

THELMA: Task Based Holistic Evaluation of Large Language Model Applications-RAG Question Answering

Udita Patel*, Rutu Mulkar*, Jay Roberts*†, Cibi Chakravarthy Senthilkumar*,
 Sujay Gandhi, Xiaofei Zheng, Naumaan Nayyar, Rafael Castrillo

Amazon.com Services Inc.

{patudita, rmulkar, jarberts, sentocb, sujaygan, zhexiaof, nayyarnn, rbarquer}@amazon.com

Abstract

We propose THELMA (Task Based Holistic Evaluation of Large Language Model Applications), a reference free framework for RAG (Retrieval Augmented generation) based question answering (QA) applications. THELMA consist of six interdependent metrics specifically designed for holistic, fine grained evaluation of RAG QA applications. THELMA framework helps developers and application owners evaluate, monitor and improve end to end RAG QA pipelines without requiring labelled sources or reference responses. We also present our findings on the interplay of the proposed THELMA metrics, which can be interpreted to identify the specific RAG component needing improvement in QA applications.

1 Introduction

Retrieval Augmented Generation (RAG) paradigm (Lewis et al., 2020) improves the performance of Large Language Models (LLMs) by using excerpts of external knowledge to generate responses. It consists of two fundamental components: a retriever and a generator. In RAG based question answering (QA), a retriever searches and retrieves sources related to the asked query; followed by feeding the query and the sources to a generator model (LLMs) tasked with generation of the response. Given that RAG enables the incorporation of external knowledge, they present an excellent choice for domain adaptation of QA application in enterprise applications. RAG based QA has been widely used in automation of customer care services (Xu et al., 2024; Zhang et al., 2024a) operations and as an enhanced capability within conventional information retrieval applications. Once developed and deployed, these applications require comprehensive monitoring and

granular, instance level error diagnosis for performance evaluation as well as continual improvement. Evaluating and improving retrievers and generators components individually don’t necessarily have effect on overall performance of downstream RAG QA applications. Additionally, performance evaluation of enterprise QA applications present several unique challenges:

1. Unavailability of reference responses: Enterprise QA applications are typically built for a domain specific source corpora where these sources are continually updated, and expanded with new sources. Continual changes in the source corpora render the reference responses and/or its annotation obsolete.
2. Granular Error Diagnosis: Monitoring of production traffic on QA applications requires analyzing erroneous interactions in fine grained, interpretable manner. This allows for targeted improvements to the various components of QA applications. Existing metrics like relevancy does not provide interpretable scores to improve a specific component in RAG systems.
3. Varying tolerance levels for sub-optimal responses: Based on the criticality of use case, the tolerance towards inaccuracies varies across different dimensions of QA applications. This poses a requirement on evaluation framework to accommodate adaptable threshold mechanisms across dimensions.
4. Holistic evaluation for robust applications: During development and iterative optimization of retrievers and generators, improvements targeted at specific component can inadvertently degrade performance of another. An evaluation framework should enable assessment of individual components as well as

*Equal contribution.

†Work done at Amazon.

their interplay in order to maintain the robustness of the application.

To overcome these challenges, we propose THELMA as a reference free, multi-dimensional, granular, domain agnostic, interpretable framework for evaluation of RAG QA applications. THELMA utilizes query, source and response as input, and processes it into logical decompositions for granular assessment. The decompositions of query, source and response are then matched against other two inputs to obtain metrics score. Our overall contribution are as follows:

1. We have formally defined and implemented THELMA, a reference free evaluation framework that identifies specific areas for improvement in the implementation of RAG-based applications through its metric interplay.
2. We have redefined the notion of response irrelevance at a finer level of granularity with three distinct metrics - evaluating non-essentiality with precision, repetition with distinctness, and incorrectness with groundedness.
3. We demonstrate metrics reliability by comparing agreement of RAGAs and THELMA metrics with preferential human annotation.

2 Related work

Traditional metrics for evaluation of retrievers like recall@k and MRR (Voorhees and Tice, 2000) depends on annotation of sources and do not capture the semantic scope of knowledge base. Similarly, generator metrics like BERTScore (Zhang et al., 2020) require ground truth references as well as aren't interpretable. In recent years, large language models have also been used as evaluators for various natural language processing (NLP) tasks under the LLM-as-a-judge paradigm. Zheng et al. (2024) examined the usage and limitation of chatbots and question answering(QA) for open ended domain, but enterprise QA applications are typically closed domain. Ru et al. (2024a) proposed RAGchecker with granular metrics which are similar to the ones proposed in THELMA but are not suitable for monitoring of real world applications where references responses are unavailable. Another class of frameworks focuses on scenarios when reference responses and ground-truth annotation for retriever are unavailable. Es et al. (2024) proposed RAGAs with three key dimensions as introduced in Trulens

