RAG based Question-Answering for Contextual Response Prediction System

Sriram Veturi*
Saurabh Vaichal*
sriram_veturi@homedepot.com
saurabh_s_vaichal@homedepot.com
The Home Depot
Atlanta, Georgia, USA

Nafis Irtiza Tripto†
The Pennsylvania State University
USA
nit5154@psu.edu

Reshma Lal Jagadheesh‡
The Home Depot
Atlanta, Georgia, USA
reshma_lal_jagadheesh@homedepot.com

Nian Yan
The Home Depot
Atlanta, Georgia, USA
nian_yan@homedepot.com

Abstract

Large Language Models (LLMs) have shown versatility in various Natural Language Processing (NLP) tasks, including their potential as effective question-answering systems. However, to provide precise and relevant information in response to specific customer queries in industry settings, LLMs require access to a comprehensive knowledge base to avoid hallucinations. Retrieval Augmented Generation (RAG) emerges as a promising technique to address this challenge. Yet, developing an accurate question-answering framework for real-world applications using RAG entails several challenges: 1) data availability issues, 2) evaluating the quality of generated content, and 3) the costly nature of human evaluation. In this paper, we introduce an end-to-end framework that employs LLMs with RAG capabilities for industry use cases. Given a customer query, the proposed system retrieves relevant knowledge documents and leverages them, along with previous chat history, to generate response suggestions for customer service agents in the contact centers of a major retail company. Through comprehensive automated and human evaluations, we show that this solution outperforms the current BERT-based algorithms in accuracy and relevance. Our findings suggest that RAG-based LLMs can be an excellent support to human customer service representatives by lightening their workload.

CCS Concepts

• **Computing methodologies → Machine learning**.

Keywords

Retrieval Augmented Generation, Response Prediction System, Question Answering System, Contact Center Agents, Automated Hallucination Measurement, Hallucination Reduction, Retrieval Strategies, Optimal Retriever Threshold, ScaNN, Embedding Strategies, Contextual Relevance, Specificity, Completeness, Hallucination Rate, Missing Rate, Human Evaluation of RAG Versus Traditional Seq-2-Seq models, RAG Deployment.

ACM Reference Format:
Sriram Veturi, Saurabh Vaichal, Nafis Irtiza Tripto, Reshma Lal Jagadheesh, and Nian Yan. 2024. RAG based Question-Answering for Contextual Response Prediction System. In *Proceedings of CIKM 2024 (1st Workshop on GenAI and RAG Systems for Enterprise)*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

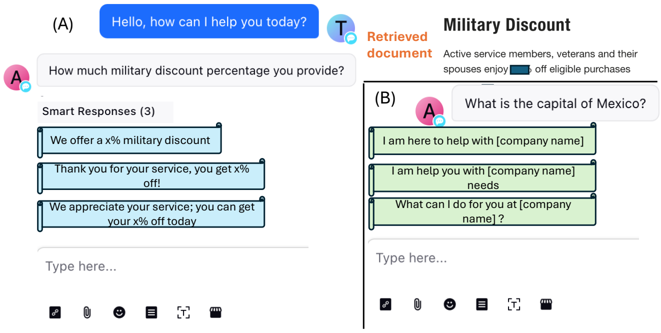
With the advent of ChatGPT and similar tools in mainstream media, Large Language Models (LLMs) have emerged as the standard solution for addressing a wide range of language understanding tasks. However, they can generate incorrect or biased information [34], as their responses are based on patterns learned from data that may not always contain necessary knowledge in a close domain. To address this issue, Retrieval Augmented Generation (RAG) [20] is commonly used to ground LLMs in factual information. The RAG architecture processes user input by first retrieving a set of documents similar to the query, which the language model then uses to generate a final prediction. While RAG-based architectures have been successful in various open-domain question answering (Q/A) tasks [15, 30, 49], limited research has explored their scaling dynamics in real conversational scenarios. Therefore, our research is one of the pioneering efforts in exploring the feasibility of an RAG-based approach for developing a knowledge-grounded response prediction system specifically tailored for the contact center of a major retail company.

LLMs have recently been widely adopted across various industries, particularly in contact centers, to enhance chatbot development and agent-facing automation [8, 12, 37]. A prime example is the Response Prediction System (RPS), an agent-assist solution that generates contextually relevant responses, enabling agents

[*] Both first authors contributed equally to this research.
[†] Work completed during internship at The Home Depot
[‡] Work completed during employment at the Home Depot

1st Workshop on GenAI and RAG Systems for Enterprise, October 24, 2024, Boise, Idaho, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn



Hello, how can I help you today?

How much military discount percentage you provide?

We offer a x% military discount

Thank you for your service, you get x% off!

We appreciate your service; you can get your x% off today

Military Discount

Active service members, veterans and their spouses enjoy [redacted] off eligible purchases

What is the capital of Mexico?

I am here to help with [company name]

I am help you with [company name] needs

What can I do for you at [company name] ?

Figure 1: Example of the Response Prediction System. (A): For a valid query, the system retrieves the relevant document and proposes the appropriate responses from where the agent choose. (B): For an out-of-domain query, it guides the user to ask a relevant question.

to efficiently address customer queries with a single click. This boosts productivity, improves customer experience, and streamlines communication processes. In industry settings, the focus is on generating accurate, contextually appropriate responses with minimal latency. Therefore, RAG-based responses, grounded in company policies, deliver swift and accurate resolutions to customer issues. Figure 1 demonstrates a possible example of RPS in real settings, where the agent can directly utilize the generated response with a single click.

However, implementing RAG for industry-specific use cases to assist human agents in generating valid responses involves several architectural decisions that can affect performance and viability. The retrieval style can be integrated into both encoder-decoder [16, 44]) and decoder-only models [5, 19, 26, 27], with various embedding and prompting techniques influencing the final LLM output. In contact centers, where the risk of hallucinations is high and can critically impact business performance, ReAct (Reason+Act) [42] prompts can help mitigate issues. Therefore, our research focuses on developing an optimal RAG based knowledge-grounded RPS for a major retail company's contact center. To ensure response accuracy, we also conduct thorough evaluations with human evaluators and automated measures, comparing RAG-based responses to human ground truth and the existing BERT-based system (Figure 2 shows an overview of traditional customer care scenario with existing and proposed system). In short, we answer the following research questions.

(1) **RQ1:** What are the effects of different embedding techniques, retrieval strategies, and prompting methods on RAG performance?
(2) **RQ2:** Do RAG-based responses provide greater assistance to human agents compared to the existing BERT-based system?
(3) **RQ3:** Can the ReAct (Reason+Act) prompting improve factual accuracy and reduce hallucinations in LLM in real-time settings?

Our findings demonstrate an overall improvement over the existing system by suggesting more accurate and relevant responses, highlighting the potential of RAG LLM as an excellent choice for customer care automation.

2 Related Work

RAG architecture: RAG has emerged as a promising solution by incorporating knowledge from external databases to overcome the hallucination, outdated knowledge, transparency issues for LLMs [5, 17, 19, 20, 43]. Traditional RAG, popularized after the adoption of ChatGPT, follows a simple process of indexing, retrieval, and generation [14] . Despite advancements in Advanced and Modular RAG, Traditional RAG remains popular in the industry due to its ease of development, integration, and quicker speed to market [21]. The core components of Traditional RAG include Retriever, Generator, and Augmentation Method, with research focusing on improving semantic representation [9, 48], query alignment [35], and integration with LLMs [16, 28, 45], which motivate our RQ1 to find the optimum setup in this specific use case of RPS in the contact center.

RAG LLM for question answering: Several open-domain question-answering (Q/A) tasks have been completed by RAG-based architectures efficiently [15, 30, 49]. With the advent of LLMs in recent periods, multiple studies also focus on utilizing LLMs for customer assistance, specifically in recommendations [13, 41] and dialogue generation [29, 47]. Recent work by [2] proposes augmenting LLMs with user-specific context from search engine interaction histories to personalize outputs, leveraging entity-centric knowledge stores derived from users' web activities. Similarly, [40] introduces a customer service question-answering approach integrating RAG with a knowledge graph (KG) constructed from historical issue data. Therefore, our study is motivated by these prior researches to integrate RAG as a retrieval tool and utilize LLM to generate responses to answer customer queries.

3 Methodology

To implement an end-to-end RAG framework with LLM, first, it is essential to create a comprehensive dataset comprising relevant question-answer pairs along with corresponding knowledge documents. Next, design choices for specific components of the RAG and LLM architecture must be finalized. Finally, the model should be thoroughly evaluated and refined before being deployed into production.

3.1 Phase I: Data Preparation

An ideal golden dataset for evaluating RAG architecture (Figure 3) should include:

(1) Domain-specific questions (previous queries) with their corresponding grounded responses.
(2) Relevant knowledge base (KB) articles (company documents) containing the policies that determine answers to specific queries.
(3) Out-of-domain questions to ensure the LLM can handle generic queries without hallucinating and can guide customers to provide relevant queries.

