-4o consistently outperforms GPT-4o-mini across all metrics. Furthermore, the effect of incorporating Chain-of-Thought reasoning depends on the metric and the LLM and it does not always improve the results.

Beyond the creation of a new dataset based on real data and the thorough evaluation of various metrics across diverse RAG pipelines, this work also shows that LLMs can be used to analyze erroneous behavior of LLM-based (and other) metrics. Since many LLM-based

metrics lack sufficient evaluations, they consequently also lack analyses that identify where these metrics work well and where they leave potential room for improvement.

We bridge this gap in literature by employing an LLM-based heuristic that is based on alignment with human labels on a small excerpt of erroneous cases for each metric and is then used to get an approximate distribution of error classes across the entire dataset. Although the accuracies of current LLMs in this task span from 0.6 to 0.97, showing that the approach does not yield perfect results in every case, we show that LLMs are generally able to classify such errors to help in analyzing and improving metrics.

In the following two sections, we discuss the retrieval and generation results separately in detail.

5.1. Retrieval

Our retrieval analysis examines variations in the embedding model, reranking model, chunk size, and `top_k` across the two selected datasets. The results show that across all datasets and clusters, BGE-M3 outperforms Multilingual-E5-Large and that more context generally yields better results. Furthermore, a powerful reranking model such as the BGE-Reranker improves retrieval recall by up to seven percent compared to no reranking.

However, more granular results for the different hyperparameters depend on the dataset and even the specific question type. For example, on the WikiEval dataset, recall approaches 1.0 once the context size reaches about 2,000 tokens, whereas on the sovanta dataset, recall continues improving until roughly 7,000 tokens. Cluster-level results show that larger context sizes are especially beneficial for summaries, while some other question types achieve high recall even with smaller contexts. These results further underline that use-case-specific considerations have to be made for optimal retrieval in RAG applications.

This also holds for the reranking model. Although the results show that the BGE-Reranker performs best, the performance of smaller reranking models depends on specific dataset characteristics. For example, the relative performance gain from using the strongest reranker is larger on the sovanta dataset, suggesting greater benefits for larger and more complex datasets. Furthermore, on the sovanta dataset, smaller reranking models do not outperform having no reranker, whereas on WikiEval they provide a clear improvement. Our analysis indicates that this difference stems from the large proportion of German queries in the sovanta dataset, which are not handled well by the smaller, monolingual rerankers.

After establishing these baseline results, we assess the performance of the LLM-based Context Relevance metric and how its results compare to that baseline. Because the metric has many possible implementations and can be computed with any LLM, we first evaluate several of its variations.

This analysis shows that incorporating Chain-of-Thought reasoning improves the metric's correlation with recall, and that GPT-4o and Llama3.1 perform best on the WikiEval and sovanta datasets, respectively. This is an important contribution of our study, as the

work by TruEra, which introduced these metric variations, lacks comparisons between them entirely (tru 2025). Our analysis also shows that the metric's accuracy depends heavily on its configuration: Correlations with recall range from -0.19 to 0.45 across variants. Furthermore, the correlations also depend on the different clusters in the sovanta dataset, with values ranging from -0.15 to 0.39. This underscores the critical importance of configuration choice.

Comparing these best-performing configurations to the baseline on the full datasets yields the following results: The correlation with recall is 0.231 on the sovanta dataset and 0.465 on WikiEval. Subsequently, we employ an LLM-based error analysis to determine the main error reasons, discovering that a significant portion of deviations is due to insufficient retrieval labels. Incorporating these findings results in improved correlation values of 0.368 and 0.525, respectively.

The error analysis also reveals that Context Relevance often misclassifies incomplete retrieval results as complete, which is to be expected to a certain degree from a metric without access to ground truth. These and other errors occur mostly with smaller context sizes, likely because lower context sizes have lower average recall, increasing the chance of overestimating relevance.

Additionally, Context Relevance only correctly reproduces the positive relationship between higher context sizes and better search results on the WikiEval dataset. On the sovanta dataset, high chunk sizes lead to lower average Context Relevance. As discussed earlier, this may stem from the *lost in the middle* effect, which penalizes relevant context placed mid-prompt (Liu et al. 2023).

Nevertheless, the analysis also shows that the Context Relevance metric accurately reproduces the relative performance of the different embedding and reranking models with respect to the baseline. This validates the use of this metric for RAG hyperparameter selection.

Overall, the results contribute to the literature by clarifying the relative performance of different Context Relevance variations, their correlations with a ground-truth baseline, and the potential issues affecting the metric.

5.2. Generation

The generation analysis examines variations in the language model, reranking model, chunk size, and `top_k` across the two datasets. In contrast to retrieval evaluation, there is no universally accepted ground-truth baseline for assessing machine-generated outputs, as numerous competing metrics have been proposed.

Consequently, most generation evaluation metrics continue to rely on human judgement as the gold-standard baseline (e.g. Papineni et al. 2002; Zhang et al. 2020; Zheng et al. 2023). However, collecting human judgement labels is costly and difficult to scale. Therefore, we first evaluate the alignment of various metrics with human judgement on 100 samples per dataset, and subsequently use the best-performing metrics as baselines for the full evaluations.