(J. Ferrara, 2024): context relevance, answer relevance and faithfulness. ARES (Saad-Falcon et al., 2024) evaluated along the same dimension as RAGAs, but curated a small set of human annotations and uses fine tuned LLMs as judges. However, relevance measured in these metrics is coarse grained and cannot provide actionable diagnosis needed for continuous improvements of applications. Wang et al. (2024) employs a rubric based approach to evaluate responses where the LLM also produces reasoning along with the scores but does not address retriever evaluation. Model based evaluations have also been proposed for diagnosis of specific issues observed in RAG based question answering. Incorrectness in responses identified by validating against a reference source has been evaluated by FactScore (Min et al., 2023) and Refchecker (Hu et al., 2024). We include FactScore's methodology termed as groundedness in our framework. Presence of irrelevant and absence of a relevant source in a set of retrieved sources has also been evaluated previously. (Salemi and Zamani, 2024) introduced eRAG where sources in a ranked retrieved set are evaluated for relevancy based on the ground truth labels of the RAG downstream task. RAGChecker (Ru et al., 2024b) measures context precision and recall by comparing sources with reference answers, while THELMA measures source precision by evaluating presence of a irrelevant source and absence of an essential source w.r.t to the question under consideration making it reference free. We establish similar precision-coverage trade-off with response metrics. Zhang et al. (2024b) evaluates repetition verbosity of responses with performance difference and relative performance difference metrics but requires reference responses, unlike THELMA's self distinctness metric.

3 THELMA Framework

THELMA is a suite of LLM-as-Judge metrics to evaluate RAG based QA applications. Proposed metrics are based on RAG triad but enables more granular diagnosis. RAG Triad is an evaluation framework introduced by trulens (J. Ferrara, 2024) to holistically assess reliability of LLM generated responses in a QA application. It consists of query(q), retrieved sources(s) and response(r) generated by the LLM, evaluated using context relevance, response relevance and groundedness using model based evaluations.

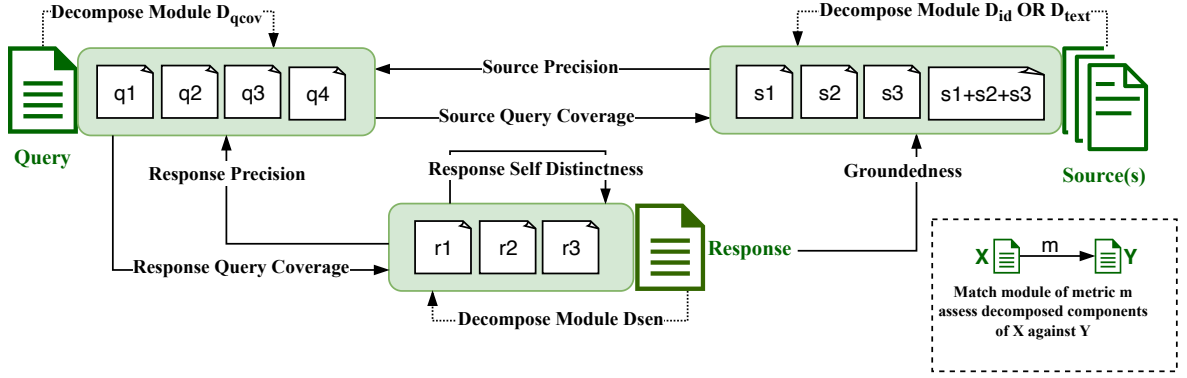


Figure 1: Overview of THELMA framework with RAG triad components; Query, Source(s) and Response.

We propose our evaluation framework under the following assumptions: 1) Query, source, response triplets are available 2) Generator should not respond with internal knowledge; the source corpora contains only accurate information, hence can be treated as ground truth. 3) Reference responses are unavailable. Each metric is defined with three modules; Decompose, Match and Aggregate.

1. Decompose module fragments the textual inputs into independent logical components, hereon referred to as *decomposed components*. Implementation is similar to claim extraction in [Min et al. \(2023\)](#).
2. Match module assesses a decomposed component against other textual inputs based on metric specific criterion.
3. Aggregate module combines individual matching scores on decomposed components to produce a score for a specific metric on given (query, source response) triplet.

We demonstrate the reliability of proposed metrics when compared against human judgement in table 1 and effectiveness of framework in interpreting actionable insights in table 2.

4 THELMA Metrics

Formally, let G be the function representing a generator LLM. Let T be the retriever model used to retrieve set s from S using q as input. G generates response $r \in R$ for a query $q \in Q$ using retrieved source set $s \in S$.

$$s = T(q), r = G(q, s) \quad (1)$$

G and T are assumed to be inaccessible for the purposes of all metrics calculations. All metrics follow

the same set of steps. First, a decompose module $D : X \mapsto \{x_1, \dots, x_n\}$ transforms given textual input into independent atomic components. Next, a match module $m : X \times Y \mapsto \{0, 1\}$ which assesses two textual inputs based on the metric specific criterion. Match module outputs 1 if the criteria is satisfied, 0 otherwise. Lastly, an aggregate module calculates a metric score over an instance of (q, s, r) triad as the macro average of the matching function, unless specified otherwise. The choice of decompose and match module, textual input and the aggregate function defines each metric. All metrics produce scores between 0 and 1 using combination of two text inputs from the (q, s, r) triplet, with higher scores denoting better performance.

