Retrieval Augmented Generation based Question-Answering for Contextual Response Prediction System

Sriram Veturi
The Home Depot
sriram_veturi@homedepot.com

Saurabh Vaichal
The Home Depot
saurabh_s_vaichal@homedepot.com

Reshma Lal Jagadheesh
The Home Depot
reshma_lal_jagadheesh@homedepot.com

Nian Yan
The Home Depot
nian_yan@homedepot.com

Abstract

Building an effective and factually accurate question-answering framework for real-world applications is non-trivial because of 1. data availability issues 2. evaluating the quality of generated content (Hallucinations) 3. human evaluation is time-consuming and expensive. In this paper, we introduce an end-to-end framework for evaluating the performance of Large Language models(LLMs) for industry use cases where there are severe training data and annotation constraints. Performance of off-the-shelf LLMs with state-of-the art approaches such as Retrieval Augmented generation and ReAct combined with different prompting techniques like Chain of Thought prompting and Chain of Verification prompting are also compared here. We also outline the different evaluation strategies that were effective in measuring the quality of data and the factual correctness of the algorithms. The paper also covers qualitative and quantitative approaches for evaluating the performance of these LLMs for generating grounded responses. This work was implemented for generating response suggestions for customer service agents in contact centers of a major retail company. We demonstrate that the grounded response suggestions generated by LLMs generate more accurate and relevant suggestive responses compared to the existing BERT-based algorithms used in production.

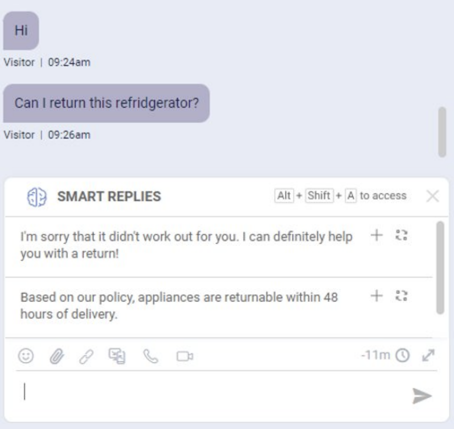
1 Introduction

LLMs have shown impressive performance in a variety of language tasks, but they can generate incorrect or biased information, as their responses are based on patterns learned from data which may not always reflect real-world accuracy or ethical considerations. To overcome this problem, Retrieval Augmented Generation (RAG) Lewis et al. (2021) is one of the most common frameworks used for grounding LLMs on factual information. RAG architecture first takes the user input as a query and then 1) retriever retrieves a set of k documents that are similar to the query and then, 2) the language model incorporates the k retrieved documents as additional information to make a final prediction. The style of retrieval could be added to both encoder-decoder (Yu (2022); Izacard et al. (2022)) and decoder-only models (Khandelwal et al. (2020); Borgeaud et al. (2022); Shi et al. (2022); Rubin et al. (2022)).

Limited research has delved into the scaling dynamics of Language Models in the context of Question-Answering using RAG. Questions persist regarding the optimal embedding strategy, the most effective retrieval techniques, and the impact of various prompting methods on RAG's performance. This paper endeavors to shed light on these queries, offering insights into best practices and implementation techniques for a RAG-based architecture. Notably, the research also involves the application of these findings in the development of a knowledge-grounded response prediction system for a major retail company's Contact Center.

1.1 Industry Applications and Challenges

LLMs find extensive applications across various industries, with a particular focus on Contact Centers. These models play a crucial role not only in developing Chatbots but also in facilitating Agent-facing automation. A noteworthy example is the implementation of Response Prediction System (RPS), a powerful agent-assist solution. RPS enables the generation of contextually relevant responses, empowering agents to efficiently respond to customer queries with a simple click. This functionality not only boosts agent productivity but also elevates the overall customer experience while streamlining communication processes. In the industry setting, the emphasis is not solely on generating contextually apt response suggestions but also on ensuring the accuracy of the provided information, with minimal latency. The utilization of RAG-based



Hi

Visitor | 09:24am

Can I return this refridgerator?

Visitor | 09:26am

I'm sorry that it didn't work out for you. I can definitely help you with a return!

Based on our policy, appliances are returnable within 48 hours of delivery.