5.2.1. LLM-as-a-Judge

The advent of many new LLM-based metrics for machine-generated output evaluation (Yu et al. 2024) reflects an implicit expectation within the research community that these methods outperform traditional approaches such as ROUGE and BERTScore. For example, the LLM judgement approach introduced by Zheng et al. in 2023 is evaluated against human judgement, yet a comparison to other metrics is not given (Zheng et al. 2023). This represents a notable oversight, as a new approach can only be validated if it demonstrably outperforms established metrics, particularly given the substantial resource requirements of LLM judgement.

In contrast, we compare several variations of LLM judgement with more traditional metrics in terms of their correlation with human judgement. The results show that LLM judgement performance varies substantially depending on the metric configuration and the chosen language model, with correlations with human judgement ranging from 0.11 to 0.73 on the sovanta dataset and from 0.34 to 0.78 on the WikiEval dataset.

Specifically, we find that employing Chain-of-Thought reasoning generally does not improve the results, which confirms similar findings from the original study (Zheng et al. 2023). Additionally, we show that providing an explicit Likert scale to the LLM for scoring does not improve the results.

Most importantly, we demonstrate the substantial benefit of providing reference answers to the LLM judge, particularly for more complex datasets. The original work focuses on LLM judges that do not use reference answers and only gives brief results for reference-guided judgement on a very small dataset. Although the authors report promising results for this case, they provide no further analysis, possibly because their datasets lack the ground-truth labels that are necessary for such an evaluation.

The evaluation presented in this work directly addresses this shortcoming. We show that the best-performing reference-free judges on the two datasets only achieve correlations of 0.21 and 0.57, respectively, while reference-guided judges increase these values to 0.73 and 0.78. The improvement is especially pronounced on the sovanta dataset, indicating that reference answers are particularly valuable for larger, more complex datasets with overlapping topics.

We also use our LLM-based error analysis approach to investigate cases where the reference-free LLM judgement performs poorly. It shows that the largest source of errors is short yet precise model answers that are misclassified as incomplete. Furthermore, reference-free LLM judgement is completely unsuited for scoring unanswerable questions, because it has no access to the supporting context. Neither of these issues is addressed in the original work.

The performance range of LLM judgement is especially important to consider when comparing the results to other types of metrics. On the sovanta dataset, ROUGE-1 achieves a correlation of 0.68 with human judgement and BERTScore-Recall even yields 0.75 on the WikiEval dataset. This indicates that although the strongest LLM judges are still the best-performing metrics overall, ROUGE and BERTScore come close to these results and surpass many lower-performing variations of LLM judgement. In particular, all variations without reference answers (that Zheng et al. focus on) yield

lower correlations with human judgement than BLEU, ROUGE, and BERTScore. This means that the power of reference answers is larger than the power of using LLMs as evaluators.

5.2.2. Baseline Results

Computing the best-performing LLM judges on the full datasets yields the results for the various hyperparameter configurations for the generation step. Across both datasets, Mistral-Large outperforms Llama3.1 by six percentage points.

Since Mistral-Large is used as the evaluator LLM on the WikiEval dataset and Llama3.1 on the sovanta dataset, these results enable an assessment of potential self-enhancement bias. The term, coined by Zheng et al., describes the tendency of LLM judges to favor answers they generated themselves. As mentioned before, the authors find that some LLMs are susceptible to this phenomenon, while others are not (Zheng et al. 2023). Since both models assess the performance difference between themselves in nearly identical ways, this suggests that neither model is susceptible to self-enhancement bias. The only caveat here is that the values stem from different datasets, so a computation of both metric variants on the same dataset would be required to provide conclusive evidence. Nevertheless, these findings provide compelling evidence against the presence of self-enhancement bias in these two LLMs.

Notably, GPT-4o performs slightly worse than Mistral-Large on the WikiEval dataset. This result is counterintuitive, given that GPT-4o has substantially higher ratings in both MMLU-Pro and LMArena and is a larger model (Wang et al. 2024; lma 2025; Abacha et al. 2024; mis 2024). Since Mistral-Large is used as the evaluator LLM in this case, these results may hint towards self-enhancement bias. However, based on the previously discussed results, we consider this explanation unlikely.

A different explanation might be verbosity bias, which refers to LLM judges favoring longer answers and was also shown by Zheng et al. to be present in most LLMs (Zheng et al. 2023). In our evaluation, Mistral-Large produces answers that are on average twice as long as those generated by GPT-4o, which may contribute to this bias. The rest of the LLM ranking follows expectations, as Mistral-Large and GPT-4o perform significantly better than Llama3.1 and Gemini-2.0-Flash, which are smaller models.

Across both datasets, increasing the context size generally improves answer quality. However, this relationship is not as stark as suggested by the retrieval evaluation. This allows for two conclusions: First, the penalty of using shorter context sizes is limited, as the LLMs appear capable of generating meaningful answers even with less contextual information. Second, the improvement in LLM output quality from additional context diminishes beyond a certain point, even though the retrieval evaluation indicates that more relevant information is retrieved.

As discussed before, Cuconasu et al. study the effect of adding irrelevant or distracting documents to answer quality in RAG, showing that although irrelevant documents can even improve the output accuracy, distracting documents with related content can reduce the accuracy by up to 67% (Cuconasu et al. 2024). Since larger context sizes inevitably introduce more distracting content, this phenomenon may explain the dimin-

ishing returns in output quality.