To create a robust test set (Table 1 for details), we utilize LLM to generate both relevant question-answer pairs from the company's

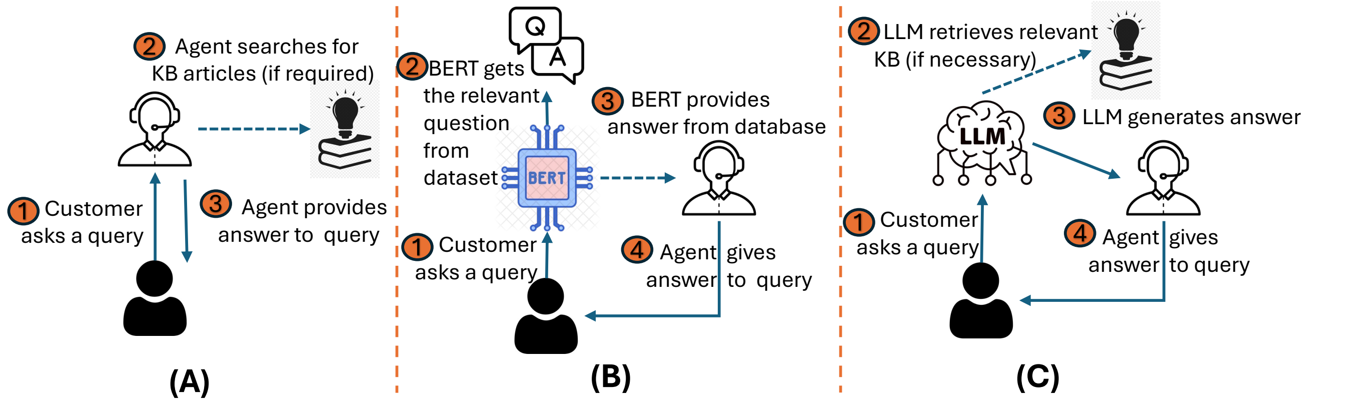

2 Agent searches for KB articles (if required)
1 Customer asks a query
3 Agent provides answer to query
(A)
2 BERT gets the relevant question from dataset
Q
A
BERT
3 BERT provides answer from database
1 Customer asks a query
4 Agent gives answer to query
(B)
2 LLM retrieves relevant KB (if necessary)
LLM
3 LLM generates answer
1 Customer asks a query
4 Agent gives answer to query
(C)

Figure 2: Overview of the systems: (A) Agents respond to queries by manually searching for relevant documents, (B) The existing BERT-based system, which extracts relevant Q/A pairs from the given query and provides suggested answers to the agents, (C) The proposed RAG LLM system, where the LLM retrieves relevant KB articles (if necessary) and generates answers based on the query and the retrieved articles.

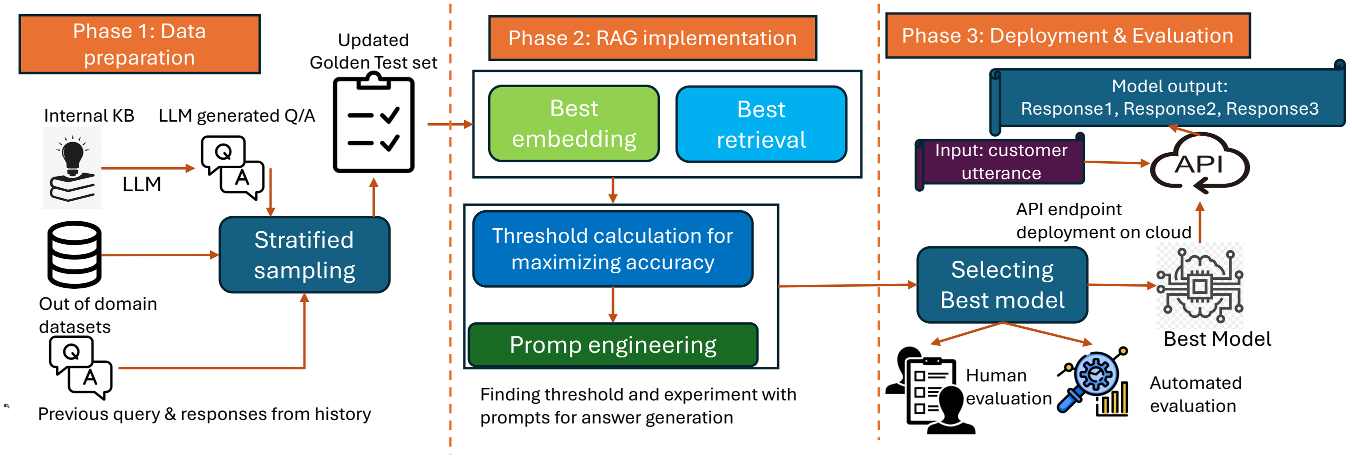

Phase 1: Data preparation
Updated Golden Test set
Internal KB
LLM generated Q/A
LLM
Q
A
Stratified sampling
Out of domain datasets
Previous query & responses from history
Phase 2: RAG implementation
Best embedding
Best retrieval
Threshold calculation for maximizing accuracy
Promp engineering
Finding threshold and experiment with prompts for answer generation
Phase 3: Deployment & Evaluation
Model output: Response1, Response2, Response3
Input: customer utterance
API
API endpoint deployment on cloud
Selecting Best model
Best Model
Human evaluation
Automated evaluation

Figure 3: End to end RAG LLM framework

Source	Total #	length (query)	length (response)
KB articles	1205	Avg. document length: 134.25	
In domain Q/A (generated by LLM from KB articles)	4785	10.7	33.59
In domain Q/A (sampled from previous history)	3000	9.58	28.81
Out domain Q/A (sampled from MS-MARCO)	3660	5.73	5

Table 1: Total number and average length (in terms of word count) for the KB articles and various Q/A pairs for the RAG implementation

KB articles (refer to Section A in the Appendix for prompts). Additionally, we supplement these relevant pairs with samples from previous queries & responses in the contact center with out-of-domain questions by sampling from open-source datasets such as MS-MARCO [3].

3.2 Phase II: RAG

The main components of the RAG architecture are the Retriever and the Generator LLM. We evaluate various strategies for each component and finalize our choices for production. Our findings are validated through experiments with several open-domain question-answer datasets, including MARCO [3], SQuAD [23], and TriviaQA [18] (details in Appendix). They present a comparable level of question-answering challenges where answers can be derived from the retrieved knowledge base.

Embedding Strategy: The best embedding strategy ensures high performance of the retriever and affects downstream tasks like response generation. We compare the Universal Sentence Encoder

(USE) embeddings [6], Google's Vertex AI embedding model text-embedding-gecko@001 [?], and SBERT-all-mpnet-base-v2 [24] from the sentence-transformers collection.

Retrieval strategies: By retrieving relevant passages from a large corpus of KB articles, the model gains crucial contextual information, enhancing response accuracy and coherence. We specifically consider ScaNN [25] for its efficiency in handling large-scale datasets and KNN HNSW [22] for its efficient memory usage as retrieval strategies in our study. Additionally, we tested different retrieval thresholds to ensure incorrect documents are not retrieved and passed to the LLM for response generation.

LLM for generation: Once the best embedding strategy, retrieval technique, and retrieval thresholds are identified, we test different prompting techniques to ensure that LLMs generate grounded factual responses. We utilize PaLM2 foundation models (text-bison, text-unicorn) [1] for text generation across all tasks, as they offer a clear path to production in terms of enterprise licenses and security requirements with Google's models, compared to other available LLMs at the time of our research.

The best model from Phase 2, incorporating the optimal embedding, retrieval, and prompting techniques, is packaged with relevant KB articles and deployed on a cloud Virtual Machine. For real-time usage, an endpoint is created that takes a customer query and the conversation context as input, generating response suggestions as output.

4 Results and Findings

First, we optimize RAG setup for our use cases, then evaluate LLM responses using automated metrics and human evaluations. Finally, we assess if prompting or ReAct strategies can improve real-world performance to an acceptable level.

4.1 Retrieval evaluation

Best setting for RAG. :

We assessed retriever efficiency using the "Recall at K" ($R@k$) metric, where K represents the top 1, 3, 5, or 10 documents retrieved, measuring how well the retriever retrieves relevant documents. The Vertex AI - textembedding-gecko@001 (768) embedding, paired with ScaNN retrieval, yielded the best outcomes. Overall, ScaNN generally outperformed KNN HNSW in most cases due to its efficient handling of large-scale datasets and superior retrieval accuracy through quantization and re-ranking techniques [7], so we include only the ScaNN results in Table 2. Similarly, Vertex AI embeddings surpassed Sentence BERT and USE due to its superior ability to capture complex semantic relationships tailored for large-scale industry applications.

Retrieval Threshold: For out-of-domain or trivial customer queries like "Hello" or "Bye," document retrieval is unnecessary, as shown by 98.59% of retrieved articles having a cosine similarity score below 0.7. In contrast, 88.96% of articles retrieved for relevant company data questions scored above 0.7 (Figure 4). This suggests that setting the retrieval threshold at 0.7 effectively determines when retrieval is needed, thereby enhancing response generation efficiency.

Embedding	Embedding size	R@1	R@3	R@5
USE	512	-	-	-
SBERT	768	+15.36	+9.42	+8.22
Vertex AI	768	+21.55	+13.87	+11.85

Table 2: Performance of different embedding technique in the company data for ScaNN. Performance is shown as the (%) of improvement wrt the lowest performing embedding (USE in this scenario)

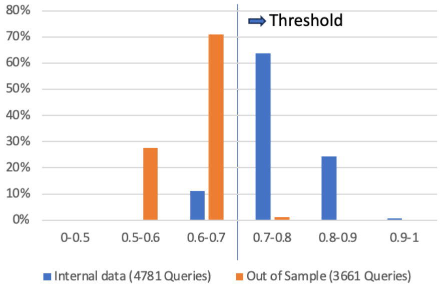


Figure 4: Cosine similarity score between query and ScaNN retrieved Document; retrieval threshold(0.7)

Metric	Improvement
Accuracy	+10.15
Hallucination	-4.76
Missing rate	-5.43
AlignScore	+5.6
Semantic similarity	+20.01
AI-generated	-40.17

Table 3: Comparision between RAG based responses and existing BERT-based ones for automated evaluations. Values indicate the difference in percentage (%) as average of all samples

4.2 Response Prediction System Evaluation

To develop an effective Response Generation System (RPS), we conducted a comprehensive evaluation comparing RAG LLM-based responses with a current BERT-based algorithm. Using 1,000 real contact center chat transcripts (PII and PCI compliant), comprising over 5,000 messages, we analyzed customer queries, human agent responses, RAG LLM suggestions, BERT-based suggestions, and retrieved knowledge base documents to assess quality, consistency, and factuality through automated measures and human evaluations.

4.2.1 Automated evaluations. We utilize the following evaluation techniques, with Table 3 illustrating our RAG LLM-based technique's performance against the current BERT-based system.

Accuracy, Hallucination and Missing rate evaluation. In a question-answer system, a response to each query can generate one of three types of responses: accurate (correctly answers the question), hallucinate (incorrect answer), or missing (no answer generated). Therefore, our approach, inspired by [32] which provides 98% agreement with human judgments, utilizes an LLM-based method. We employ ChatGPT-3.5-turbo as our evaluator LLM. We prompted the LLM with a query, generated response, and original human response, categorizing the LLM's responses as "correct" for factual and semantic alignment, "incorrect" for mismatches, and "unsure" for semantic challenges. Evaluation includes Accuracy (correct responses), Hallucination Rate (incorrect responses), and Missing Rate (unsure responses) metrics as the proportion of corresponding responses. Overall, RAG LLM improves accuracy by reducing hallucinations and missing rates compared to BERT responses.

AlignScore: To ensure response alignment with KB articles, we use AlignScore [46] to measure information consistency. Evaluating RAG LLM and BERT-based models on utterances with relevant KB article retrieval by RAG, RAG LLM shows a statistically significant 5.6% improvement via Student's t-test. This enhancement derives from integrating retrieved documents as prompts for LLM responses, whereas BERT relies on query-answer pairs in its training dataset.

Semantic similarity: To ensure usability by human agents, coherence between generated and original human responses is crucial. We measure semantic similarity using LongFormer embeddings [4], calculating cosine similarity between generated and original human responses for both models. RAG LLM exhibits an average 20% higher similarity, a statistically significant improvement.

Human touch: Customer service are typically preferred to be handled by humans [11, 38], emphasizing the importance of generating human-like responses. We use the AI text detector GPTZero [33], with a 99.05% true positive rate for human responses in our dataset, to evaluate response naturalness. Assessing AI percentage (utterances identified as AI-generated), the BERT-based system, which selects responses from human-generated options, sounds more human.

4.2.2 *Human evaluations.* Our method aims to support rather than replace humans through a human-in-the-loop approach. We thoroughly evaluate the quality of RAG LLM and BERT responses using human annotators. Each response is assessed against several criteria, and the average score is computed from all annotators' evaluations. Evaluation metrics were grouped into three main categories:

Human Preference Score: Following the classical approach of which version humans prefer most [31, 39], we evaluated which model's responses—"BERT" or "RAG"—were preferred by human evaluators.

Quantitative Metrics: Similar to [32], we evaluated factual accuracy (based on human judgment of 'correct,' 'incorrect,' or 'unsure'). Accuracy, Hallucination, and Missing rates were calculated as the number of correct, incorrect, and unsure responses divided by the total number of responses evaluated, respectively.

Qualitative Metrics:

(1) **Contextual Relevance:** Assessed whether the predicted responses were appropriate and in line with the 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 or were too general. Human annotators scored these metrics on a scale of 0 (lowest) to 2 (highest).

Metric	Improvement
Contextual Relevance	+48.14
Specificity	+97.97
Completeness	+70.15
Accuracy	+45.69
Hallucination Rate	-27.49
Missing Rate	-70.02
Preference	+200.61

Table 4: Human evaluation comparison (% diff.) between RAG and existing BERT-based ones.

K	Strategy	Accuracy	Hallucination Rate	Missing Rate
1	**ReAct**	-2.13	+51.40	-34.38
3	**ReAct**	+7.08	-13.48	-19.38
3	**CoVe**	-43.65	+27.43	+11.35
3	**CoTP**	-3.45	+1.33	+1.98

Table 5: Comparison (% diff.) of ReAct RAG (with different values of k), CoVe and CoTP performance with respect to baseline on company data.

The results, as detailed in the Table 4, Responses generated by the RAG model demonstrated a 45% improvement in factual accuracy and a 27% decrease in the rate of hallucinations compared to the existing model. Moreover, the human evaluators favored responses from the RAG model over the current production model 75% of the times.

The Response Prediction System was deployed using Flask, a standard micro web framework, and Gunicorn, chosen for its performance, flexibility, and simplicity in production system configuration. The API receives customer queries as input and provides answers as output. The API was thoroughly load tested using Locust, an open-source performance/load testing tool for HTTP and other protocols, to ensure it meets real-time latency requirements in a production setting. Finally, the API was integrated with the Agent Workspace UI to deliver predictions to Contact Center agents, assisting customers in real-time.

4.3 Evaluation for ReAct and prompting techniques

Experiments with ReAct. To answer our third RQ, we utilized ReAct Tools to determine when to activate the information retrieval component within the RAG framework, while maintaining the same retrieval, embeddings, and generation strategies. We evaluated two scenarios: "RAG with ReAct" and "RAG without ReAct," with K = 3. As shown in Table 5. While ReAct improved the accuracy by 7% and reduced hallucination by 13.5%, it resulted in slower performance 6, making it inconvenient in real-time conversation.

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

Table 6: Latency Comparison (seconds) between ReAct RAG and non-ReAct RAG based on 10000 queries

Prompting Techniques Experiments. We evaluated Chain of Verification (CoVe) [10] and Chain of Thought Prompting (CoTP) [36] to improve factual accuracy and reduce hallucinations. However, both techniques are time-consuming, requiring multiple LLM calls per query, and did not show significant improvements for the Company data. CoVe was 43% less accurate and CoTP was 3% less accurate (see Table 5). Therefore, we decided against using these prompting techniques.

5 Conclusion

In this study, we demonstrate the practical challenges of implementing a RAG-based Response Prediction System in an industry setting. We evaluated various retrieval and embedding strategies combined with different prompting techniques to identify the best combinations for different use cases. Our evaluations show that retrieving relevant knowledge base articles and generating responses from LLMs can be more contextually relevant and accurate than BERT responses, which choose from the most relevant query-answer pairs. We also highlight that ReAct and advanced prompting techniques may not be practical for industry settings due to latency issues. Overall, our approach indicates that implementing RAG-based LLM response generation for contact centers is feasible and can effectively aid humans, reducing their workload. In the future, we plan to advance our work in three directions. Firstly, we aim to evaluate other LLMs. Secondly, we will test if query rewriting and reformulation can improve retrieval performance. Lastly, we intend to explore advanced RAG approaches to integrate knowledge bases from various sources.

Limitations

Despite ongoing significant research, LLMs remain unpredictable. Though this paper showcases work on grounding the generated responses, LLMs to a certain degree are still capable of generating inaccurate information based on their learnt parametric memory. This work also does not focus on other LLM issues such as context length constraints, prompt injections and quality of Knowledge base data. Addressing other open challenge such as biases along with their ethical consideration are also not considered in the scope of this paper. The paper also does not address or evaluate RAG for multilingual data sources.

Ethics Statement

The data set used for training and validation of LLMs in this paper do not have any unbalanced views or opinions of individuals that might bias the generated response. The LLMs are still capable of generating inaccurate responses based their parametric memory even when relevant contextual information might be provided. Filtering toxic responses and prompt injections are also not considered in this evaluation.

References

[1] 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. arXiv:2305.10403 [cs.CL]

[2] Jinheon Baek, Nirupama Chandrasekaran, Silviu Cucerzan, Allen Herring, and Sujay Kumar Jauhar. 2024. Knowledge-augmented large language models for personalized contextual query suggestion. In *Proceedings of the ACM on Web Conference 2024*. 3355–3366.

[3] 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. arXiv:1611.09268 [cs.CL]

[4] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).

[5] 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. arXiv:2112.04426 [cs.CL]

[6] 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. arXiv:1803.11175 [cs.CL]

[7] Wei-Cheng Chang, Jyun-Yu Jiang, Jiong Zhang, Mutasem Al-Darabsah, Choon Hui Teo, Cho-Jui Hsieh, Hsiang-Fu Yu, and SVN Vishwanathan. 2024. PEFA: ParamEter-Free Adapters for large-scale embedding-based retrieval models. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 77–86.

[8] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132* (2024).

[9] 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. arXiv:2209.11755 [cs.CL]

[10] 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. arXiv:2309.11495 [cs.CL]

[11] Teresa Fernandes and Elisabete Oliveira. 2021. Understanding consumers' acceptance of automated technologies in service encounters: Drivers of digital voice assistants adoption. *Journal of Business Research* 122 (2021), 180–191.

[12] Samuel Kernan Freire, Chaofan Wang, and Evangelos Niforatos. 2024. Chatbots in knowledge-intensive contexts: Comparing intent and llm-based systems. *arXiv preprint arXiv:2402.04955* (2024).

[13] Luke Friedman, Sameer Ahuja, David Allen, Zhenning Tan, Hakim Sidahmed, Changbo Long, Jun Xie, Gabriel Schubiner, Ajay Patel, Harsh Lara, et al. 2023. Leveraging large language models in conversational recommender systems. *arXiv preprint arXiv:2305.07961* (2023).

[14] 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. arXiv:2312.10997 [cs.CL]

[15] Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282* (2020).

[16] 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. arXiv:2208.03299 [cs.CL]

[17] 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 (2023), 1–43. http://jmlr.org/papers/v24/23-0037.html

[18] 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)*, Regina Barzilay and Min-Yen Kan (Eds.). Association for Computational Linguistics, Vancouver, Canada, 1601–1611. https://doi.org/10.18653/v1/P17-1147

[19] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through Memorization: Nearest Neighbor Language Models. arXiv:1911.00172 [cs.CL]

[20] 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. arXiv:2005.11401 [cs.CL]

[21] 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. arXiv:2307.03172 [cs.CL]

[22] Yu. A. Malkov and D. A. Yashunin. 2018. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. arXiv:1603.09320 [cs.DS]

[23] 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*, Jian Su, Kevin Duh, and Xavier Carreras (Eds.). Association for Computational Linguistics, Austin, Texas, 2383–2392. https://doi.org/10.18653/v1/D16-1264

[24] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084 [cs.CL]

[25] Google Research. 2020. ScaNN: Efficient Vector Similarity Search. https://github.com/google-research/google-research/tree/master/scann.

[26] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning To Retrieve Prompts for In-Context Learning. arXiv:2112.08633 [cs.CL]

[27] Weijia Shi, Julian Michael, Suchin Gururangan, and Luke Zettlemoyer. 2022. kNN-Prompt: Nearest Neighbor Zero-Shot Inference. arXiv:2205.13792 [cs.CL]

[28] 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. arXiv:2301.12652 [cs.CL]

[29] Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, et al. 2022. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *arXiv preprint arXiv:2208.03188* (2022).

[30] Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering. *Transactions of the Association for Computational Linguistics* 11 (2023), 1–17.

[31] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems* 33 (2020), 3008–3021.

[32] 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? arXiv:2308.10168 [cs.CL]

[33] Edward Tian. 2023. GPTZero. Online; accessed 23-Mar-2023. https://gptzero.me/

[34] SM Tonmoy, SM Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313* (2024).

[35] 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. arXiv:2308.11761 [cs.CL]

[36] 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. arXiv:2201.11903 [cs.CL]

[37] Andreas Wortmann, Olivier Barais, Benoit Combemale, and Manuel Wimmer. 2020. Modeling languages in Industry 4.0: an extended systematic mapping study. *Software and Systems Modeling* 19 (2020), 67–94.

[38] Laurie Wu, Alei Fan, Yang Yang, and Zeya He. 2022. Tech-touch balance in the service encounter: The impact of supplementary human service on consumer responses. *International Journal of Hospitality Management* 101 (2022), 103122.

[39] Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. 2023. Human preference score: Better aligning text-to-image models with human preference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2096–2105.

[40] 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. *arXiv preprint arXiv:2404.17723* (2024).

[41] Fan Yang, Zheng Chen, Ziyan Jiang, Eunah Cho, Xiaojiang Huang, and Yanbin Lu. 2023. Palr: Personalization aware llms for recommendation. *arXiv preprint arXiv:2305.07622* (2023).

[42] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. arXiv:2210.03629 [cs.CL]

[43] 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. arXiv:2211.12561 [cs.CV]

[44] 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*, Daphne Ippolito, Liunian Harold Li, Maria Leonor Pacheco, Danqi Chen, and Nianwen Xue (Eds.). Association for Computational Linguistics, Hybrid: Seattle, Washington + Online, 52–58. https://doi.org/10.18653/v1/2022.naacl-srw.7

[45] Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. 2023. Augmentation-Adapted Retriever Improves Generalization of Language Models as Generic Plug-In. arXiv:2305.17331 [cs.CL]

[46] Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating Factual Consistency with A Unified Alignment Function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 11328–11348.

[47] Kai Zhang, Fubang Zhao, Yangyang Kang, and Xiaozhong Liu. 2023. Memory-augmented llm personalization with short-and long-term memory coordination. *arXiv preprint arXiv:2309.11696* (2023).

[48] Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. 2023. Retrieve Anything To Augment Large Language Models. arXiv:2310.07554 [cs.IR]

[49] Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, and Bin Cui. 2024. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473* (2024).

A Evaluation of RAG Approach for Open Source Datasets

To evaluate the effectiveness of the RAG-based approach, we also conducted a sample study using three open-domain datasets following a similar methodology.

A.1 Dataset Statistics

We consider three popular open source question answering datasets focusing on a varitey of topics. A random subset of **MS MARCO**[3], **SQuAD** [23], **TriviaQA** [18] were considered for the evaluation. Table 7 shows the brief overview of the considered datasets.

Dataset	Unique Docs	Ave. Doc. Tokens	Unique Quest-ions	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 7: Open source dataset random sample statistics

A.2 Retrieval and Embedding Performance

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, SBERTall-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 8 for more details.

A.3 Evaluation of Generated Responses

Table 9 shows the accuracy, hallucination, and missing rate for the open sources datasets (through automated evaluation as described in Subsection 4.2.1. As observed in the Retrieval evaluation section, we notice a similar relationship in the accuracy and hallucination as document size increases. From Table 7, we observe that the token length of the TriviaQA documents is much larger than MS-MARCO and SQuAD. Similarlt, we observed lower accuracy and higher hallucination rates with TriviaQA when compared to MS-MARCO and SQuAD. Table 10 and 11 show performance of Chain of Thought Prompting and Chain of Verification performance on open-source datasets. Accuracy and hallucination rate improvement vary based on the open source dataset.

B Prompt Examples

We utilize LLMs for various tasks in our methodology, including question-answer pair generation from knowledge base articles, response generation, factual accuracy evaluation, and advanced CoTP & CoV prompts. Therefore, we include the specific prompts used for these different tasks.

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.

document: <retrieved_document>,
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 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: <document>,
question: <question>,
output:

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

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

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

dataset	accuracy		hallucination_rate		missing_rate	
	baseline	CoTP	baseline	CoTP	baseline	CoTP
SQUAD	98.14	94.58	1	4.16	0.86	1.9
TRIVIA	63.95	86.27	22.85	6.45	13.18	6.38
MS-MARCO	92.1	90.94	4.64	4.32	3.26	4.74

Table 10: 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	98.14	95.96	1	3.24	0.86	0.8
TRIVIA	63.95	63.76	22.85	23.83	13.18	12.4
MS-MARCO	92.1	92.6	4.64	5.54	3.26	1.86

Table 11: Baseline vs CoVe evaluation metrics (%) without retrieval (question-doc. pair used as it is)

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 (open source datasets). 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: