QADLM: Combines QA Paris and Doc-Enhanced QA System with Human Preferences

Xuewen Zhang, Juyi Qiao and Junming Jiao

December 23, 2024

QADLM: Combines QA Paris and Doc-Enhanced QA System with Human Preferences

1st Xuewen Zhang
Li Auto Inc.
Beijing, China
zhangxuewen@lixiang.com

2nd Juyi Qiao
Li Auto Inc.
Beijing, China
qiaojuyi@lixiang.com

3rd Junming Jiao
Li Auto Inc.
Beijing, China
jiaojunming@lixiang.com

***Abstract*—Recent advancements in LLMs like GPT-4 and PaLM have significantly improved QA system, yet their application in customer service poses challenges such as slow response times and hallucinations. Traditional NLP methods, while more cost-effective, struggle with sustainability and maintaining knowledge bases. This paper introduces QADLM, a two-stage QA system that integrates LLMs with traditional NLP techniques to overcome these limitations. In the first stage, a funnel-shaped matching model leverages a domain-specific FAQ corpus to enhance user intent recognition. In the second stage, a fine-tuned RAG model retrieves relevant knowledge documents and generates high-quality responses. Extensive experiments conducted on a new energy vehicle company's dataset show that the proposed system outperforms conventional approaches in response speed and quality. The optimized model's hallucination rate decreased by 29.7%, and semantic similarity improved by 19.5%. This demonstrates the system's robustness and applicability in customer service scenarios.**

***Index Terms*—two-stage question answering, large language model, retrieval-augmented generation**

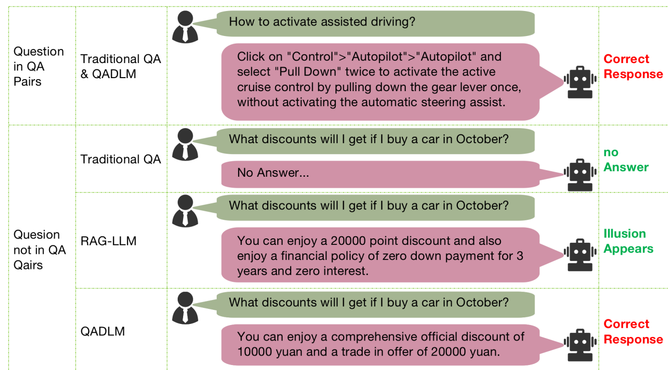

Fig. 1. **Example Model Answer**. In **questions in QA pairs**, all models can answer correctly. In **quesion not in QA pairs**, **Traditional QA** cannot answer, Although **RAG-LLM** answered, it experienced hallucinations, our **QADLM** answered correctly.

I. Introduction

In recent years, Large language models (LLMs) like GPT-4[1], OPT[2], PaLM[3], BLOOM[4], and GLM-130B[5] have greatly expanded the capabilities of machines in language understanding and generation. Recent advancements in LLMs have also significantly improved question answering[6, 7, 8], one of the most essential applications of language technology. However, QA system in customer service scenarios still face several challenges. If LLMs are used independently to build customer service QA systems, issues such as slow response times and high hallucination rates may arise. Conversely, continuing to employ traditional natural language processing (NLP) techniques results in prohibitively high costs for maintaining the knowledge base, making sustainable development difficult.

To address these challenges, this paper introduces an innovative two-stage customer service QA system that integrates traditional NLP techniques with LLMs. In the first stage, a multi-level funnel-shaped matching model is constructed, utilizing a domain-specific FAQ corpus to enhance the accuracy of user intent understanding. The second stage incorporates a retrieval-augmented generation (RAG) LLM framework, which retrieves relevant knowledge documents when preliminary matching is insufficient for determining an answer, employing a fine-tuned RAG-LLM model to generate responses. This approach resolves the limitations of existing methods by enhancing the system's ability to handle complex queries and effectively integrate external knowledge sources.

To validate the effectiveness of our proposed method, extensive experiments were conducted using experimental data from a new energy vehicle company. Results indicate that our system outperforms traditional methods in terms of response speed and answer quality. The contributions of this paper are as follows: 1. An innovative two-stage customer service QA model is proposed, consisting of a FAQ-based query matching model and a knowledge document-based fine-tuned RAG-LLM model. This integrated approach enhances the system's ability to handle complex and non-standardized questions and significantly improves user experience. 2. Experimental validation shows that this model outperforms traditional methods in terms of response speed and answer quality. Notably, the fine-tuned RAG-LLM model demonstrates better results in both hallucination rate and semantic similarity compared to previous metrics.

II. Related Work

The development of document-enhanced QA system is a comprehensive endeavor that demands interdisciplinary collaboration, integrating large language models, domain-specific document question answering, retrieval augmentation, and reinforcement learning through human feedback. In this section, we provide a concise overview of the relevant literature in these areas.

LLMs. particularly self-supervised[9] ones, have garnered significant attention in contemporary NLP. Their vast number of parameters enables them to capture and retain diverse knowledge, resulting in exceptional performance across various challenges. Notable examples of LLMs include GPT-3[1], OPT[2], PALM[3], BLOOM[4], and GLM-130B[5]. A remarkable feature of LLMs is their prompt-based in-context learning (ICL), which facilitates task transfer without the need for tuning, using demonstration samples. Recent research has focused on optimizing[10, 11, 12, 13] ICL and analyzing [14, 15, 16, 17]. The QADLM system leverages the strengths of LLMs, to enhance the customer service QA system. By incorporating LLMs into the second stage of the QADLM framework, the system can retrieve relevant knowledge documents and generate responses that are not only accurate but also contextually relevant. Moreover, the fine-tuning of the RAG-LLM model within QADLM addresses the challenge of high computational costs and slow response times associated with deploying LLMs in practical applications.

Multi-document Question Answering (Doc QA). Many document retrieval methods ensure quality through two components: retriever and reader[18]. The retriever aims to select the needed documents from numerous sources. Recent studies often utilize dense retrievers[19, 20] or commercial search engines [21] to accomplish this. The reader's purpose is to identify suitable text segments, using sequence models to generate content[22, 23, 24], which is effective for data reasoning models[19, 20] or aggregating information from multiple documents[17, 21]. Some researchers also enrich these components by applying query decomposition[24, 25, 26] or search engine retrieval[24]. In the context of this work, QADLM leverages the strengths of LLMs, especially their self-supervised learning capabilities, to enhance the customer service QA system. By integrating these models into our two-stage system, we aim to capitalize on their knowledge retention and generation capabilities. The first stage of our system utilizes a multi-level funnel-shaped matching model to accurately understand user intent, while the second stage employs a retrieval-augmented generation framework to retrieve and generate responses from relevant knowledge documents.

Retrieval-augmentation. The mainstream information retrieval methods are divided into sparse vector based methods and dense vector based methods, such as DPR [27], Competitor [28], REALM [29]. Among these retrieval methods, techniques such as RAG [30], fusion in decoders [31], and Atlas [32] were used. QADLM also includes these methods, and our model interacts with multiple documents to improve overall accuracy. In order to improve retrieval efficiency, QADLM will use many small retrievers to complete it through hierarchy.

Reinforcement Learning from Human Feedback (RLHF). Rating the generated text, mature methods include BLEU [33], ROUGE[34], METEOR[35], and BERTScore[36]. Recently, some researchers believe that learning human preferences from human feedback[37,38] can bring good results. Moreover, QADLM's approach to RLHF is not limited to post-generation evaluation but is integrated into the model's training pipeline. This enables the model to anticipate and incorporate human preferences during the generation process itself, leading to more natural and human-like responses. By doing so, QADLM not only meets the current standards of quality in text generation but also sets a new benchmark for incorporating human feedback into the development of intelligent QA system.

III. Our Model: QADLM

QADLM combines traditional NLP techniques and LLM to construct a framework for implementing a two-stage customer service question answering system. The main work is as follows: In the first stage, a multi-level, funnel-shaped matching model was designed using traditional natural language processing techniques and domain FAQ as the base corpus. This method improves the accuracy of understanding the intention of feedback content through multi-stage intention comprehension. Based on the constructed corpus, the funnel method is used to understand user feedback intentions layer by layer and provide corresponding standard answers. If the user's feedback intention is still incomprehensible, an optimization loop will be formed by improving the feedback corpus through intention. In the second stage, based on the organized knowledge document, this article designed the RAG-LLM framework based on the knowledge document. When the initial match is insufficient to determine the answer, the system will further retrieve relevant knowledge documents and generate answers through the RAG-LLM model. The goal of this stage is to enhance the answering ability of the question answering system using document knowledge, especially when dealing with complex or unclearly queries included in the FAQ. The process is shown in Figure 2. This study provides a new methodological framework for the vertical application of intelligent question answering systems.

A. QA paris Question Answering model

In this section, we employ both sparse feature matching and dense vector matching techniques.

Sparse feature matching utilizes the BM25 model to perform retrieval based on text similarity calculations, implemented using jieba for word segmentation and the Gensim library.

In contrast, **dense vector matching** leverages a pre-trained language model to encode user-input questions and standard questions from the database, training a twin-tower structure to capture similarities in the feature space. The pre-trained language model is employed as an encoder to model the user's input question along with the standard and similar questions in the database. A dual-tower structure is used to train the model, ensuring that relevant questions are closer to each other in the feature space. The question representation model converts the natural language questions and the standard/similar questions in the database into vectorized feature representations. An

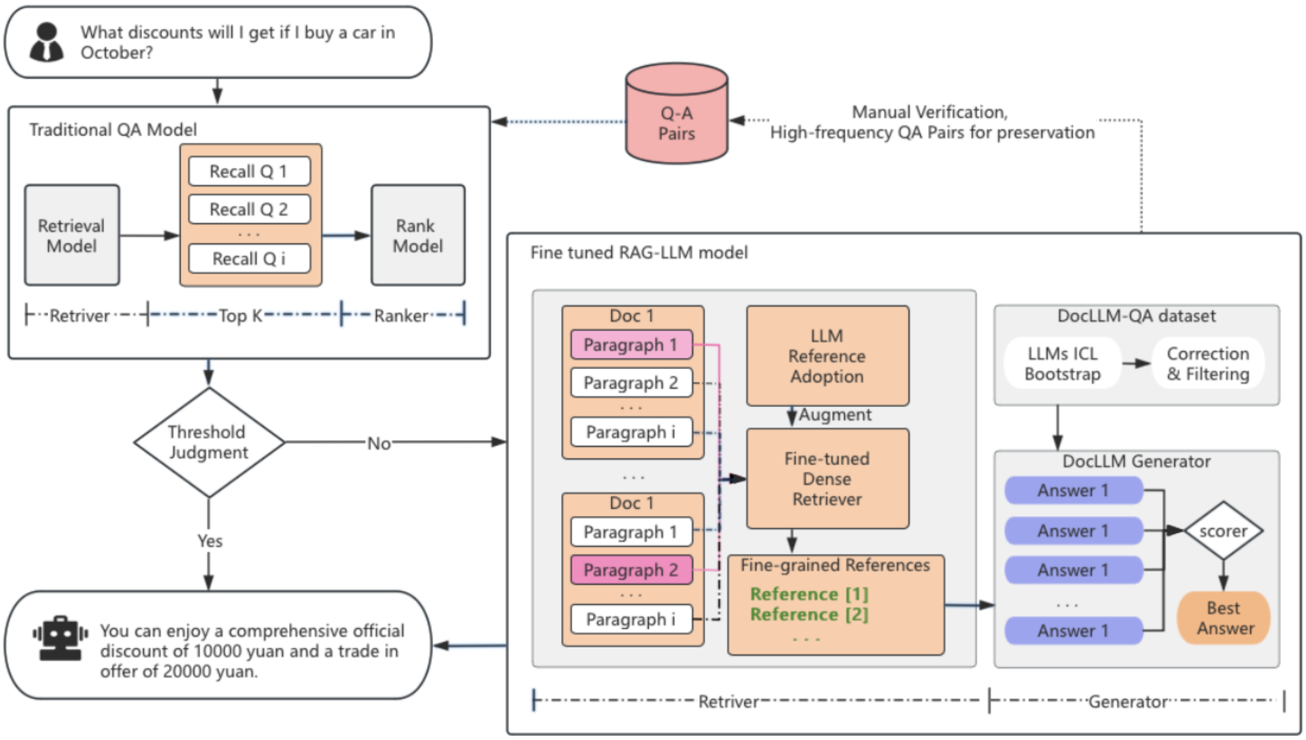

Fig. 2. QADLM QA Flow.

average pooling layer is applied to the output feature sequences to compress them into one-dimensional vectors. The input sequence from the user and the standard/similar questions stored in the database are represented by Equation (1) and (2), where q represents the characters corresponding to the input question text, and t represents the characters of the standard and similar questions in the database.

$$\text{Input} = [[\mathrm{CLS}], \mathrm{q}_1, \mathrm{q}_2, \ldots, [\mathrm{SEP}]] \tag{1}$$

$$\text{Input} = [[\mathrm{CLS}], \mathrm{t}_1, \mathrm{t}_2, \ldots, [\mathrm{SEP}]] \tag{2}$$

A Siamese neural network is used between the user's input question and the stored standard question, and parameter sharing is implemented between them. During the model's training process, when a user input question-standard question or a similar question test sample is input, the user input question and the standard/similar question from the database are separately processed by the user input question representation model and the database's standard/similar question representation model for feature extraction. The feature extraction process is represented by formulas (3) and (4).

$$\mathrm{h}_{\mathrm{q}} = \mathrm{MeanPooling}\left(\mathrm{ENC}_{\mathrm{q}}(\mathrm{q})\right) \tag{3}$$

$$\mathrm{h}_{\mathrm{t}} = \mathrm{MeanPooling}\left(\mathrm{ENC}_{\mathrm{t}}(\mathrm{T})\right) \tag{4}$$

Here,$ENC_q(q)$ and $ENC_t(T)$ respectively represent the pre-trained models used for feature extraction of the user's input question and the standard/similar questions stored in the database. The similarity between the two is calculated by the formula shown in (5).

$$S(\mathrm{q},\ \mathrm{T}) = \frac{\mathrm{h}_{\mathrm{q}}^{\mathrm{T}}\ \mathrm{h}_{\mathrm{T}}}{\|\ \mathrm{h}_{\mathrm{q}}\|\, \mathrm{h}_{\mathrm{T}}\|} \tag{5}$$

The loss function is expressed in formula (6).

$$\text{loss} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp\left[\ \mathrm{S}\left(\mathrm{q}_{\mathrm{i}},\ \mathrm{T}_{\mathrm{i}}\right)\right]}{\sum_{\mathrm{j}=1}^{\mathrm{B}} \exp\left[\ \mathrm{s}\left(\mathrm{q}_{\mathrm{i}},\ \mathrm{T}_{\mathrm{j}}\right)\right]} \tag{6}$$

B. Doc-Enhanced Question Answering model

The construction process of the Doc-Enhanced Question Answering model is outlined as follows:

(1) Baseline Model Selection: When initially selecting a set of pretrained large models, three key aspects were considered. First, the SuperCLUE ranking of various capabilities of Chinese general large models was taken into account, reflecting model performance across multiple tasks, including text classification and named entity recognition. Second, the star ratings of related applications on the GitHub platform were examined, as this metric partially indicates the popularity and influence of models in practical applications. Finally, the open-source and commercial viability of the models was evaluated, as this is crucial for future use and customization, providing greater flexibility and scalability. Based on this comprehensive assessment, Baichuan2-13B-Chat (referred to as Baichuan), ChatGLM2-6B (referred to as ChatGLM), and Llama-2-13B-Chat (referred to as Llama) were selected as the initial set of foundational pretrained models. These models performed exceptionally well in the SuperCLUE rankings, demonstrating superior performance across various tasks. Additionally, they received a considerable number of stars on GitHub, indicating community recognition and support. Importantly, all selected models are open-source and commercially available, offering users greater flexibility for secondary development and customization based on practical needs. A comparative analysis with ChatGPT was conducted to comprehensively assess the performance of these models, providing critical insights into their advantages and limitations across different tasks and scenarios, thus informing future applications and improvements.

(2) Model Fine-Tuning Selection: This study employs the Lora method for fine-tuning large pretrained models to meet the requirements of question-answering tasks.

(3) Model Optimization Phase: With the widespread application of large models, issues related to hallucination have emerged, where generated texts may deviate from or inaccurately represent the original content. This research categorizes such issues into three types: information conflict, fabrication, and information mismatch. To address these problems, a dual approach to fine-tuning optimization is proposed, focusing on both data and model aspects. Data optimization involves deduplication of annotated corpora and manual removal of data that may induce hallucinations. Model optimization employs RAG techniques, which combine large models with external knowledge sources. By constructing external knowledge bases, knowledge vector repositories, vector retrieval, and answer generation, this method effectively alleviates hallucination issues, enhances the quality and validity of generated texts, and addresses data security concerns.

C. Model Ensemble

The model relies on two sub-models to process queries: the FAQ-based Query Matching model and the RAG-LLM model based on knowledge documents. The FAQ-based Query Matching model aims to quickly provide precise answers by matching user queries with entries in a predefined FAQ database. This approach is highly efficient when addressing common or standardized questions. However, not all user queries can be satisfactorily answered by the FAQ model. In such cases, the system invokes the RAG-LLM model, which retrieves relevant fragments from knowledge documents and, by integrating language generation techniques, constructs personalized responses tailored to the user's query. This method not only enhances the ability to handle complex and non-standardized questions but also significantly improves the user experience by offering more in-depth and detailed information.

IV. Experiments

A. Datasets

The experimental data presented in this study is sourced from a certain new energy vehicle company, primarily encompassing the company's customer service-related textual corpus and associated documents. The textual corpus includes a structured collection of 3,591 FAQ entries stored in a question-and-answer format. To enhance recognition accuracy, each standard question is accompanied by several similar variants, as illustrated in Table 1.

TABLE I
FORMAT OF COMMON FAQ DATA

Standard Question	Similar Questions	Answer
What charging methods are available for vehicles?	What are the vehicle charging methods? How is the vehicle charged? How do vehicles get charged? How should I choose the charging method for my vehicle?	Charging methods include 380V fast charging and 220V slow charging. Fast charging utilizes a 12V auxiliary power supply for guidance, allowing for a charge from 20% to 80% in as little as 30 minutes, with a maximum charging power of 60KW. Slow charging supports portable charging guns and national standard slow charging piles, with the capacity to fully charge from 0% to 100% in approximately 6 hours, and a maximum charging power of 7KW.

The question matching dataset primarily derives from historical customer service dialogues within the automotive company. Through manual annotation, 10,000 pairs of matching questions and 20,000 pairs of non-matching questions were randomly selected. In the matching question pairs, a clear semantic correlation exists between the two questions, typically involving similar inquiries or the same subject matter. Conversely, non-matching question pairs often refer to different questions or topics, lacking apparent semantic connections. The annotation method designates matching question pairs with a label of 1, while non-matching pairs are labeled with 0. The detailed format is presented in Table 2.

TABLE II
MATCHING DATASET FORMAT

User Input Question	Standard Question / Similar Question	Matching Label
What is the maximum climbing gradient?	What is the maximum climbing gradient of the drive system?	1
What are the differences between the driving modes?	What are the distinctions among the three driving modes of the vehicle?	1
How far is the red line in the reversing camera reminder?	Can the reversing camera be turned off?	0
Do l need to press the accelerator and brake for automatic parking?	Do l need to hold the steering wheel for the automatic parking system?	0

The document data comprises training materials for customer service staff, announcements published on the company's official website, and internal shared documents, totaling 2,452 documents. This data is utilized for the RAG-LLM model to generate user responses. For better optimization of the model in later stages, the organized documents are categorized into six distinct types based on actual business needs. Some documents encompass multiple business categories and are classified under comprehensive business documentation. The specific number of documents for each business scenario is detailed in Table 3.

TABLE III
STATISTICS OF DOCUMENT COUNTS ACROSS DIFFERENT BUSINESS SCENARIOS

Category	Quantity
Vehicle Pre-sales	34
Vehicle After-sales	79
Charging Related	87
Roadside Assistance	34
Financial Related	95
E-commerce Related	51
Comprehensive Documents	72

B. Maintaining the Integrity of the Specifications

Based on the QAparis experiments, this study aims to compare the model performance of sparse feature versus dense vector retrieval matching during training. Throughout the training process, parameters are shared among the models, utilizing the Chinese-bert-wwm pre-trained language model for feature extraction. The training consists of a total of 10 epochs and employs random sampling techniques. The training parameters include a learning rate of 2e-5, a hidden layer dimension of 768, a batch size of 32, the AdamW optimizer, a maximum input length of 128, and 12 encoder layers. Model performance is evaluated using recall rates.

The implementation of the fine-tuning framework for LLMs provided by ModelScope facilitates a streamlined approach for both fine-tuning and inference of our model. All experiments were conducted using NVIDIA A100 80GB and A100 32GB GPUs. The fine-tuning process employed a Low-Rank Adaptation (LoRA) strategy, with specific configurations including a LoRA rank set 4, a scaling factor for the learning rate (LoRA alpha) established at 8, and a dropout rate for overfitting management (LoRA dropout) fixed at 0.05. The LoRA target modules were designated to encompass all relevant modules. The maximum length of input sequences was constrained to 3072 tokens. For training, the AdamW optimizer was utilized, with a learning rate of 1e-6, and a batch size of 1 per GPU. The model was trained for four epochs using DeepSpeed's ZeRO-23 optimization, with checkpoint 1700 identified a s the optimal model. During the inference phase, greedy decoding was implemented by setting the do_sample parameter to false, ensuring stability in output generation. The repetition penalty was calibrated between 1.00 and 1.02, while the maximum number of new tokens generated was limited to 512. The vLLM framework was employed to enhance the efficiency of the inference process, which required approximately 40 minutes to produce the final results on a A100 32GB GPU.

C. Competition Results

Based on the previously established experimental setup, the results of the twin-tower model experiments are presented, with the final outcomes summarized in Table 4. In the table, Dr (Dense Retriever) refers to the dense vector retrieval model. The results indicate that the retrieval models corresponding to dense vectors exhibit commendable performance. On the test set, the models utilizing the optimized sampling strategy outperformed BM25 across all four metrics, with the most notable improvement observed in Recall@3.

TABLE IV
PERFORMANCE COMPARISON OF MODELS BM25 AND DR ON DEV AND TEST SETS ACROSS DIFFERENT RECALL METRICS.

Models	Recall@1	Recall@3	Recall@5	Recall@10
Dev				
BM25	73.58%	83.02%	86.01%	88.88%
Dr	85.43%	89.01%	93.45%	96.84%
Test				
BM25	74.21%	84.33%	86.52%	89.32%
Dr	88.76%	90.23%	95.03%	97.05%

According to the previous experiments, the evaluation of the RAG-LLM model results was conducted using two key metrics: hallucination rate and semantic similarity. These metrics are crucial for assessing the accuracy and reliability of the experimental outcomes. The hallucination rate was determined by voting from seven experts, while semantic similarity was computed using the TF-IDF algorithm.

In the case of the hallucination rate, the value was established through votes cast by the seven professionals, whose expertise and experience provide significant reference for evaluating the hallucination rate, ensuring objectivity and accuracy in the assessment results. Analyzing the hallucination rate can help researchers identify issues and biases present in the experimental outcomes, allowing for necessary adjustments and improvements to enhance the reliability and effectiveness of the experiments.

As another key metric, semantic similarity plays an essential role in evaluating the semantic accuracy of the experimental results. Semantic similarity is quantified using the TF-IDF algorithm, which measures the degree of semantic similarity between texts. In the evaluation of experimental results, the level of semantic similarity reflects the proximity between the experimental outcomes and actual situations. A higher semantic similarity indicates greater consistency and accuracy between the experimental results and real-world conditions, while a lower semantic similarity may suggest potential semantic biases or errors within the experimental results.

(1) Hallucination Rate Analysis. This section compares the degree of hallucination phenomena exhibited by large models before and after optimization across different scenarios, as illustrated in Figure 3. The analysis of optimization results reveals that hallucination issues are alleviated in all scenarios post-optimization. For instance, in the vehicle after-sales scenario, hallucination rates for Llama decreased by 25.8%, while ChatGLM showed a reduction of 29.7%, with ChatGLM exhibiting the most significant improvement.

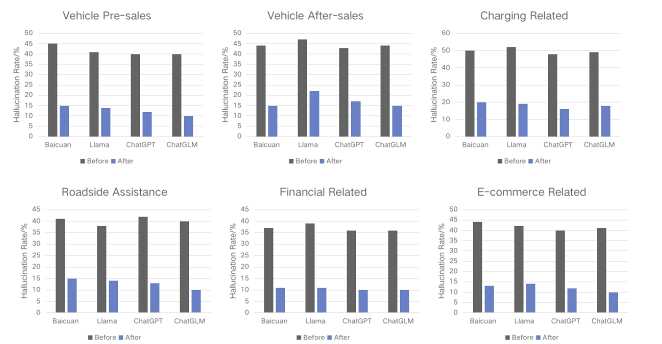

Fig. 3. Comparison Analysis of Hallucination Rates Before and After "Fine-tuning + Optimization" for Different Intents in LLMs.

(2)Semantic Similarity Analysis. According to the data presented in Figure 4, all models exhibited improvements in performance following optimization, with increases ranging from 15.6% to 25%. ChatGLM demonstrated the best performance in this process, followed by Baichuan, while ChatGPT and Llama showed comparatively lesser enhancements.

Based on the above experiments and analyses, ChatGLM was ultimately selected for constructing a large language pre-trained model for an intelligent customer service question answer system in the enterprise after-sales domain.

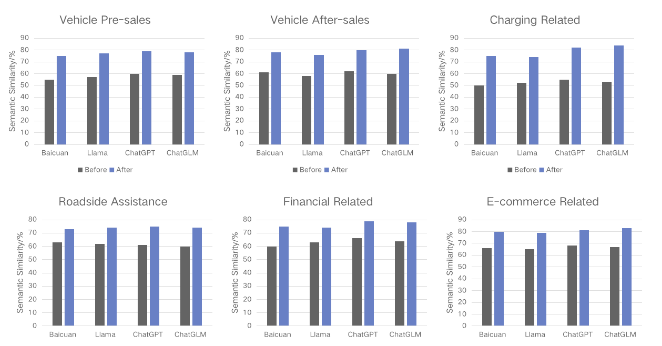

Fig. 4. Comparison Analysis of Semantic Similarity Before and After "Fine-tuning + Optimization" for Different Intents in LLMs.

D. Ablation Studies

Ablation Study on Noise File Filters. Table 5 presents the experimental results of a single model inferred with and without noise document filters. Our findings indicate that crucial information still exists within certain noise documents. Consequently, although the model's accuracy improves with the inclusion of noise documents, the hallucination rate correspondingly deteriorates. Given that the entire model is intended for use in a customer service system, the accuracy metric is of paramount importance. Therefore, in our data processing for the experiments, we opted to preemptively exclude noise documents.

TABLE V
ABLATION STUDY OF NOISY DOCUMENT EXPERIMENT.

Noisy Document Filter	Hallucination Rate	Semantic Similarity
×	12.75%	77.25%
✓	13.42%	77.79%

Ablation Study on Each Component. We conducted additional experiments to perform an ablation study on each component. We compared the system's two submodules: the QA pair matching module of the twin-tower model and the RAG-LLM module. The results are presented in Table 6.In the QA pair matching module of the twin-tower model, we assessed the performance with and without this module. As indicated in Table 6, the removal of the QA pair matching module resulted in significantly poorer performance in terms of semantic similarity and no-answer rate compared to the other two scenarios.In the RAG-LLM module, we experimented with not fine-tuning the LLM model and instead retrieving answers directly through prompts. The results showed a notable increase in the hallucination rate. Additionally, within the same dataset, there was also an increase in the no-answer rate.

TABLE VI
ABLATION STUDY ON DIFFERENT SUB-MODULES.

Noisy Document Filter	Hallucination Rate	Semantic Similarity	No-Answer
No QA pairs	12.79%	63.21%	40%
No PEFT RAG-LLM	30%	75.23%	17%
All models	12.75%	77.25%	13%

V. CONCLUSION

This paper constructs a framework to implement a two-stage customer service QA system. Firstly, utilizing traditional natural language processing techniques and an FAQ corpus specific to the automotive sector, a multi-level, funnel-shaped matching model is designed. Secondly, based on the organized knowledge documents, a RAG-LLM framework is developed; when preliminary matching is insufficient to determine an answer, the system further retrieves relevant knowledge documents to generate responses using a fine-tuned RAG-LLM model. This research offers a methodological framework for the application of intelligent QA system, significantly enhancing response speed and answer quality.

REFERENCES

[1] Achiam, Josh, et al. "Gpt-4 technical report." arXiv preprint arXiv:2303.08774 (2023).

[2] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al . 2022. Opt: Open pre-trained transformer language models. arXiv preprint arXiv:2205.01068 (2022).

[3] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. arXiv preprint arXiv:2204.02311 (2022).

[4] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. arXiv preprint arXiv:2211.05100 (2022).

[5] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. arXiv preprint arXiv:2210.02414 (2022).

[6] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. Transactions of the Association for Computational Linguistics 7 (2019), 453–466.

[7] 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. 2383–2392.

[8] Kim, Jaewoong, and Moohong Min. "From rag to qa-rag: Integrating generative ai for pharmaceutical regulatory compliance process." arXiv preprint arXiv:2402.01717 (2024).

[9] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. IEEE Transactions on Knowledge and Data Engineering 35, 1 (2021), 857–876.

[10] Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. 2022. What Makes Good In-Context Examples for GPT-3?. In Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures. 100–114.

[11] Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Noisy Channel Language Model Prompting for Few-Shot Text Classification. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 5316–5330.

[12] Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al . 2022. Selective annotation makes language models better few-shot learners. arXiv preprint arXiv:2209.01975 (2022).

[13] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In International Conference on Machine Learning. PMLR, 12697–12706.

[14] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? arXiv preprint arXiv:2202.12837 (2022).

[15] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning To Retrieve Prompts for In-Context Learning. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2655–2671.

[16] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An Explanation of In-context Learning as Implicit Bayesian Inference. In International Conference on Learning Representations.

[17] Liu, Weihao, et al. "MMHQA-ICL: Multimodal In-context Learning for Hybrid Question Answering over Text, Tables and Images." arXiv preprint arXiv:2309.04790 (2023).

[18] Zhu, Fengbin, et al. "Retrieving and reading: A comprehensive survey on open-domain question answering." arXiv preprint arXiv:2101.00774 (2021).

[19] Trivedi, Harsh, et al. "Teaching Broad Reasoning Skills via Decomposition-Guided Contexts." Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. 2022.

[20] Ferguson, James, et al. "IIRC: A dataset of incomplete information reading comprehension questions." arXiv preprint arXiv:2011.07127 (2020).

[21] Lazaridou, Angeliki, et al. "Internet-augmented language models through few-shot prompting for open-domain question answering." arXiv preprint arXiv:2203.05115 (2022).

[22] Lewis, Patrick, et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks." Advances in Neural Information Processing Systems 33 (2020): 9459-9474.

[23] Xiong, Wenhan, et al. "Answering complex open-domain questions with multi-hop dense retrieval." arXiv preprint arXiv:2009.12756 (2020).

[24] Press, Ofir, et al. "Measuring and narrowing the compositionality gap in language models." arXiv preprint arXiv:2210.03350 (2022).

[25] Adolphs, Leonard, et al. "Boosting search engines with interactive agents." arXiv preprint arXiv:2109.00527 (2021).

[26] Huebscher, Michelle Chen, et al. "Zero-shot retrieval with search agents and hybrid environments." arXiv preprint arXiv:2209.15469 (2022).

[27] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for OpenDomain Question Answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 6769–6781.

[28] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised Dense Information Retrieval with Contrastive Learning. Transactions on Machine Learning Research (2022).

[29] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In International conference on machine learning. PMLR, 3929–3938.

[30] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems 33 (2020), 9459–9474.

[31] Gautier Izacard and Édouard Grave. 2021. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. 874–880.

[32] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. arXiv preprint arXiv:2208.03299 (2022).

[33] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 311–318.

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

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

[36] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating Text Generation with BERT. In International Conference on Learning Representations.

[37] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems 35 (2022), 27730–27744.

[38] 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.