Furthermore, the evaluation results show that using a powerful reranker not only improves retrieval quality, but also directly affects answer quality. LLMs may benefit from reranking, as it leads to more relevant content being placed at the start of the query, thereby improving answer quality because of the aforementioned *lost in the middle* effect. However, since the improvement from reranking on output quality is relatively small, this effect does not appear to have a substantial impact, as the improvement in answer quality may also simply stem from the more accurate retrieval in general, irrespective of the order.

As mentioned before, these baseline results are supported by the live feedback results from side-by-side feedback in Document Chat, where Mistral-Large also outperforms Llama3.1 and larger values of `top_k` generally lead to higher ratings, albeit with diminishing returns past a certain context size, possibly due to increased latency. These results show that the live feedback approach proposed by LMArena (Chiang et al. 2024) can not only be applied to LLM evaluations, but also to RAG.

5.2.3. BLEU, ROUGE, BERTScore

As mentioned earlier, current literature predominantly focuses on LLM-based metrics, with limited comparisons to traditional metrics for assessing machine output quality. This gap is particularly evident for BLEU, ROUGE, and BERTScore.

Across both datasets, the n-gram-based metrics BLEU and ROUGE show similar correlations with the respective baselines, though correlations are significantly higher on the WikiEval dataset. This similarity is expected, as both metrics follow comparable methodologies. In both cases, ROUGE-1 is the best-performing metric. On the sovanta dataset, ROUGE-1 is followed by ROUGE-L, ROUGE-2, and BLEU, whereas on the WikiEval dataset, BLEU outperforms ROUGE-L and ROUGE-2.

In the original ROUGE evaluation, ROUGE-2 and ROUGE-L perform better than ROUGE-1, which is contrary to our results. Since the basis for this evaluation is summaries, this indicates that the optimal ROUGE variant also depends on the task, though the performance order of the variants is the same across both datasets in our case. Therefore, we recommend ROUGE-1 for RAG evaluation tasks.

On both datasets, BERTScore yields results similar to BLEU and ROUGE. It is even the best-performing metric of the three on the WikiEval data, while it performs slightly worse than BLEU and ROUGE on the sovanta dataset.

In both cases, BERTScore-Recall significantly outperforms BERTScore-Precision and BERTScore-F1, with differences of up to 45 percentage points. In the original work, the authors recommend using BERTScore-F1 in most settings, although BERTScore-Recall also performs best on their image captioning task (Zhang et al. 2020). Based on our analysis, we recommend using BERTScore-Recall for RAG evaluation.

BERTScore is also compared to ROUGE and BLEU in the original work. As mentioned before, the results show that BERTScore only slightly outperforms BLEU on sentence translation, but that it achieves good results on the image captioning task where BLEU and ROUGE do not (Zhang et al. 2020). As BERTScore performs similarly to

BLEU and ROUGE on our data, the original findings appear to be scenario-specific and not directly applicable to RAG.

A possible explanation for this is that in RAG, answers are based on retrieved content and are therefore likely to reproduce the same words and statements used in the underlying content. This suggests that differently phrased answers are less common in RAG than for example in translation systems, as supported by our error analysis, where lexical variation accounts for only a minority of errors. For this reason, the ability of BERTScore to capture semantic meaning versus BLEU's and ROUGE's analysis of n-grams might not be overly beneficial for RAG applications. This may also explain why these metrics perform comparatively well for RAG relative to LLM-based metrics.

Notably, all three metrics perform better overall than any LLM-based metric without access to reference answers. This underlines the earlier point that reference answers are more important for accurate evaluation than the use of LLMs. Furthermore, Table 2.3 shows that BERTScore and ROUGE are evaluated with very large datasets, especially compared to the LLM-based metrics. This suggests a more robust basis for these metrics.

Nevertheless, our analysis highlights systematic limitations of these metrics. The LLM-based error analysis shows that the metrics tend to underestimate answer quality when the RAG system provides extra information not given in the gold answer or uses different phrasing. These issues are expected due to the design of the metrics.

Notably, these limitations affect not only the n-gram-based metrics but also BERTScore, despite its focus on semantic meaning. This indicates that stronger embedding models might be necessary to take full advantage of this approach.

5.2.4. RAG Triad

Finally, we discuss the performance of the RAG Triad, consisting of Groundedness, Answer Relevance, and Context Relevance. As noted earlier, existing literature lacks sufficient evaluations of these metrics against meaningful baselines.

For Groundedness, Llama3.1 achieves the best results on the sovanta dataset, while Claude-3.7-Sonnet performs best on the WikiEval dataset. However, both configurations yield correlations with the baseline close to zero. We also examine two variations of the Groundedness metric: filtering out trivial statements and incorporating answerability (tru 2025).

On the sovanta dataset, filtering trivial statements does not improve results, whereas it yields improvements on the WikiEval dataset. This shows that the effect of this extension depends on the data.

Because the sovanta dataset contains questions that are not answerable based on the context, it is a natural candidate for evaluating the answerability extension. The evaluation shows that incorporating this aspect into the metric only improves the results in some cases and that the best overall metric variant does not use it. Nevertheless, Groundedness performs significantly better on the sovanta dataset when unanswerable questions are excluded, suggesting that further optimization is needed to account for this factor.