4.1 Source Precision

Retrievers sometimes can retrieve related but unnecessary sources for a given query ([Wu et al., 2024](#)). These irrelevant sources progressively degrade response quality by detracting LLMs from generating correct answers ([Cuconasu et al., 2024](#)). Presence of unnecessary sources is evaluated with source precision.

Source precision is the degree to which the retrieved source set s is essential to answer the query. Source precision of set s with respect to q , is

$$\begin{aligned} & \text{Source_Precision}(s, q) \\ &= \frac{1}{|D_{id}(s)|} \sum_{s_j \in D_{id}(s)} m_{sp1}(s_j, q). \end{aligned} \quad (2)$$

The necessary and unnecessary components in a source set can be evaluated with two variants. First variant is as follows. **Decompose module:** D_{id} is an identity function, hence each source s_j remains unchanged as retrieved by the retriever. **Match**

module: m_{sp1} operates at a single source level, assigning a score of 1 to entire source s_j that contains any essential information for answering the query. This diagnosis can uncover issues in retriever, where a retrieved source s_j is entirely non-essential for answering any query.

$$\begin{aligned} & \text{Source_Precision}(s, q) \\ &= \frac{1}{|D_{text}(s)|} \sum_{s_j \in D_{text}(s)} m_{sp2}(s_j, q) \end{aligned} \quad (3)$$

The second variant, aims to identify all non essential facts within the entire source set s . **Decompose module:** D_{text} transforms the source set s into a set of components, where each component s_j is a standalone fact, conveying one piece of information. **Match module:** m_{sp2} assesses the essentiality of a decomposed source component for answering the query. This granular level evaluation of s quantifies the presence of total amount of non-essential information. This diagnosis can uncover issues in source chunking strategies as well as in how the presence of non essential information in the sources is detracting the LLM from generating the correct answer.

4.2 Source Query Coverage

Retrievers will sometimes fail to retrieve sources containing the answer(s) for all or few of subqueries. Ideally, generator should reject answering such queries (or part thereof) but only do so with 45% success rate, and instead use its intrinsic knowledge to generate ungrounded responses (Chen et al., 2023). Therefore, absence of necessary sources required to fully answer given query is evaluated with source query coverage.

Source query coverage is degree to which a source set s answers all components of the decomposed query. Source query coverage of s with respect to q , is

$$\begin{aligned} & \text{Source_Query_Coverage}(s, q) = \\ & \frac{1}{|D_{qcov}(q)|} \sum_{q_l \in D_{qcov}(q)} \max_{s_j \in D_{ic}(s)} m_{sqcov}(q_l, s_j). \end{aligned} \quad (4)$$

Decompose module: D_{qcov} transforms query q into a set of decomposed query components, where each component represents a standalone question from the original query while resolving pronouns, ignoring greeting and factual statements that do not

represents a question. **Match module:** Query component q_l is assessed independently for source components and also against the entire source set. Assessing with source components maintains prompt performance which was observed to deteriorate when the number of retrieved sources or the length of the sources increased. Assessment with entire source set handles multi-hop query scenarios. m_{sqcov} assesses the presence of an answer for query component q_l in each of the source components $s_j \in D_{ic}(s)$. If any of source component answers q_l , the matching score is set as 1.

4.3 Response Precision

LLMs tend to prefer generating lengthy, verbose responses adding extraneous details along with necessary information required to answer the query (Zhang et al., 2024b). This verbosity in enterprise applications using proprietary models (LLMs) leads to increased cost, as they often employ token-based pricing models. Presence of *extraneous, unnecessary* information is evaluated with response precision metric.

Response precision is the degree to which the amount of information provided in response necessary to answer the query. A precise response answers the question directly and avoids additional unnecessary information. Response precision of r with respect to q , is

$$\begin{aligned} & \text{Response_Precision}(r, q) \\ &= \frac{1}{|D_{text}(r)|} \sum_{r_k \in D_{text}(r)} m_{rp}(r_k, q). \end{aligned} \quad (5)$$

Decompose module: We use the same decompose D_{text} to transform both the source set s into facts and the generated response r into claims.

Match module: m_{rp} assesses the essentiality of a response component r_k for answering any part of the query.

4.4 Response Query Coverage

Generator LLMs may produce incomplete and/or insufficient answers due to ambiguous generator prompts or language complexities in retrieved sources. Additionally, LLMs can struggle to identify relevant information when it's present within lengthy source texts, thereby missing answering the query (Liu et al., 2024). *Incompleteness* of response is evaluated with response query coverage.