Figure 1: Example of Response Prediction System that can be used by Contact Center Agents

responses, grounded in company policies, proves highly effective in delivering swift and accurate resolutions to customer issues. As the LLMs sometimes tend to make up false information, it becomes very important to have a human in the loop (contact center agents in our case) to inhibit the leakage of factually incorrect information especially in the business setting. Additionally, it contributes to the reduction of training time for agents, offering quick access to the latest information available on company websites. Figure 1 shows use of RPS in Contact Centers

1.2 Related Work

LLMs have demonstrated significant capabilities but face challenges such as hallucination, outdated knowledge, and lack of transparency in reasoning processes. RAG Lewis et al. (2021) has emerged as a promising solution by incorporating knowledge from external databases to overcome these problems.These RAG models (Khandelwal et al. (2020); Borgeaud et al. (2022); Izacard et al. (2023); Yasunaga et al. (2023)),can retrieve knowledge from an external data store when needed, reducing hallucination and increasing coverage of grounded responses. Some of these approaches requires access to internal LM representation (eg, Izacard et al. (2023)) which might not be feasible since not all LLMs are open to public and fine tuning also may not be an option.

Despite the rapid developments in RAG research, there has been a lack of systematic consolidation in the community, which poses challenges in understanding the comprehensive landscape of RAG advancements. Recent study by Gao et al. (2024) divides the advancements in RAG into three categories: Naive/Traditional RAG, Advanced RAG and Modular RAG. The traditional RAG approach gained prominence shortly after the widespread adoption of ChatGPT. The approach follows a simple process that includes indexing, retrieval, and generation. Advanced RAG has been developed to address the shortcomings of traditional RAG. In terms of retrieval quality, Advanced RAG implements pre-retrieval and post-retrieval strategies. The modular RAG structure diverges from the traditional RAG framework, providing greater flexibility. It integrates various methods to enhance modules, such as using search for similarity retrieval and applying a fine-tuning method for the retriever (Liu et al. (2023)). Though all three approaches have their own pros and cons it's important to choose the methodology that works best for individual use cases.

Considering the traditional RAG approach the taxonomy of it's core components are Retriever, Generator, and Augmentation Method. To improve the semantic representation of the retriever, chunk optimization and fine tuning the embedding model (Dai et al. (2022),Zhang et al. (2023)) are some of the common approaches used in research. Query rewriting (Wang et al. (2023)) and embedding transformation (Li et al. (2023)) is used for aligning the queries and documents and in order to align the retriever and LLM, plugin adapters (Luo et al. (2023),Xu et al. (2023)) and LLM supervised training (Yu et al. (2023),Shi et al. (2023),Izacard et al. (2022)) is used. For the Generator component most research either perform post retrieval with frozen LLM (closed source LLM) like Xu et al. (2023), Ma et al. (2023),Zhuang et al. (2023) or they fine tune LLM (open source LLM) directly Cheng et al. (2023), Li et al. (2023). Though RAG has been a popular approach with a lot of advanced research focusing on improving either the retriever, generator or embedding components there is still shortcomings on which techniques works best for traditional RAG approach. Though there has been a lot of research in improving some of the short comings of traditional RAG with Advanced RAG and Modular RAG approaches, traditional RAG is still one of the most popular techniques used in industry due to ease of development and integration resulting in quicker speed to market.

The paper focuses on empirically answering these three research questions for the traditional RAG approach: **RQ1:** What are the effects of embedding technique and retrieval strategy in terms

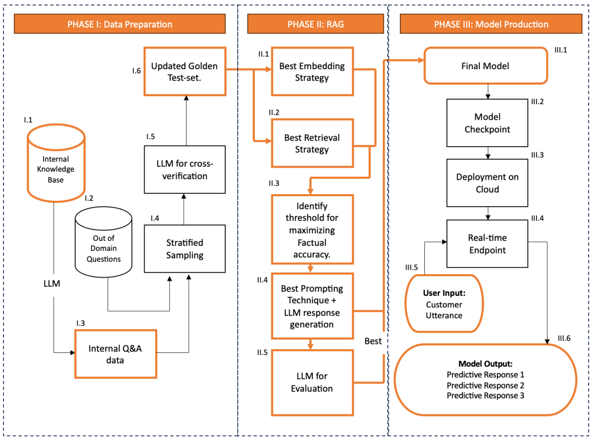


Figure 2: End to end RAG LLM framework

of improving the performance of RAG? **RQ2:** Can ReAct (Yao et al. (2023) help with improving the factual accuracy and reduce hallucination of LLMs? **RQ3:** What is the effect of different prompting techniques for generating grounded responses by LLM?