Both the authors of ARES and RAGAs evaluate the Groundedness metric on relatively

small datasets and achieve high agreement with human Groundedness judgement (0.95 and 0.72, respectively) (Saad-Falcon et al. 2023; Es et al. 2023). Although these results are not directly comparable due to different baselines, they suggest that the original evaluations may overestimate the metric's performance.

Regarding Answer Relevance, Mistral-Large and Claude-3.7-Sonnet work best on the two datasets. Notably, the metric often produces very high average scores, in some cases reaching an average of 1 on the WikiEval dataset, which prevents correlation computations. Applying Chain-of-Thought reasoning yields mixed results, improving scores on the WikiEval dataset but having no effect on the sovanta dataset. This again shows that the optimal metric design depends on the dataset and the used LLM.

An important consideration for the metric is the topic of unanswerable questions. Due to the nature of the metric, abstentions (which are expected for unanswerable questions) inherently receive low scores. For this reason, the correlation with the baseline increases from 0.16 to 0.26 on the sovanta dataset when unanswerable questions are ignored. This factor should be considered when computing the metric, although omitting certain question types technically reduces its reference-free nature. Instead, we argue that in the future, an extension of the metric should be developed that automatically ignores abstentions in a similar approach to the Groundedness answerability extension.

The authors of ARES and RAGAs evaluate Answer Relevance with the same datasets as Groundedness, yielding lower, but still high accuracies compared to Groundedness. Notably, our results are reversed, as Answer Relevance has a higher correlation with the baseline than Groundedness.

Lastly, we provide a detailed analysis of the RAG Triad's optimal configuration, performance, and limitations, as these factors are not discussed in the original work (tru 2025). The results indicate that the optimal weighting of the components in the RAG Triad is 50% Answer Relevance, 25% Groundedness, and 25% Context Relevance. This holds for both datasets and is therefore our general recommendation.

Notably, the RAG Triad does not consistently outperform its individual components. It does perform better than its parts on the sovanta dataset, but is outperformed by Answer Relevance on the WikiEval dataset. This is important, because the RAG Triad's value proposition lies in improving upon its individual components.

Furthermore, the RAG Triad performs worse than the reference-based metrics BLEU, ROUGE, and BERTScore and is even outperformed by the simple reference-free LLM judgement on the WikiEval dataset. Conversely, the RAG Triad shows greater potential on the more challenging sovanta dataset where the reference-free LLM judgement fails to accurately classify answers.

Nevertheless, the overall performance of the RAG Triad and its components is substantially lower than reported in the original literature when evaluating against a human-aligned baseline. The issue is exacerbated by the fact that the WikiEval dataset used in the original RAGAs study is also used in this work. This again highlights the importance of meaningful baselines to determine the performance of a metric and shows the tendency of the original work to overestimate that performance.

The LLM-based error analysis reveals that the RAG Triad frequently overestimates answer quality, unlike the other metrics. This is mainly due to partially correct an-

swers, where certain information from the gold response is missing. Again, unanswerable queries are a major source of error on the sovanta dataset.

In total, unanswerable questions are one of the main challenges for the RAG Triad, with the correlation with the baseline increasing from 0.18 to 0.29 when these questions are excluded. In contrast, all of the other metrics yield almost the same results, regardless of whether unanswerable questions are excluded or not.

However, reducing occurrences of hallucinations is often cited as a major advantage of Retrieval-Augmented Generation (Shuster et al. 2021; Yu et al. 2024; Oro et al. 2024). This also requires that the system outputs *I don't know* when the information asked for is not present in the retrieved context. We therefore argue that this factor should be a major consideration in RAG evaluations. The fact that the RAG Triad, a metric specifically designed for RAG evaluation, performs the weakest on unanswerable questions underscores the need for further research to better account for such cases.

6. Conclusion

This work systematically evaluates Retrieval-Augmented Generation (RAG) systems by combining controlled pipeline experiments with real-world production data and live human preference feedback.

The central contribution is a unified evaluation framework that independently assesses retrieval and generation, compares traditional, embedding-based, and LLM-based metrics against established baselines, and leverages a new, real-world dataset drawn from sovanta AG's Document Chat application. Furthermore, this work shows that LLMs can be effectively used for error analysis of RAG metrics. This closes a gap in the current literature, as many LLM-based metrics are not adequately evaluated.

Unlike synthetic or purely LLM-generated datasets, the sovanta dataset contains multilingual queries, heterogeneous sources, and unanswerable questions. These unanswerable questions are particularly valuable, as mitigating hallucinations is a critical requirement for RAG systems in practical deployment. Furthermore, the dataset addresses other issues with existing datasets, such as limited reproducibility and flawed labels. Notably, the dataset also contains very challenging retrieval cases that require careful pipeline design. In addition, all evaluations are performed on the existing WikiEval dataset to ensure reliable and comparable results.

Regarding retrieval, the study shows that performance is strongly influenced by design parameters. Across experiments, BGE-M3 embeddings and a strong reranker consistently improve recall, with gains of up to four percentage points over weaker embeddings and up to seven percentage points over no reranking. However, optimal chunk size, `top_k`, and the performance of smaller reranking models are not universal: Larger context sizes improve performance, but with diminishing returns that depend on the dataset and the type of question. Dataset characteristics such as language and vector index size affect the performance of reranking models.

The evaluation shows that Context Relevance is a promising LLM-based metric for retrieval that does not require ground-truth labels, as it accurately reproduces the relative performance of different embedding and reranking models on both datasets. However, the metric only partially correctly assesses the performance of different context sizes due to incomplete retrieval results being misclassified as complete and the *lost in the middle* effect.

For generation, the work systematically benchmarks different LLM-as-a-Judge configurations, traditional n-gram and embedding metrics, and the RAG-specific scores Groundedness, Answer Relevance, and the RAG Triad. The evaluation shows that powerful, reference-guided LLM judges have the highest correlations with human judgements. However, the performance of these judges is highly sensitive to configuration, and especially reference-free LLM judges underperform relative to traditional metrics

like BLEU, ROUGE, and BERTScore, which yield moderate to high correlations with the baseline.

Groundedness and Answer Relevance yield only weak to moderate results, with optimal configurations that are dataset-specific. For the RAG Triad, the optimal weighting is determined as 50% Answer Relevance, 25% Groundedness, and 25% Context Relevance. This metric outperforms naive reference-free LLM judgement on the sovanta dataset, but performs worse than it and is even outperformed by Answer Relevance on the WikiEval dataset. Unanswerable questions present a notable challenge for the RAG Triad, as the metric lacks the gold labels necessary to determine when an abstention is the correct answer.

Overall, Mistral-Large and GPT-4o are the best-performing LLMs for the RAG pipeline, followed by Llama3.1 and Gemini-2.0-Flash. Furthermore, the generation evaluation results confirm that higher context sizes and strong reranking models lead to higher answer quality, beyond just better retrieval results.

Crucially, the study complements these offline evaluations with live human preference signals on different LLMs and values for `top_k` from Document Chat. These results rank Mistral-Large ahead of Llama3.1, confirming the offline evaluation results. For `top_k`, higher values are generally better, confirming retrieval results, though `top_k`=10 was preferred over even higher values, highlighting practical trade-offs between marginal answer quality improvements and responsiveness.

Limitations and Future Work

Although the sovanta dataset is a realistic corpus based on production data, it remains modest in size at 111 standalone prompts, is concentrated on specific enterprise domains, and only contains German and English prompts. As shown by this work, results may differ for other types of data and other languages. Use-case-specific evaluations are thus often necessary for definitive conclusions.

The live human preference evaluation would also benefit from more votes, reducing confidence intervals and therefore giving more reliable results. The feature will remain active in Document Chat, collecting more data and contributing to further improvements on the underlying RAG pipeline. In addition, non-uniform sampling of candidates and the inclusion of further hyperparameters could increase the value of the approach.

Moreover, this work does not evaluate turn-by-turn conversations, follow-up prompts, or extensions of RAG with other tools and agentic workflows that are becoming increasingly popular, indicating an area for further research.

Furthermore, while this study shows that LLMs can be used to classify metric errors, such analyses are approximate and their accuracies depend on the LLM used, the system prompt, and the metric itself. Additional evaluations with larger human-annotated datasets are necessary to further investigate this field.

In terms of the metrics, the Answer Relevance and Groundedness components of the RAG Triad require further development for increased overall alignment with human judgement and for better handling of unanswerable questions, for example by automatically ignoring abstentions.

In conclusion, this study presents a practical framework for evaluating RAG pipelines and demonstrates how retrieval and generation performance can be measured in a structured and comparable way. By testing various metrics against baselines with data from a production RAG application, it identifies both the strengths and limitations of current evaluation methods. These findings provide guidance for RAG systems in terms of hyperparameter selection, pipeline configuration, and evaluation metrics.

A. Prompts

A.1. sovanta Dataset Clustering

The following prompt was used to cluster the prompts in the sovanta dataset:

> You are an AI classifier. Your task is to assign a category label to a pair of Q&A entries based on the following categories:
> 'HR': Everything related to human resources and company policies
> 'TECH': Everything related to information technology and data science
> 'SUMMARY': When the user asks for a summary, a creative writing or a long text, or when the answer is a long text
> 'CONTRACTS': Questions regarding contracts and projects
> 'OTHER': Everything that you cannot clearly attribute to one of the other categories
>
> Return only the category name in uppercase, such as HR, TECH, SUMMARY, CONTRACTS, or OTHER. Do not explain your choice.
> Prompt: <the prompt>
> Answer: <the gold answer from the dataset>

The cluster UNKNOWN was assigned automatically when the relevant text was empty.

A.2. Retrieval Error Classes

The following system prompt was used to identify retrieval error classes on both datasets:

> You are an expert language model evaluator tasked with analyzing retrieval quality in a Retrieval-Augmented Generation (RAG) system. You will be given:
> - A query: the user's question.
> - A relevant text: this contains the gold-standard information needed to correctly answer the query.
> - A retrieved text: this is the passage retrieved by the system, intended to help answer the query.
> Your task is to classify the retrieval error by comparing the retrieved text against the relevant text and the query.
> Choose exactly one of the following classes:
> 1 - The retrieved text does contain the needed information, but the relevant text contains extra or broader content that is technically unnecessary to answer the query or describes the information differently. The essential answer is still present.
> 2 - The retrieved text omits important details that are needed to answer the query properly or entirely lacks the necessary information to answer the question.
> OTHER: Any other kind of deviation that does not fall into the categories above.
>
> Output your result in the following format: [[n]]
> where [[n]] should be replaced by one of: [[1]], [[2]], or [[OTHER]].

Class three was assigned automatically when the recall was high.

A.3. LLM Judgement Prompts

The following section includes the four system prompts used for the generation of LLM judgements.

A.3.1. Reference-guided LLM Judgement without CoT

> Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to the user question displayed below. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, and level of detail of the response. You will be given a reference answer and the assistant's answer. Be as objective as possible.
> You must rate the AI assistant's response on a scale of 1 to 5 by strictly following this format: "Rating: [[rating]]", for example: "Rating: [[3]]"

A.3.2. Reference-guided LLM Judgement with CoT

Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to the user question displayed below. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, and level of detail of the response.
You will be given a reference answer and the assistant's answer. Begin your evaluation by comparing the assistant's answer with the reference answer. Identify and correct any mistakes. Be as objective as possible.
After providing your explanation, you must rate the AI assistant's response on a scale of 1 to 5 by strictly following this format: "Rating: [[rating]]", for example: "Rating: [[3]]"

A.3.3. Reference-guided LLM Judgement with CoT and explicit Likert Scale

Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to the user question displayed below. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, and level of detail of the response.
You will be given a reference answer and the assistant's answer. Begin your evaluation by comparing the assistant's answer with the reference answer. Identify and correct any mistakes. Be as objective as possible.
After providing your explanation, you must rate the AI assistant's response on the following scale of 1 to 5:
- 5 Perfect: Output matches the gold answer exactly or improves upon it with no errors.
- 4 Good: Minor deviations from gold answer; still valid, useful, and accurate.
- 3 Acceptable: Reasonable but some flaws; not as complete or precise.
- 2 Poor: Significant issues; only partially correct or unclear.
- 1 Bad: Completely wrong or misleading.

Output the score strictly following this format: "Rating: [[rating]]", for example: "Rating: [[3]]"

A.3.4. Reference-free LLM Judgement

Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to the user question displayed below. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of the response. Begin your evaluation by providing a short explanation. Be as objective as possible.
After providing your explanation, please rate the response on a scale of 1 to 5 by strictly following this format: "Rating: [[rating]]", for example: "Rating: [[3]]"

A.4. Generation Error Classes

The following system prompts were used to identify error classes for the respective generation metrics.

A.4.1. BLEU, ROUGE, BERTScore

You are an expert in evaluating natural language generation quality. Your task is to help analyze cases where automatic evaluation metrics (like BLEU and ROUGE) diverge significantly from LLM-based or human judgement of quality. You will be given:
- A query: the original question asked by the user.
- A gold answer: the reference or expected answer.
- A model answer: the actual generated answer.
Your task is to classify the reason why BLEU/ROUGE may have failed, using one of the following error classes:
1 - Over-Elaboration: The model answer includes extensive elaboration, giving a lot more information than the gold answer, but with the information from the gold answer still present.
2 - Lexical Variation (same meaning, different words): The model answer expresses the same meaning as the gold answer using different words, phrasing, sentence structure, or slightly more information. BLEU/ROUGE fail to capture the semantic equivalence.
OTHER: Any other kind of deviation that does not fall into the categories above.

Output your result in the following format: [[n]]
where [[n]] should be replaced by one of: [[1]], [[2]], or [[OTHER]].

A.4.2. Reference-free LLM Judgement

You are an expert in evaluating natural language generation quality. Your task is to help analyze cases where reference-free LLM judgement diverges significantly from reference-based LLM judgement.
You will be given:
- A query: the original question asked by the user.
- A gold answer: the reference or expected answer.
- A model answer: the actual generated answer.
Your task is to classify the reason why the reference-free judgement may have failed, using one of the following error classes:
1 - Language Diversion: The model answer is semantically correct, but it uses a different language than the gold answer.
2 - Overly Short Answer: The model answer closely resembles the gold answer, but it is very short or minimal, leading the reference-free method to underestimate its correctness due to lack of elaboration or context.
3 - Missing Information: The answer partially answers the question, but certain information from the gold answer is missing.
4 - "I Don't Know" is the correct answer: The gold answer indicates that the information is not present in the context. Reference-free scoring fails to recognize this as correct due to lack of supporting context.
OTHER: Any other kind of deviation that does not fall into the categories above.

Output your result in the following format: [[n]]
where [[n]] should be replaced by one of: [[1]], [[2]], [[3]], [[4]], or [[OTHER]].

A.4.3. RAG Triad

You are an expert in evaluating natural language generation quality. Your task is to help analyze cases where reference-free LLM judgement diverges significantly from reference-based LLM judgement.
You will be given:
- A query: the original question asked by the user.
- A gold answer: the reference or expected answer.
- A model answer: the actual generated answer.
Your task is to classify the reason why the reference-free judgement may have failed, using one of the following error classes:
1 - Partially Correct Answer: The answer contains parts of the information from the gold answer, but certain information from the gold answer is missing.
2 - Related Answer: The answer is topically related to the query and may seem plausible, but does not contain any of the information from the gold answer.
3 - "I Don't Know" is the correct answer or question unclear: The gold answer indicates that the information is not present in the context or asks a follow-up question for clarification. Reference-free scoring fails to recognize this as correct due to lack of supporting context.
OTHER: Any other kind of deviation that does not fall into the categories above.

Output your result in the following format: [[n]]
where [[n]] should be replaced by one of: [[1]], [[2]], [[3]], or [[OTHER]].

Bibliography

(2019). GitHub - onnx/onnx: Open standard for machine learning interoperability — github.com. https://github.com/onnx/onnx?tab=readme-ov-file. [Accessed 10-07-2025].

(2023). WikiEval dataset on huggingface. https://huggingface.co/datasets/explodinggradients/WikiEval. [Accessed 12-02-2025].

(2024). BAAI/bge-reranker-v2-m3 · Hugging Face — huggingface.co. https://huggingface.co/BAAI/bge-reranker-v2-m3. [Accessed 13-06-2025].

(2024). Large Enough — Mistral AI — mistral.ai. https://mistral.ai/news/mistral-large-2407. [Accessed 15-07-2025].

(2024). New embedding models and API updates — openai.com. https://openai.com/index/new-embedding-models-and-api-updates/. [Accessed 21-06-2025].

(2025). BLEU - a Hugging Face Space by evaluate-metric — huggingface.co. https://huggingface.co/spaces/evaluate-metric/bleu. [Accessed 17-06-2025].

(2025). LlamaIndex - Build Knowledge Assistants over your Enterprise Data — llamaindex.ai. https://www.llamaindex.ai. [Accessed 10-07-2025].

(2025). LMArena — lmarena.ai. https://lmarena.ai. [Accessed 22-08-2025].

(2025). mistralai/Mistral-Small-3.1-24B-Instruct-2503 · Hugging Face — huggingface.co. https://huggingface.co/mistralai/Mistral-Small-3.1-24B-Instruct-2503. [Accessed 15-07-2025].

(2025). SAP Generative AI Hub - Availability of Generative Models. https://me.sap.com/notes/3437766/E. [Accessed 10-07-2025].

(2025). TruLens. https://www.trulens.org/getting_started/core_concepts/rag_triad/. [Accessed 11-02-2025].

Abacha, A. B., W.-w. Yim, Y. Fu, Z. Sun, M. Yetisgen, F. Xia, and T. Lin (2024). Medec: A benchmark for medical error detection and correction in clinical notes. *arXiv preprint arXiv:2412.19260*.

Bojar, O. r., C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, P. Koehn, and C. Monz (2018, October). Findings of the 2018 conference on machine translation (wmt18). In *Proceedings of the Third Conference on Machine Translation, Volume 2:*

Shared Task Papers, Belgium, Brussels, pp. 272–307. Association for Computational Linguistics.

Boubdir, M., E. Kim, B. Ermis, S. Hooker, and M. Fadaee (2024). Elo uncovered: Robustness and best practices in language model evaluation. *Advances in Neural Information Processing Systems 37*, 106135–106161.

Chen, J., R. Bao, H. Zheng, Z. Qi, J. Wei, and J. Hu (2024). Optimizing retrieval-augmented generation with elasticsearch for enhanced question-answering systems. *arXiv preprint arXiv:2410.14167*.

Chen, J., H. Lin, X. Han, and L. Sun (2024). Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 38, pp. 17754–17762.

Chen, J., S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu (2024). Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation.

Chiang, W.-L., L. Zheng, Y. Sheng, A. N. Angelopoulos, T. Li, D. Li, H. Zhang, B. Zhu, M. Jordan, J. E. Gonzalez, et al. (2024). Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*.

Cuconasu, F., G. Trappolini, F. Siciliano, S. Filice, C. Campagnano, Y. Maarek, N. Tonellotto, and F. Silvestri (2024, July). The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 2024, pp. 719–729. ACM.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186.

Dubey, A., A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. (2024). The llama 3 herd of models. *arXiv e-prints*, arXiv–2407.

Elo, A. E. (1967). The proposed uscf rating system, its development, theory, and applications. *Chess life 22*(8), 242–247.

Enevoldsen, K., I. Chung, I. Kerboua, M. Kardos, A. Mathur, D. Stap, J. Gala, W. Siblini, D. Krzemiński, G. I. Winata, S. Sturua, S. Utpala, M. Ciancone, M. Schaeffer, G. Sequeira, D. Misra, S. Dhakal, J. Rystrøm, R. Solomatin, Ömer Çağatan, A. Kundu, M. Bernstorff, S. Xiao, A. Sukhlecha, B. Pahwa, R. Poświata, K. K. GV, S. Ashraf, D. Auras, B. Plüster, J. P. Harries, L. Magne, I. Mohr, M. Hendriksen, D. Zhu, H. Gisserot-Boukhlef, T. Aarsen, J. Kostkan, K. Wojtasik, T. Lee,

M. Šuppa, C. Zhang, R. Rocca, M. Hamdy, A. Michail, J. Yang, M. Faysse, A. Vatolin, N. Thakur, M. Dey, D. Vasani, P. Chitale, S. Tedeschi, N. Tai, A. Snegirev, M. Günther, M. Xia, W. Shi, X. H. Lù, J. Clive, G. Krishnakumar, A. Maksimova, S. Wehrli, M. Tikhonova, H. Panchal, A. Abramov, M. Ostendorff, Z. Liu, S. Clematide, L. J. Miranda, A. Fenogenova, G. Song, R. B. Safi, W.-D. Li, A. Borghini, F. Cassano, H. Su, J. Lin, H. Yen, L. Hansen, S. Hooker, C. Xiao, V. Adlakha, O. Weller, S. Reddy, and N. Muennighoff (2025). Mmteb: Massive multilingual text embedding benchmark. *arXiv preprint arXiv:2502.13595*.

Es, S., J. James, L. Espinosa-Anke, and S. Schockaert (2023). Ragas: Automated evaluation of retrieval augmented generation. *arXiv preprint arXiv:2309.15217*.

Ganesan, K. (2018). Rouge 2.0: Updated and improved measures for evaluation of summarization tasks. *arXiv preprint arXiv:1803.01937*.

Huang, L., W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, et al. (2025). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems 43*(2), 1–55.

Järvelin, K. and J. Kekäläinen (2002). Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS) 20*(4), 422–446.

Kulkarni, A., Y. Zhang, J. R. A. Moniz, X. Ge, B.-H. Tseng, D. Piraviperumal, S. Swayamdipta, and H. Yu (2025). Evaluating evaluation metrics – the mirage of hallucination detection.

Lewis, P., E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems 33*, 9459–9474.

Li, X., A. Shakir, R. Huang, J. Lipp, and J. Li (2025). Prorank: Prompt warmup via reinforcement learning for small language models reranking. *arXiv preprint arXiv:2506.03487*.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81.

Lin, C.-Y. and F. J. Och (2004). Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd annual meeting of the association for computational linguistics (ACL-04)*, pp. 605–612.

Lin, T.-Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick (2014). Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pp. 740–755. Springer.

Liu, N. F., K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang (2023). Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Liu, Y., L. Huang, S. Li, S. Chen, H. Zhou, F. Meng, J. Zhou, and X. Sun (2023). Recall: A benchmark for llms robustness against external counterfactual knowledge. *arXiv preprint arXiv:2311.08147*.

Ma, X., T. Teofili, and J. Lin (2023). Anserini gets dense retrieval: Integration of lucene's hnsw indexes. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 5366–5370.

Mortaheb, M., M. A. A. Khojastepour, S. T. Chakradhar, and S. Ulukus (2025). Re-ranking the context for multimodal retrieval augmented generation. *arXiv preprint arXiv:2501.04695*.

Oro, E., F. M. Granata, A. Lanza, A. Bachir, L. De Grandis, and M. Ruffolo (2024). Evaluating retrieval-augmented generation for question answering with large language models.

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318.

Rashkin, H., V. Nikolaev, M. Lamm, L. Aroyo, M. Collins, D. Das, S. Petrov, G. S. Tomar, I. Turc, and D. Reitter (2023). Measuring attribution in natural language generation models. *Computational Linguistics 49*(4), 777–840.

Saad-Falcon, J., O. Khattab, C. Potts, and M. Zaharia (2023). Ares: An automated evaluation framework for retrieval-augmented generation systems. *arXiv preprint arXiv:2311.09476*.

Shakir, A., D. Koenig, J. Lipp, and S. Lee (2024). Boost your search with the crispy mixedbread rerank models.

Shuster, K., S. Poff, M. Chen, D. Kiela, and J. Weston (2021). Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.

Wang, L., N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei (2024). Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Wang, Y., X. Ma, G. Zhang, Y. Ni, A. Chandra, S. Guo, W. Ren, A. Arulraj, X. He, Z. Jiang, et al. (2024). Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems 37*, 95266–95290.

Ward, K. P. S. R. T. and J. H. F. Reeder (2002). Corpus-based comprehensive and diagnostic mt evaluation: Initial arabic, chinese, french, and spanish results.

Wei, J., X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems 35*, 24824–24837.

Yu, H., A. Gan, K. Zhang, S. Tong, Q. Liu, and Z. Liu (2024). Evaluation of retrieval-augmented generation: A survey. *arXiv preprint arXiv:2405.07437*.

Zhang, T., V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi (2020). Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Zheng, L., W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, et al. (2023). Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems 36*, 46595–46623.

Zhong, Z., H. Liu, X. Cui, X. Zhang, and Z. Qin (2024). Mix-of-granularity: Optimize the chunking granularity for retrieval-augmented generation. *arXiv preprint arXiv:2406.00456*.

Ehrenwörtliche Erklärung

Ich versichere, dass ich die beiliegende Bachelor-, Master-, Seminar-, oder Projektarbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und in der untenstehenden Tabelle angegebenen Hilfsmittel angefertigt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Declaration of Used AI Tools

Tool	Purpose	Where?	Useful?
ChatGPT	Rephrasing	Throughout	+
ChatGPT	Proofreading	Throughout	++
ChatGPT	Code generation	Throughout code	+

Unterschrift
Mannheim, den 2. September 2025