Response query coverage is degree to which the response answers the query. Response query cover-

	Model	Context Relevance			Answer Relevance			Faith.
		SP1	SP2	SQC	RP	RQC	SD	GR
RAGAs	Sonnet 3.5	0.90	0.35	0.70	0.64	0.50	0.26	0.96
	Haiku 3.5	0.94	0.42	0.17	0.26	0.43	0.42	0.50
	Llama 3.3 70b	0.92	0.28	0.29	0.63	0.68	0.52	0.94
THELMA	Sonnet 3.5	1.0	0.35	0.76	0.84	0.75	1.0	0.96
	Haiku 3.5	1.0	0.42	0.82	0.95	0.87	1.0	0.96
	Llama 3.3 70b	1.0	0.64	0.70	0.89	0.88	1.0	1.0

Table 1: Agreement with human annotator in pairwise comparisons. Source Precision, Source Query Coverage is compared with coarse grained Context Relevance in RAGAs. Response Precision, Response Query Coverage, and Response Self Distinctness is compared with coarse grained Answer Relevance in RAGAs. Groundedness is compared with Faithfulness in RAGAs.

age of r with respect to q , is

$$\text{Response_Query_Coverage}(r, q) = \frac{1}{|D_{qcov}(q)|} \sum_{q_i \in D_{qcov}(q)} m_{rqcov}(q_i, r). \quad (6)$$

Decompose module: D_{qcov} used in source query coverage, see 4.2. **Match module:** m_{rqcov} validates that the intent of the decomposed query component q_i is addressed in response. It also assess whether the logical structure of the response resolves the query.

4.5 Response Self-Distinctness

LLMs also demonstrate verbosity by repeating information through paraphrasing (Briakou et al., 2024). In enterprise applications, it is crucial to identify repetitiveness as it can decrease user readability while also contributing to cost and latency. Presence of *repetitive, redundant information* is evaluated with response self-distinctness.

Response self distinctness is the degree to which components of the response are dissimilar from each other. A self-distinct response will not repeat information within itself. Response self distinctness of r , is

$$\text{Response_Self_Distinctness}(r) = \frac{1}{|D_{sen}(r)|} \sum_{r_j, r_u \in D_{sen}(r)} (1 - m_{sd}(r_j, r_u)). \quad (7)$$

Decompose Module: D_{sen} splits the response r into separate sentences using primary sentence terminators i.e. Periods, exclamation marks, and question marks. **Match Module:** Match module calculates cosine similarity between two response sentences represented with titan-v1 embedding. $(1 - m_{sd})$ is treated as distinctness score.

4.6 Groundedness

A response r generated by an LLM M may contain facts derived from its intrinsic knowledge acquired during pre-training (Huang et al., 2025). The intrinsic knowledge may not necessarily align with facts and domain specific information present in source s provided as input. For evaluating a closed-domain enterprise application, the domain knowledge stored in the authoritative source(s) is assumed to be the ground truth. Thus, we measure the presence of *incorrectness* of the response r based on its infidelity to the facts present in provided source(s) s . Groundedness of r with respect to s , is

$$\text{Groundedness}(r, s) = \frac{1}{|D_{text}(r)|} \sum_{r_i \in D(r)} m_{gr}(r_i, s). \quad (8)$$

Decompose Module: D_{text} transforms response r in to a set of independent claims, where each component is a sentence representing a stand alone factual statement extracted from the response as proposed in Min et al. (2023). **Match Module:** validates each fact component r_i against the given source set s . Matching module takes as input a fact component r_i , as well as the retrieved source set s .

5 Experiments

We augment and report results on randomly sampled subset (240 data points) of WikiEval from RAGAs (Es et al., 2024). WikiEval dataset is augmented manually by modifying data points to contain sub-optimal sources or responses representing issues assessed by each metrics. The dataset is

Metric interplay	Interpretation	Component to be improved
SD↓ RP↑	Lengthier responses with relevant but repetitive information, low user readability.	Prompt or Generator
SQC↓ RQC↓	Inaccurate retrieval OR missing information in source corpora.	Retriever or Source text.
SP↓ SQC↑	All components of the query are being addressed, but some retrieved sources are only loosely relevant.	Retriever
RQC↓ SQC↑	Information required to answer the query is present in source but not used in response.	Prompt or Generator
RP↓ SP ₁ ↑	Response contains extraneous information but majority of retrieved sources are essential.	Prompt or Source chunking
SQC↓ RQC↑ GR↓	Generator responding to queries (or part thereof) that are not addressed in source, causing ungroundedness.	Prompt

Table 2: Diagnosis of RAG QA applications with interplay of THELMA metrics. Arrows denote low(↓) and high(↑) metric scores.

annotated with preferential data annotation strategy. For each instance in the dataset, annotators were given two sets of source or response pairs for every query, q . All annotators are developers or scientists from the adopter teams and are fluent in English. They were given clear instructions with metrics definitions. We then compared agreement of RAGAs and THELMA with human annotation in pairwise comparison. An agreement between human and evaluation framework would entail evaluation framework scoring the better source, response higher than the other. For fair comparison of frameworks, we use the same underlying LLM for both frameworks and compare our metrics against the most relevant metrics from RAGAs. Our proposed metrics consistently score the better source set and response with higher value (see Table 1). Consistent to conclusions by Es et al. (2024), we find the source metrics to be the hardest dimensions to evaluate. Open source model(Llama) based THELMA also produced comparable results to proprietary models. Table 2 demonstrate how THELMA scores are interpreted by application developers to make further improvements.