2 Methodology

2.1 Phase I: Data Preparation

An ideal golden test set for RAG architecture evaluation should have the following: 1. Domain (here, company documents) specific questions and their corresponding grounded responses. (I.3) 2. The data should also contain relevant Knowledge Base article that has the response to the Domain specific question (I.1) 3. Out of domain questions for the company documents should be included to make sure the LLM is not hallucinating while generating responses for these questions. Refer Figure 2. Firstly, Company Knowledge Base articles are used with LLM to generate questions and answer pairs. Out of domain questions are identified using LLM and sampled from open source data sets like MS-MARCO (Bajaj et al. (2018)). LLMs are further used downstream to make sure that the generated Question and Answers are available in the Knowledge Base article (I.5). Refer Section A for prompts.

2.2 Phase II: RAG

The main components of the RAG architecture are Retriever (II.2) and LLM for generator (II.4). **Embedding strategies:** The best embedding strategy (II.1) ensures high performance of retriever and it also affects downstream tasks like response generation. **Retrieval strategies:** By retrieving relevant passages or documents from a large corpus (II.2), the model gains contextual information, leading to more accurate and coherent responses. **LLM for generation:** Once the best embedding strategy and retrieval technique is identified, we test different prompting techniques to make sure LLMs are generating grounded factual responses (II.4). Additionally we also ran experiments on different retrieval threshold (II.3) to make sure incorrect documents are not retrieved and passed to the LLM to generate responses. LLMs (II.5) were also used for evaluating the performance of predicted response and compare it with the actual grounded response (golden test set). Criteria used for LLM evaluation are outlined in Section 4.2

Dataset	Unique Docs	Ave. Doc. Tokens	Unique Questions	Ave. Question Tokens	Unique Answers	Ave. Answer Tokens
MS-MARCO	4997	58.80	5000	5.67	4999	14.37
SQUAD	5000	94.21	4994	10.33	4413	2.59
TRIVIA	3530	4321.4	4087	12.95	3471	1.67

Table 1: Open source dataset random sample statistics

2.3 Phase III: Model Production

This phase starts with the best model (embedding, retrieval and prompting technique) from Phase II. In node III.2 the Knowledge Base articles, retriever etc are packaged together for the next step. Then the model package is deployed on a cloud Virtual Machine (III.3). To use it in real time an endpoint is created (III.4) that takes a customer query (III.5) and context of the conversation as input and generates response suggestions as the output (III.6)

3 Experimental Settings

3.1 Datasets and Models

The following open-source datasets were used for evaluation: A random subset of following open-source datasets **MS MARCO** Bajaj et al. (2018), **SQuAD** Rajpurkar et al. (2016), **TriviaQA** Joshi et al. (2017) , refer Table 1. Internal company data for evaluation is described in Section 2.1

Embedding Strategy: Compared Universal Sentence Encoder (USE) Cer et al. (2018) embeddings, Google's Vertex AI embedding model named text-embedding-gecko@001 Anil et al. (2023), and SBERT-all-mpnet-base-v2 Reimers and Gurevych (2019) from the sentence-transformers collection. **Retrieval Strategy:** We experimented with ScaNN (Research (2020)) and KNN HNSW (Malkov and Yashunin (2018)) on multiple data sets described. While ScaNN is designed to handle large-scale data sets, it might consume more memory due to quantization and reranking. KNN HNSW, on the other hand, is known for its efficient memory usage, which can be crucial for systems with limited memory. **Natural Language Generation:** The PaLM 2 Anil et al. (2023) for Text (`text-bison`, `text-unicorn`) foundation models are optimized for a variety of natural language tasks such as sentiment analysis, entity extraction, and content creation. Types of content that the PaLM 2 for Text models can generate include document summaries, answers to questions, and labels that classify content. At the time of writing this paper, we had a clearer path to production in terms of enterprise licenses and security requirements with text-bison@001 and text-embedding-gecko@001 as compared to other LLMs available. Also, at this time we have not fine-tuned the base model.