6 Conclusion

This paper presents THELMA, a novel reference free evaluation framework for RAG based question answering task. The framework provides granular

independent assessment of retrievers and generators while also providing insights to application developers on specific areas of improvements by introducing precision-coverage trade off. The evaluation of the framework on THELMA-WikiEval dataset has shown consistent alignment with human judgments regardless of the underlying LLM.

7 Limitations

The framework is unlikely to be useful to evaluate QA systems with unverified source corpus. Secondly, the framework does not perform well on creative or opinionated questions e.g. summarization, essay writing, text comparisons. In the current state, the framework has only been evaluated for its performance on unstructured text sources. Thirdly, the framework heavily depends on the performance of underlying LLMs.

Ethics Statement

We do not anticipate any ethical concerns with the framework presented here. In terms of cost, proposed metrics produce comparable performance with open source LLMs. Customer data or personally identifiable information is not discussed or released through this paper.

References

- Eleftheria Briakou, Zhongtao Liu, Colin Cherry, and Markus Freitag. 2024. [On the implications of verbose llm outputs: A case study in translation evaluation](#). *Preprint*, arXiv:2410.00863.
- Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2023. [Benchmarking large language models in retrieval-augmented generation](#). *Preprint*, arXiv:2309.01431.
- Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. [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 '24, page 719–729, New York, NY, USA. Association for Computing Machinery.
- Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. [RAGAs: Automated evaluation of retrieval augmented generation](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.
- Xiangkun Hu, Dongyu Ru, Lin Qiu, Qipeng Guo, Tianhang Zhang, Yang Xu, Yun Luo, Pengfei Liu, Yue Zhang, and Zheng Zhang. 2024. [RefChecker: Reference-based Fine-grained Hallucination Checker and Benchmark for Large Language Models](#). *arXiv preprint*. ArXiv:2405.14486 [cs].
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. [A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions](#). *ACM Transactions on Information Systems*, 43(2):1–55.
- O. M. Ozturk J. Ferrara, Ethan-Tonic. 2024. Rag triad; TruLens trulens.org. https://www.trulens.org/getting_started/core_concepts/rag_triad/. [Accessed 11-03-2025].
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. [Lost in the middle: How language models use long contexts](#). *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Sewon Min, Kalpesh Krishna, Xinxu Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. [FActScore: Fine-grained atomic evaluation of factual precision in long form text generation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.
- Dongyu Ru, Lin Qiu, Xiangkun Hu, Tianhang Zhang, Peng Shi, Shuaichen Chang, Cheng Jiayang, Cunxiang Wang, Shichao Sun, Huanyu Li, Zizhao Zhang, Binjie Wang, Jiarong Jiang, Tong He, Zhiguo Wang, Pengfei Liu, Yue Zhang, and Zheng Zhang. 2024a. [Ragchecker: A fine-grained framework for diagnosing retrieval-augmented generation](#). *Preprint*, arXiv:2408.08067.
- Dongyu Ru, Lin Qiu, Xiangkun Hu, Tianhang Zhang, Peng Shi, Shuaichen Chang, Cheng Jiayang, Cunxiang Wang, Shichao Sun, Huanyu Li, Zizhao Zhang, Binjie Wang, Jiarong Jiang, Tong He, Zhiguo Wang, Pengfei Liu, Yue Zhang, and Zheng Zhang. 2024b. [RAGChecker: A Fine-grained Framework for Diagnosing Retrieval-Augmented Generation](#). *arXiv preprint*. ArXiv:2408.08067 [cs].
- Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2024. [ARES: An automated evaluation framework for retrieval-augmented generation systems](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 338–354, Mexico City, Mexico. Association for Computational Linguistics.
- Alireza Salemi and Hamed Zamani. 2024. Evaluating retrieval quality in retrieval-augmented generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2395–2400.
- Ellen M. Voorhees and Dawn M. Tice. 2000. [The TREC-8 question answering track](#). In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).
- Yang Wang, Alberto Garcia Hernandez, Roman Kyslyi, and Nicholas Kersting. 2024. [Evaluating quality of answers for retrieval-augmented generation: A strong llm is all you need](#). *Preprint*, arXiv:2406.18064.
- Siye Wu, Jian Xie, Jiangjie Chen, Tinghui Zhu, Kai Zhang, and Yanghua Xiao. 2024. [How easily do irrelevant inputs skew the responses of large language models?](#) In *First Conference on Language Modeling*.
- Zhentao Xu, Mark Jerome Cruz, Matthew Guevara, Tie Wang, Manasi Deshpande, Xiaofeng Wang, and Zheng Li. 2024. [Retrieval-augmented generation with knowledge graphs for customer service question answering](#). In *Proceedings of the 47th International ACM SIGIR Conference on Research and*

Development in Information Retrieval, SIGIR '24, page 2905–2909, New York, NY, USA. Association for Computing Machinery.

Tianyang Zhang, Zhuoxuan Jiang, Shengguang Bai, Tianrui Zhang, Lin Lin, Yang Liu, and Jiawei Ren. 2024a. [RAG4ITOps: A supervised fine-tunable and comprehensive RAG framework for IT operations and maintenance](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 738–754, Miami, Florida, US. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with bert](#). *Preprint*, arXiv:1904.09675.

Yusen Zhang, Sarkar Snigdha Sarathi Das, and Rui Zhang. 2024b. [Verbosity \$\neq\$ veracity: Demystify verbosity compensation behavior of large language models](#). *Preprint*, arXiv:2411.07858.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhonghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, Red Hook, NY, USA. Curran Associates Inc.

A THELMA Examples

A.1 Source precision

1. **Query:** When was the Chimnabai Clock Tower completed, and who was it named after?
2. **Sources and match scores**
 - (a) The Chimnabai Clock Tower, also known as the Raopura Tower, is a clock tower situated in the Raopura area of Vadodara, Gujarat, India. It was completed in 1896 and named in memory of Chimnabai I (1864,–1885), a queen and the first wife of Sayajirao Gaekwad III of Baroda State. It was built in Indo-Saracenic architecture style. *score* = 1
 - (b) Chimnabai Clock Tower was inaugurated by Mir Kamaluddin Hussainkhan, the last Nawab of Baroda. During the rule of Gaekwad, it was a stoppage

for horse drawn trams. It was constructed from the funds raised by the citizens of Baroda State. The premises of the tower were rented by the Vadodara Municipal Corporation for the last three decades. In 2017, the municipal commissioner, Vinod Rao, ordered to vacate the tower to conserve it as a heritage monument. The mechanical system of the tower clock was replaced by DC motor.' *score* = 0

3. **Source Precision:** 0.5

A.2 Source Query Coverage

1. **Source:** The Chimnabai Clock Tower, also known as the Raopura Tower, is a clock tower situated in the Raopura area of Vadodara, Gujarat, India. It was completed in 1896. It was built in Indo-Saracenic architecture style.
History
 It was inaugurated by Mir Kamaluddin Hussainkhan, the last Nawab of Baroda. During the rule of Gaekwad, it was a stoppage for horse drawn trams. It was constructed from the funds raised by the citizens of Baroda State. The premises of the tower were rented by the Vadodara Municipal Corporation for the last three decades. In 2017, the municipal commissioner, Vinod Rao, ordered to vacate the tower to conserve it as a heritage monument. The mechanical system of the tower clock was replaced by DC motor.'
2. **Query:** When was the Chimnabai Clock Tower completed, and who was it named after?
3. **Decomposed query and match scores :**
 - (a) When was the Chimnabai Clock Tower completed? *score* = 1
 - (b) Who was Chimnabai Clock Tower named after? *score* = 0
4. **Source Query Coverage:** 0.5

A.3 Response Precision

1. **Query:** When was the Chimnabai Clock Tower completed, and who was it named after?
2. **Response:** The Chimnabai Clock Tower was completed in 1896. It was named after Chimnabai I, who was a queen and the first wife of Sayajirao Gaekwad III of Baroda State. The construction of clock tower was completed in 1896.

3. Decomposes response and match scores:

- (a) The Chimnabai Clock Tower was completed in 1896. *score* = 1
- (b) The place was named after Chimnabai I. *score* = 1
- (c) Chimnabai I was a queen. *score* = 0
- (d) Chimnabai I was the first wife of Sayajirao Gaekwad III. *score* = 0
- (e) Sayajirao Gaekwad III was the ruler of Baroda State. *score* = 0
- (f) The clock tower was constructed. *score* = 0
- (g) The construction of the clock tower was completed in 1896. *score* = 1

4. Response Precision: 0.42

A.4 Response Query Coverage

1. **Query:** When was the Chimnabai Clock Tower completed, and who was it named after?
2. **Response:** The Chimnabai Clock Tower was completed in 1896.

3. Decomposed query and match scores :

- (a) When was the Chimnabai Clock Tower completed? *score* = 1
- (b) Who was Chimnabai Clock Tower named after? *score* = 0

4. Response Query Coverage: 0.5

A.5 Response Self Distinctness

1. **Response:** The Chimnabai Clock Tower was completed in 1896. It was named after Chimnabai I, who was a queen and the first wife of Sayajirao Gaekwad III of Baroda State. The construction of clock tower was completed in 1896.

2. Decomposes response and match scores:

- (a) The Chimnabai Clock Tower was completed in 1896. *score* = 1
- (b) It was named after Chimnabai I, who was a queen and the first wife of Sayajirao Gaekwad III of Baroda State. *score* = 0
- (c) The construction of clock tower was completed in 1896. *score* = 1

3. Response Self Distinctness: 1-0.66 = 0.33

A.6 Groundedness

1. **Source:** The Chimnabai Clock Tower, also known as the Raopura Tower, is a clock tower situated in the Raopura area of Vadodara, Gujarat, India. It was completed in 1896 and named in memory of Chimnabai I (1864,–1885), a queen and the first wife of Sayajirao Gaekwad III of Baroda State. It was built in Indo-Saracenic architecture style.

History

Chimnabai Clock Tower was built in 1896. The tower was named after Chimnabai I (1864,–1885), a queen and the first wife of Sayajirao Gaekwad III of Baroda State. It was inaugurated by Mir Kamaluddin Hussainkhan, the last Nawab of Baroda. During the rule of Gaekwad, it was a stoppage for horse drawn trams. It was constructed from the funds raised by the citizens of Baroda State. The premises of the tower were rented by the Vadodara Municipal Corporation for the last three decades. In 2017, the municipal commissioner, Vinod Rao, ordered to vacate the tower to conserve it as a heritage monument. The mechanical system of the tower clock was replaced by DC motor.'

2. **Response:** The Chimnabai Clock Tower was completed in 1856. It was named after Chimnabai I, who was a queen and the first wife of Sayajirao Gaekwad III of Baroda State. The construction of clock tower was completed in 1856.

3. Decomposes response and match scores:

- (a) The Chinnabai Clock Tower was completed in 1896. *score* = 0
- (b) The place was named after Chinnabai I. *score* = 1
- (c) Chinnabai I was a queen. *score* = 1
- (d) Chinnabai I was the first wife of Sayajirao Gaekwad III. *score* = 1
- (e) Sayajirao Gaekwad III was the ruler of Baroda State. *score* = 1
- (f) The clock tower was constructed. *score* = 1
- (g) The construction of the clock tower was completed in 1896. *score* = 0

4. Groundedness: 0.71

B Prompts

B.1 Source, Response Decompose Module

You are a claim extractor and your job is to extract claims from snippets of text. You always follow the below guidelines:

<guidelines>

- You only respond with a list of stand-alone claims.
 - The list is always inside an <output> </output> tags.
 - The extracted claims contain only one piece of stand-alone information such that each claim can be verified independently.
 - The extracted claims should be independently verifiable but do not need to be true.
 - The extracted claims should be stand-alone facts.
 - The extracted claims are independent so there is very little overlap between them.
 - The list of extracted claims should contain all information from the input.
 - Each extracted claim should not be decomposable into multiple claims.
 - Generate as many claims as possible.
- </guidelines>

Human: Here are some examples:

<examples>

Human: Please extract claims from the following text:

<input>

Text: Jay got his PhD in Santa Barbara and during that time he got a dog named Basil. His PhD was focused on the mathematics of plasmas and he did some deep learning on the side. </input>

Assistant:

<output>

- Jay got a PhD in Santa Barbara.
- Jay got a dog named Basil in Santa Barbara.
- Basil is a dog from Santa Barbara.
- Jay worked on plasmas during his PhD.
- Jay worked on deep learning during his PhD.

</output>

</examples>

Human: Please extract claims from the following text: <input> {input} </input>

B.2 Response Precision and Source Precision Match Module

You are an editor. Your task is to identify if given fact contains essential information to answer a query. You will be given a pair of query and fact. Your task is to rate whether the fact is essential to answer the query. Please follow the instructions carefully.

Evaluation Criteria: Relevance Score captures whether the information in the fact is essential to answer the specific query asked. This dimension assesses if the response provides relevant details and does not have any irrelevant details.

Instructions:

<instructions>

- Read the query carefully to thoroughly understand the user's query. Identify the key points that are being asked about.
- Analyze the response to ensure it is essential to answer the query.
- Check if response is providing details

about same entity or event asked in the query.

- Check if the response provides any extraneous details which are not needed to answer the query.
 - Provide the response in `<output>` `</output>` tags.
 - Respond only with a value in the scale provided below. No explanations.
- `</instructions>`

Scale:

`<scale>`

Extraneous - The response is not required to answer the core intent of the query.

Essential - The response is essential to answer the query.

`</scale>`

Human: Please rate the essentiality of the fact to answer the query:

`<input>`

Question: {query}

Response: {response}

`</input>`

B.3 Query Decompose Module

You are a question decomposer. You will follow following guidelines.

`<guidelines>`

- You only respond with a list of questions.
- The list is always inside of `<output>` `</output>` tags.
- You should resolve any ambiguous pronouns in the input text.
- Each extracted question contain only one question.
- Each extracted question is one short question.
- All the extracted questions are from the input questions received.

`</guidelines>`

Here are some examples:

`<examples>`

Human: Please extract questions from the following text:

`<input>`

Text: Imagine you are a travel agent and you are trying to book flights to London. What is the most effective way to book

flights to London? `</input>`

Assistant: `<output>` What is the most effective way to book flights to London? `</output>`

`</examples>`

Human: Please extract questions from the following text:

`<input>`

Text: {input}

`</input>`

B.4 Groundedness Match Module

CLAIM_EXAMPLE_KS_1 Hawaii is allegedly named after Hawai'iloa, a legendary Polynesian navigator who is said to have discovered the island. Other accounts attribute the name to the legendary realm of Hawaiki, a place from which some Polynesians are said to have originated, the place where they transition to in the afterlife, or the realm of the gods and goddesses. James Cook, the English explorer and navigator who captained the first European expedition to reach the Hawaiian Islands, called it O-Why-hee (from Hawaiian) and the Sandwich Islands after his patron, the Earl of Sandwich.

CLAIM_EXAMPLE_KS_2 = You can deploy a model trained with SageMaker to your own deployment target. To do that, you need to know the algorithm-specific format of the model artifacts that were generated by model training. For more information about output formats, see the section corresponding to the algorithm you are using in Common Data Formats for Training. - You can deploy multiple variants of a model to the same SageMaker HTTPS endpoint. This is useful for testing variations of a model in production. For example, suppose that you've deployed a model into production. You want to test a variation of the model by directing a small amount of traffic, say 5%, to the new model. To do this, create an endpoint configuration that describes both variants of the model. You specify the ProductionVariant in your request to the CreateEndPointConfig. For more information, see ProductionVariant.

You are a fact checker. You will be given a claim and a knowledge base. You will then answer with a 0 or 1 indicating if the claim is not supported or supported by the knowledge base. The scale is addressed here:

<scale>

- 0: The claim is not supported by the knowledge base.
- 1: The claim is definitely supported by the knowledge base.

</scale>

You always adhere to the following guidelines:

<guidelines>

- The output is always between <output> and </output>.
- Only the score is returned. Explanations are forbidden.

</guidelines>

Here is an example:

<example>

<example1>

Human: Is the claim supported by the knowledge base?

<input>

<knowledge source> {CLAIM_EXAMPLE_KS_1}

</knowledge source>

<claim>

The capital of Hawaii is Honolulu.

</claim>

</input>

Assistant: <output>0</output>

</example1>

<example2>

Human: Is the claim supported by the knowledge base?

<input>

<knowledge source>

{CLAIM_EXAMPLE_KS_2}

</knowledge source>

<claim>

SageMaker allows you to deploy different model variants to the same endpoint.

</claim>

</input>

Assistant: <output>1</output>

</example2>

</example>

Human: Is the claim supported by the knowledge base?

C Human Annotation

As mentioned in Section 5, we manually augmented on a randomly sampled subset of WikiEval dataset from RAGAs (Es et al., 2024). Originally, each data point in this dataset contains a question, correct response (positive), correct sources (positive), ungrounded response (negative), irrelevant response (negative) and irrelevant sources (negative). We modified this dataset by adding sub-optimal responses and sources for each datapoint corresponding to each metric. Specifically, we took a subset of 20 datapoints from WikiEval dataset and added the following columns for each of them.

- We took the correct response and added irrelevant facts that is not essential to answer the question. This serves as negative for Response Precision metric i.e., We test whether the Response Precision metric scores the correct response higher than the modified negative response.
- We took the correct response and removed one of the essential piece of information to answer the question. This serves as negative for Response Query Coverage metric i.e, We test whether the Response Query Coverage metric scores the correct response higher than the one with missing essential points.
- We took the correct response and rephrased the facts in the response and added them to the response such that it contains redundant information. We use this response with redundant information as negative to test Self-distinctness whether the metric scores correct response higher than the response with redundant information.
- We took the correct set of sources and added randomly chosen irrelevant sources to them to create negative for Source Precision metric at chunk level (variant 1). For fact level variants, we added irrelevant information to some of the sources to create the negatives. Using these negatives, we test whether the Source Precision metric scores the correct sources higher than the negative sources.
- We took the correct set of sources and removed essential facts necessary to answer the

question. This serves as negative for Source Query Coverage metric which we used to test whether Source Query Coverage metric scores the correct sources higher than the sources with missing essential information.

In this way, we create negatives for each dimension and compare our metric with RAGAs. As mentioned in Section 2, RAGAs does not contain coarse-grained metrics. Hence, we compare THELMA metrics with its closest counterpart in RAGAs as mentioned in 1.