4 Results

4.1 Retrieval evaluation

Our objective here was to figure out the best combination of retriever and embedding strategy. For datasets mentioned in section 3.1, certain combinations stood out for their effectiveness. We observed a specific trend in Recall values in lower k values (1, 3) versus higher k values (5, 10) for SQuAD and TRIVIA. For SQuAD, Vertex AI - textembedding-gecko@001 (768) embedding with ScaNN retrieval performed the best at lower k but at higher k, SBERT-all-mpnet-base-v2 (768) with ScaNN performed better. For TRIVIA, SBERT-all-mpnet-base-v2 (768) embedding with HNSW KNN retrieval performed the best at lower k but at higher k, SBERT-all-mpnet-base-v2 (768) with ScaNN performed better. For MSMARCO, Vertex AI -textembedding-gecko@001 (768) embedding with ScaNN retrieval combination was a clear winner. Refer Table 2 for more details.

However, for our Company Data, the Vertex AI - textembedding-gecko@001 (768) embedding, used along with ScaNN retrieval, yielded the best outcomes, refer Table 3. Overall, ScaNN generally outshone KNN HNSW in most data sets for retrieval.

4.1.1 Retrieval Threshold

RAG based response and thus document retrieval is not needed for responding to out of domain question/statements made by customers in the RPS.

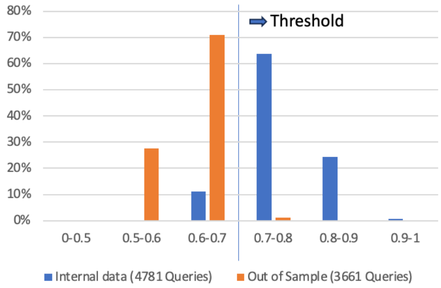


Figure 3: Cosine similarity score between query and ScaNN retrieved Document; retrieval threshold(0.7)

For these unrelated queries (questions irrelevant to Company Data), 98.59 percent of the retrieved articles had a cosine similarity score below 0.7. In contrast, 88.96 percent of the articles retrieved for the Company Data questions scored above 0.7. Hence, setting the threshold at 0.7 effectively helps us decide if retrieval is needed.

4.2 Generation Evaluation

4.2.1 Accuracy and Hallucination evaluation

We demonstrate the use of LLM based evaluation in this section. Word-level metrics like BLEU, METEOR, and ROUGE are utilized generally to measure the quality of the generated content. However, to assess the performance of text-bison@001, we can't rely solely on standard natural language generation (NLG) metrics because of following limitations: Lack of contextual and semantic understanding, Dependence on reference texts, inability to measure novelty or creativity, lack of flexibility in language use, inadequate for evaluating dialogue systems, insensitivity to content appropriateness, etc. Given these limitations, we used an LLM-based approach which as stated in the (Sun et al. (2023)), has 98 percent agreement with actual human judgements. We used this approach except for section 4.2.3. We prompted the LLM so that it could respond with one of three options: correct, incorrect, or unsure. If the generated response is both factually and semantically in line with the correct answer, the LLM labels it as “correct”. If there's a mismatch between the generated answer and the correct answer, it is labeled as “incorrect”. In cases where the LLM can't clearly categorize the response due to complex reasoning or semantic differences, it responds with “unsure”. Based on these labels, we calculated three metrics: Accuracy: The proportion of responses labeled 'correct' out of all tested samples. Hallucination Rate: The proportion of responses labeled 'incorrect' out of

Dataset	Embedding Strategy	Retrieval Strategy	Recall @ 1	Recall @ 3	Recall @ 5	Recall @ 10
SQUAD	USE (512)	HNSW KNN	0.4188	0.5946	0.6634	0.7424
SQUAD	SBERT - all-mpnet-base-v2 (768)	HNSW KNN	0.6708	0.8444	0.8902	0.9336
SQUAD	Vertex AI - textembedding-gecko@001 (768)	HNSW KNN	0.6958	0.8486	0.8804	0.911
SQUAD	USE (512)	ScaNN	0.4282	0.6116	0.6834	0.7666
SQUAD	SBERT - all-mpnet-base-v2 (768)	ScaNN	0.685	0.8636	0.913	0.9584
SQUAD	Vertex AI - textembedding-gecko@001 (768)	ScaNN	0.7156	0.874	0.908	0.9414
TRIVIA	USE (512)	HNSW KNN	0.459	0.6004	0.6604	0.7333
TRIVIA	SBERT - all-mpnet-base-v2 (768)	HNSW KNN	0.793	0.8691	0.8921	0.9171
TRIVIA	Vertex AI - textembedding-gecko@001 (768)	HNSW KNN	0.6423	0.7487	0.782	0.8233
TRIVIA	USE (512)	ScaNN	0.4101	0.6039	0.6687	0.7548
TRIVIA	SBERT - all-mpnet-base-v2 (768)	ScaNN	0.7086	0.8654	0.8992	0.9288
TRIVIA	Vertex AI - textembedding-gecko@001 (768)	ScaNN	0.5936	0.759	0.8038	0.8537
MS-MARCO	USE (512)	HNSW KNN	0.5263	0.6856	0.7347	0.784
MS-MARCO	SBERT - all-mpnet-base-v2 (768)	HNSW KNN	0.9128	0.9798	0.9878	0.9925
MS-MARCO	Vertex AI - textembedding-gecko@001 (768)	HNSW KNN	0.8194	0.9241	0.9425	0.9577
MS-MARCO	USE (512)	ScaNN	0.5347	0.6996	0.7518	0.8035
MS-MARCO	SBERT - all-mpnet-base-v2 (768)	ScaNN	0.9132	0.9816	0.9896	0.9944
MS-MARCO	Vertex AI - textembedding-gecko@001 (768)	ScaNN	0.8296	0.9376	0.9581	0.9738

Table 2: Recall@K for retrieval and embedding strategies for different data sets.

Dataset	Retrieval	Recall@1	Recall@3	Recall@5
Company	ScaNN	+0.0665	+0.0579	+0.0552

Table 3: With ScaNN retriever on Company Data, Vertex AI embeddings perform better than SBERT by 6.65% for recall@1 and by 5.79% for recall@3

K	Strategy	Accuracy	Hallucination Rate	Missing Rate
1	ReAct	-0.0213	+0.5140	-0.3438
3	ReAct	+0.0708	-0.1348	-0.1938

Table 4: Comparison (% diff.) between ReAct RAG and non-ReAct RAG NLG metrics on Company data

all tested samples. Missing Rate (when LLM is "unsure"): The proportion of responses labeled 'unsure' out of all tested samples. Refer Table 6. Generally, accuracy is better for k=3 than k=1 for the same data set and combination of embedding and retrieval methods except for TriviaQA. Also for TriviaQA, the rate of 'hallucinations' or incorrect answers jumps notably at k=3. The missing rate, doesn't vary between k=1 and k=3 compared to the hallucination rate. This indicates that it's a more consistent measure regardless of how many results are considered.

As observed in the Retrieval evaluation section, we see a similar relationship in the accuracy and hallucination as document size increases. From Table 1, we know that the token length of the TriviaQA documents is much larger than MS-MARCO and SQuAD. We observed lower accuracy and higher hallucination rates with TriviaQA when compared to MS-MARCO and SQuAD. We recommend chunking the documents into smaller sections to yield better results both from Retrieval and Generation perspectives in the RAG setup.

4.2.2 Experiments with ReAct

ReAct Yao et al. (2023) combines reasoning and action with LLMs. In our expirements with ReAct, ReAct Tools were used to decide when to use the information retrieval component within the RAG framework while using the same retrieval, embeddings, and generation strategy (ScaNN with text-embedding-gecko@001 and text-bison@001). Two scenarios were evaluated:"RAG with reAct" and another "RAG without reAct", both with k = 3. Refer Table 4, 7% increase in Accuracy Rate and a 13.5% decrease in the Hallucination Rate was observed with ReAct, however ReAct was much slower, refer Table 5

RAG Strategy	95th Percentile	99th Percentile
reAct	4.0942	6.2084
non-reAct	0.8850	1.1678

Table 5: Latency Comparison (seconds) between ReAct RAG and non-ReAct RAG based on 10000 queries

4.2.3 Response Prediction System evaluation

We compare the effectiveness of the current algorithm in production with our proposed RAG architecture. A set of 1000 real Contact Center chat

k	Data	Accuracy	Hallucination rate	Missing rate
1	SQUAD	84.74	8.36	6.9
1	TriviaQA	58.48	28.68	12.8
1	MS-MARCO	89.06	7.06	3.86
3	SQUAD	91.32	5.48	3.2
3	TriviaQA	25.08	67.41	7.46
3	MS-MARCO	89.9	6.93	3.16

Table 6: Generation quality metrics (%) using text-bison@001 and ScaNN

dataset	accuracy		hallucination_rate		missing_rate	
	baseline	CoTP	baseline	CoTP	baseline	CoTP
SQUAD	0.9814	0.9458	0.01	0.0416	0.0086	0.019
TRIVIA	0.6395	0.8627	0.2285	0.0645	0.1318	0.0638
MS-MARCO	0.921	0.9094	0.0464	0.0432	0.0326	0.0474

Table 7: Baseline vs CoTP evaluation metrics (%) without retrieval (question-doc. pair used as it is)

dataset	accuracy		hallucination_rate		missing_rate	
	baseline	CoVe	baseline	CoVe	baseline	CoVe
SQUAD	0.9814	0.9596	0.01	0.0324	0.0086	0.008
TRIVIA	0.6395	0.6376	0.2285	0.2383	0.1318	0.124
MS-MARCO	0.921	0.926	0.0464	0.0554	0.0326	0.0186

Table 8: Baseline vs CoVe evaluation metrics (%) without retrieval (question-doc. pair used as it is)

transcripts (PII and PCI compliant) which included over 5,000 messages (conversation turns) sent by agents in response to a variety of customer questions were used for comparison; a randomized human evaluation was also conducted. Evaluation metrics were grouped into three main categories. A) Qualitative Metrics: 1) Contextual Relevance: Assessed whether the predicted responses were appropriate and in line with the specific context of the conversation. 2) Completeness: Checked if the predicted responses were fully-formed and could be used as complete answers by the agents in specific parts of the conversation. 3) Specificity: Determined whether the predicted responses were tailored to the specific conversation being evaluated or if they were too general. Human labelers were asked to give a score of 0, 1, or 2; 0 being the lowest and 2 being the highest for these three metrics. B) Quantitative Metrics for Factual Accuracy(based on human evaluator judgment of 'correct', 'incorrect', or 'unsure'). Accuracy: Calculated as the number of correct responses divided by the total number of responses evaluated. Hallucination Rate: Measured as the number of incorrect responses divided by the total number of responses evaluated. C) Preference Metric: Evaluated which model's responses: "Production" or "RAG" were preferred by the human evaluators. Refer Table 9 for results.

Contextual Relevance	+0.4814
Specificity	+0.9797
Completeness	+0.7015
Accuracy	+0.4569
Hallucination Rate	-0.2749
Missing Rate	-0.7002
Preference	+2.0061

Table 9: Human evaluation comparison (% diff.) between RAG based responses and existing BERT-based ones. Note: Average scores were compared for qualitative metrics.

	Accuracy	Hallucination Rate	Missing Rate
CoVe	-0.4365	+2.7430	+11.3573
CoTP	-0.0345	-0.0133	+1.9878

Table 10: Evaluation metrics (% diff.) for answers using CoTP and CoVe relative to the Baseline answers

4.2.4 Prompting Techniques Experiments

We tested Chain of Verification (CoVe) Dhuliawala et al. (2023) and Chain of Thought Prompting (CoTP) Wei et al. (2023) to further improve factual accuracy and minimize hallucinations. However, implementing CoVe and CoTP in real-time is time-consuming, requiring multiple LLM calls per query. Moreover, we did not observe any improvement in accuracy and hallucination metrics on company data. High latency is unacceptable for our use-case where agents have to quickly respond to customer queries using the RPS. Generations Metrics for experiments on open-source datasets can be found in Table 7 and Table 8. Accuracy and hallucination rate improvement vary based on the open source dataset. However, no significant improvements were observed for the Company data (ref. section 2.1) with CoVe (43% less accurate) and CoTP (3% less accurate), refer Table 10, and hence, we decided against employing these prompting techniques. Refer Appendix A for prompts.

5 Conclusion

We demonstrate practical challenges of implementing RAG based Response Prediction System in production in an industry setting. We evaluated the performance of various retrieval, embedding strategy combined with different prompting techniques to identify which combinations works best for different use cases. We also show how ReAct might not be a practical solution for industry settings primarily due to it's latency issues. Experiments with CoTP and CoVe did not result in any improvement in metrics. During offline evaluation it was interesting to see that RAG suggested responses are 45.62% more accurate, 48.14% more relevant, 97.97% more specific and 70.15% more complete than production model responses (Table 9). By using RAG, hallucination rate goes down by 27.54%. In the future, we plan to work further in three directions. Firstly, extend this work to evaluate others LLMs. Secondly, test if Query rewriting and reformulation helps improve retrieval performance. Lastly, test advanced RAG approaches to combine Knowledge base from various sources.

References

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. Ms marco: A human generated machine reading comprehension dataset.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens.

Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder.

Xin Cheng, Di Luo, Xiuying Chen, Lemao Liu, Dongyan Zhao, and Rui Yan. 2023. Lift yourself up: Retrieval-augmented text generation with self memory.

Zhuyun Dai, Vincent Y. Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B. Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot dense retrieval from 8 examples.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Atlas: Few-shot learning with retrieval augmented language models.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models.

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. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Xiaoqian Li, Ercong Nie, and Sheng Liang. 2023. From classification to generation: Insights into crosslingual retrieval augmented icl.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts.

Ziyang Luo, Can Xu, Pu Zhao, Xiubo Geng, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023. Augmented large language models with parametric knowledge guiding.

Yubo Ma, Yixin Cao, YongChing Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples!

Yu. A. Malkov and D. A. Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Google Research. 2020. Scann: Efficient vector similarity search. `https://github.com/google-research/google-research/tree/master/scann`.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning.

Weijia Shi, Julian Michael, Suchin Gururangan, and Luke Zettlemoyer. 2022. knn-prompt: Nearest neighbor zero-shot inference.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. 2023. Replug: Retrieval-augmented black-box language models.

Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. 2023. Head-to-tail: How knowledgeable are large language models (llm)? a.k.a. will llms replace knowledge graphs?

Xintao Wang, Qianwen Yang, Yongting Qiu, Jiaqing Liang, Qianyu He, Zhouhong Gu, Yanghua Xiao, and Wei Wang. 2023. Knowledgpt: Enhancing large language models with retrieval and storage access on knowledge bases.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. Recomp: Improving retrieval-augmented lms with compression and selective augmentation.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models.

Michihiro Yasunaga, Armen Aghajanyan, Weijia Shi, Rich James, Jure Leskovec, Percy Liang, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. 2023. Retrieval-augmented multimodal language modeling.

Wenhao Yu. 2022. Retrieval-augmented generation across heterogeneous knowledge. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*, pages 52–58, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. 2023. Augmentation-adapted retriever improves generalization of language models as generic plug-in.

Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. 2023. Retrieve anything to augment large language models.

Shengyao Zhuang, Bing Liu, Bevan Koopman, and Guido Zuccon. 2023. Open-source large language models are strong zero-shot query likelihood models for document ranking.

A Appendix

Prompt for Chain of Prompting: Prompt for quote extraction: You are a reading comprehension and quote extraction expert. Please extract, word-for-word, any quotes relevant to the question. If there are no quotes in this document that seem relevant to the provided question, please say "I can't find any relevant quotes".

For document: doc and question: question, output:

Prompt for answer generation:

You are a reading comprehension and answer generation expert. Please answer the question from the document provided. If the document is not related to the question, simply reply: "Sorry, I cannot answer this question". Following are the guidelines you need to follow for generating the responses:

1) They should always be professional, positive, friendly, and empathetic. 2) They should not contain words that have a negative connotation (Example: "unfortunately"). 3) They should always be truthful and honest. 4) They should always be STRICTLY less than 30 words. If the generated response if greater than 30 words, rephrase and make it less than 30 words.

For document: document and question: question, output:

Prompt for Hallucination Judgement:

You need to check whether the prediction of a question-answering systems to a question is correct. You should make the judgement based on a list of ground truth answers provided to you. You response should be "correct" if the prediction is correct or "incorrect" if the prediction is wrong. Your response should be "unsure" where there is a valid ground truth and prediction is "Sorry, I don't know." or if you are not confident if the prediction is correct.

Below are the different cases possible:

1) Examples where you should return "correct".

Question: What is the customer registration process? Ground Truth: The customer registration process is a way for customers to create an account with them. This allows them to track their purchases, receive personalized offers, and more. The process is simple and can be completed in a few minutes. Prediction: The customer registration process is a process that allows customers to register their information with them. This process allows customers to receive benefits such as discounts, special offers, and personalized shopping experiences. Correctness: correct

Question: What happens if my refund is pending? Ground Truth: Sorry, I don't know. Prediction: Sorry, I don't know. Correctness: correct

2) Examples where you should return "incorrect".

Question: What do I need to do to get the military discount? Ground Truth: You need to have a smartphone and be registered for the discount. If you don't have a smartphone, you can use discount code RC5. If you are in the pilot 425 stores area, you can key in your phone number. Prediction: The military discount is available to active duty military members, veterans, and their families. The discount is 10 percent off eligible purchases. Correctness: incorrect

Question: How do I apply for the consumer card? Ground Truth: Sorry, I don't know. Prediction: You can apply for the consumer card in-store, online or by mail. Correctness: incorrect

3) Examples where you should return "unsure".

Question: What is the Return Policy? Ground Truth: The Return Policy is available on the website. You can find it by searching for "Return Policy" or by clicking on the link in the article. Prediction: Sorry, I don't know. Correctness: unsure

Provide correctness for the below question, ground truth and prediction:

Question: [question] Ground Truth: [ground truth] Prediction: [prediction] Correctness:

Prompt for Generating Baseline Response and Plan Verification (Chain of Verification):

Below is a question: [question]

Below is the document from which the answer should be generated: [document]

You are an subject matter expert working at Contact Centers. Your expertise includes quote extraction, answer generation, and asking verification questions to improve the overall factual accuracy of the answers you provide.

Your first goal is to extract, word-for-word, any quotes relevant to the question that could be used to answer the question. If there are no quotes in this document that seem relevant to the provided question, simply return: "I can't find any relevant quotes".

Your second goal is to use *solely* the quotes extracted from the first goal and generate a concise and accurate answer (using the below listed guideline) by rephrasing the quotes to answer the question. If the quotes could not be used to answer the question, simply return: "Sorry, I cannot answer this question". 1) They should always be professional, positive, friendly, and empathetic. 2) They should not contain words that have a negative connotation (Example: "unfortunately"). 3) They should always be truthful and honest. 4) They should always be STRICTLY less than 30 words. If the generated response if greater than 30 words, rephrase and make it less than 30 words.

Your third goal is to generate a list of potential areas that might require verification based on the content of the document to increase factual accuracy of the answer. Your response should be in the below format:

"' Quotes: [Your Extracted Quotes]

Answer: [Your Answer]

Potential Areas for Verification: 1) Your Specific point or segment from your answer. 2) Your Another point or segment from your answer. N) Your Nth point or segment from your answer. "'

Prompt for Executing Verification Questions and Generating Verified Response (Chain of Verification):

Below is a question: [question]

Below is the answer: [answer]

Below is the document from which the answer was generated: [document]

Based on the potential areas for verification: [areas of verification]

You are an subject matter expert working at Contact Centers. Your expertise includes improvising answers to questions about the company to increase factual correctness using the factual accuracy verification questions provided to you.

Your goal is to check each verification point against the document, provide feedback on any inconsistencies, and then generate a final verified (using the below listed guidelines), concise and accurate answer in strictly less than 30 words that addresses the factual inconsistencies. 1) They should always be professional, positive, friendly, and empathetic. 2) They should not contain words that have a negative connotation (Example: "unfortunately"). 3) They should always be truthful and honest. 4) They should always be STRICTLY less than 30 words. If the generated response if greater than 30 words, rephrase and make it less than 30 words.

Your response should be in the below format:

“‘ Feedback: 1) Your Verification for point 1. 2) Your Verification for point 2. N) Your Verification for point N.

Final Verified Response: [Your Revised Response] “‘

Prompt for generating answer from a document and question.

You are a question answering bot. Your job is to generate answer to the question using the provided articles. The answers should be derived only from the articles. If the answer is not present in the articles, return the text - NOANSWERFOUND. The answer should be less than 10 words and in a sentence format.

Example where answer could not be found in the articles: Question: Which county is Smyrna city in? Document: Georgia is a southeastern U.S. state whose terrain spans coastal beaches, farmland and mountains. Capital city Atlanta is home of the Georgia Aquarium and the Martin Luther King Jr. National Historic Site, dedicated to the African-American leader’s life and times. Return Text: NOANSWERFOUND

Example where answer could be found in the articles: Question: Which county is Smyrna city in? Document: Smyrna is a city in Cobb County, Georgia, United States. Cobb County is a county in the U.S. state of Georgia, located in the Atlanta metropolitan area in the north central portion of the state. Return Text: Cobb County of the state of Georgia

Provide answer to the below Question/Query using the below Document.

Question: [question] Document: [document] Return Text: