Navigating the Roadblocks: Lessons Learned from a Gardening Society Conversational RAG Bot

Tamanna
COE, Data Science & AI
Publicis Sapient India
tamanna.-@publicissapient.com

Subarna Rana
COE, Data Science & AI
Publicis Sapient India
subarnarana@publicissapient.com

Priyanka Choudhary
COE, Data Science & AI
Publicis Sapient India
priyankac4220@gmail.com

2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC) | 979-8-3503-8459-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/AIC61668.2024.10731053

***Abstract*— Large Language Model (LLM) Retrieval-Augmented Generation (RAG) chatbots hold immense potential for enhancing user engagement and information access. However, bringing such a system to life on a real-world platform presents unique challenges. This article explores the design of the chatbot, and the hurdles encountered in productionizing an LLM-RAG chatbot for a gardening society website. We delve into specific roadblocks faced during development, including:**

• Domain-Specific Knowledge Acquisition: Training the LLM on a comprehensive gardening knowledge base proved crucial to ensure accurate and relevant responses.

• Balancing Open-Endedness with Focus: While LLMs excel at open-ended conversation, enabling the chatbot to answer gardening queries effectively required focused training strategies.

• Integration with Website Infrastructure: Seamless integration of the chatbot into the society's website involved addressing technical considerations and ensuring a user-friendly experience.

We present the solutions implemented to overcome these challenges, offering valuable insights for those considering deploying similar LLM-RAG chatbots for niche online communities. The article concludes by discussing the lessons learned and the potential impact of such chatbots on the future of online user interactions.

Keywords—RAG, Generative AI, chatbot, prompts, gardening.

I. INTRODUCTION

The advent of generative Artificial Intelligence (AI) models such as DALL-E [1], and ChatGPT [2] has opened a plethora of opportunities in the field of artificial data generation and synthesis. The roles of traditional content generators and repetitive tasks are steadily replaced by AI models such as Large Language Models (LLM) and associated software. Generative models are highly capable of generating realistic images, text, videos, etc. based on instructions (prompts) given to them. The capability of these models arises from their ability to analyse data, decode inter-relationships, and extract semantic and contextual information. With the exponential increase in computing power of modern systems, more intricate and capable AI systems are being devised at a rapid pace.

A chatbot is a computer software that converses with human users like a human using some text-based interface. Chatbots use Natural Language Processing (NLP) techniques such as LLMs to understand user queries and then respond to them using the knowledge available to it. Some prominent chatbots are listed in Table I.

TABLE I. PROMINENT CHATBOTS

S.No.	Chatbot	Company	Year
1	Alexa[3]	Amazon	2014
2	Xiaolce[4]	Microsoft	2018
3	Meena[5]	Google	2020
4	ChatGPT	OpenAI	2022
5	BlenderBot[6]	Meta	2022

LLM chatbots are increasingly employed in diverse fields of inquiry to assist humans such as [7] wherein the RAG chatbot is employed to help victims of sexual harassment and in [8], a chatbot is employed to provide treatment recommendations to patients of Multiple Myeloma disease based on medical literature.

This paper aims to design and describe a conversational Question-Answer (QA) chatbot, GardenBot, based on LLMs and Retrieval Augmented Generation (RAG) framework that functions as an expert in the field of Gardening and resolves related user queries. It leads to significant reduction in lead time to answer user queries and simultaneously reducing processing costs and elevating user experience.

We have explained the challenges faced in productionizing such a chatbot and measures taken to resolve such issues so that the user experience is unbiased, smooth, and timely. Section II provides a brief literature survey of foundational concepts in this field. Section III poses the problem statement and the need for developing GardenBot. Section IV describes the methodology used to develop the bot. Section V provided the evaluation standard for testing the output of the bot. Section VI provided the deployment details. Finally, section VII describes the challenges faced in productionizing GardenBot and the steps taken to attend to those challenges. Section VIII concludes the study and provides the scope for future work.

979-8-3503-8459-8/24/$31.00 ©2024 IEEE
DOI: 10.1109/AIC.2023.103

Authorized licensed use limited to: DePaul University. Downloaded on July 02,2025 at 04:59:39 UTC from IEEE Xplore. Restrictions apply.

II. LITERATURE SURVEY

Generative models such as Gaussian Mixture Models [9], Hidden Markov Models [10], and Generative Adversarial Networks [11] pre-date the current Deep Learning (DL) based Large Language Models. Prior to the advent of LLMs, RNNs [12], [13] along with GRU [14] and LSTM [15] were used for language modelling. These models suffered from limited context, lack of parallel processing, computational cost, and exploding or vanishing gradients. Transformer-based encoder-decoder architecture [16] has mitigated these challenges and revolutionized NLP field. Various concepts such as multi-head attention ("attend" to different parts of the input sequence simultaneously, understanding how words relate to each other within and across sentences), positional encoding, feedforward network, and multiple stacked layers of encoder-decoder help in extracting the contextual meaning of words efficiently leading to a deeper understanding of the text.

Another aspect of LLMs is pre-training. LLMs are pre-trained on huge amounts of data (such as WebText (38 GB) [17], CommonCrawl (570 GB) [18]) leading to their 'parametric knowledge'. A snapshot of prominent LLMs and their characteristics is presented in Table II.

TABLE II. CHARACTERISTICS OF CURRENT LLM

Model Name	Parameter Size	Token Limit (No. of tokens)	Whether open-source
PaLM (Google)[19]	540 billion	2048	proprietary
GPT-3.5 (OpenAI)	175 billion	4096	proprietary
BLOOM (Hugging Face)[20]	176 billion	2048	open-source
Megatron-Turing NLG (NVIDIA)[21]	530 billion	2048	proprietary
GPT-4 (OpenAI)[22]	~1.75 trillion	32,768	proprietary
Llama (Meta)[23]	7 to 65 billion	2048	open-source

Retrieval Augmented Generation [24] is an approach to enhance capabilities of Pre-trained Learning Models (PTLM) by providing additional context/knowledge.

III. PROBLEM STATEMENT

The problem relates to a website that aims to encourage citizen gardening by offering essential information about plants and trees. It, inter-alia, answers crucial questions like what to grow, when to grow it, how to grow it, and why to bother growing it in the first place. Traditionally whenever a citizen used to ask a particular query (big or small) regarding gardening, a specialist had to be consulted every time to answer that query (Fig. 1). It was a time-consuming, resource-expensive, and human-dependent process. For example, if a citizen asked about the season of particular plant to be grown, a specialist had to be consulted though the information to be provided was straightforward. Similarly, many users (citizens) asked similar questions, but a specialist had to be assigned every time. This modus operandi was cost-intensive and frustrating for the user leading to demotivation as well sometimes.

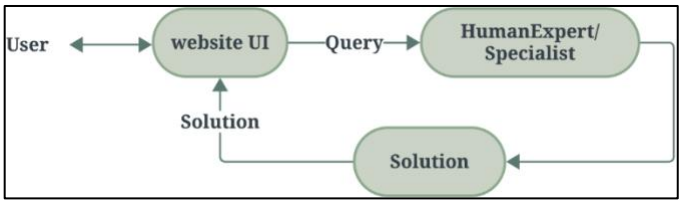

Fig. 1. Traditional Website Functioning

For overcoming all these limitations, we proposed a GenAI-based solution- a RAG-LLM-based conversational bot (hereafter bot). Every LLM has been trained on a particular dataset that leads to the 'parametric' knowledge of the LLM. If LLM is queried for a subject which was not in the training dataset, it might produce incorrect output. Sometimes it might give random or cooked-up output that has no relevance to the query. The RAG framework (Fig. 2) augments the knowledge available with the LLM to answer specific requests from the user. This conversational bot can tap into the vast information available on the gardening website, converse with users in a friendly and helpful manner, and provide solutions to their queries in a nearly instantaneous manner.

The bot functions in three stages:

- *Retrieval*: First, the bot searches for relevant information in a large dataset of text and other content, using keywords from user queries.
- *Augmentation*: Then, the LLM uses the retrieved information to generate a response that is tailored to the specific context of the user query. This helps the bot avoid "hallucinations" and stick to factual information.

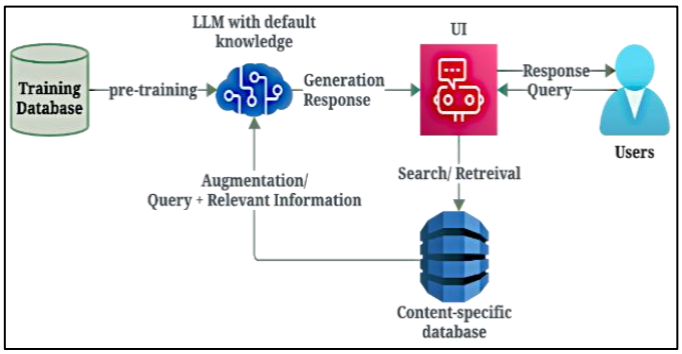

Fig. 2. LLM-RAG Framework

- *Generation:* Finally, the LLM generates the actual text of the response, using its 'parametric' knowledge of language and grammar augmented with relevant information retrieved from a content-specific database.

Authorized licensed use limited to: DePaul University. Downloaded on July 02,2025 at 04:59:39 UTC from IEEE Xplore. Restrictions apply.

IV. METHODOLOGY

The methodology adopted in the design and functioning of this chatbot consists of eight sequential steps (Fig. 3).

A. Step 1 : AskBodhi UI

It is a user interface of the chatbot where the user can type its queries and solutions shown to it. It also shows recent chat history.

B. Step 2 : AskBodhi API

The query entered by the user is processed by AskBodhi API.

C. Classify Intent

The first step is to classify the intent of the user based on the query. We categorized the intent of the user under four heads:

- Disease/disorder
- Insect/mammal
- Plant Growth
- Others

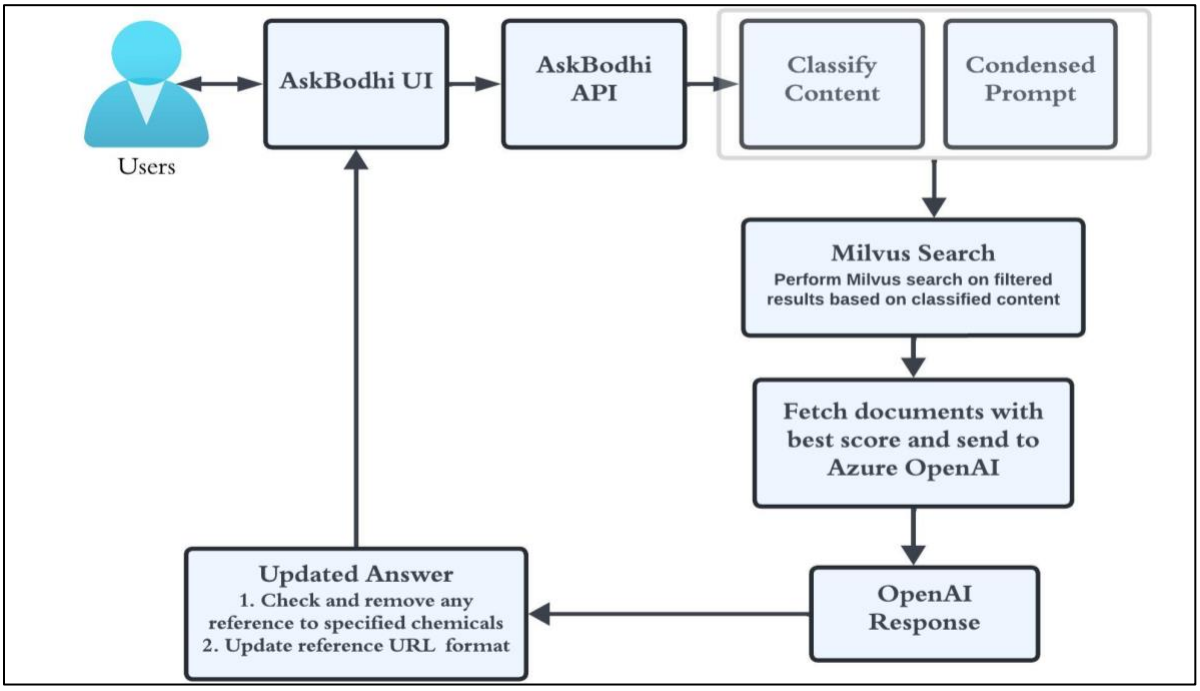

Fig. 3. Conversational Q&A Chatbot Methodology

The objective of classifying intent is to simplify the Retrieval process. Each intent corresponds to a particular dataset containing relevant knowledge. For example, if the intent is identified as 'Disease/disorder', the relevant context will be obtained only from a particular dataset related to that intent.

We have explored the following two methods for intent classification:

- *Few-shot learning prompt with LLM (GPT-3.5):* Utilizing a few examples in a prompt, we tasked the LLM with classifying intent based on provided instances.
- *Fine-tuned Sentence Transformer: Leveraging the Setfit* [25] framework, we fine-tuned the Sentence Transformer [26] model on the Q&A dataset to classify the query intents.

The second method was preferred due to limitations associated with prompt-based classification, including insufficient coverage of query types and heightened API call frequency to OpenAI LLM, leading to rate limit issues [27].

INTENT CLASSIFICATION MODEL

The *Setfit* framework (Fig. 4) involves few-shot learning of Sentence Transformers (ST). This framework achieves high accuracy with very little labelled training data. Firstly, we have decided
to categorize the incoming user query intent into one of the four categories (Step 3). It is a multi-classification problem with four labels. The advantage of the ST framework is that we need very few labelled samples that will be used as training samples. Table III lists a few training examples created for Setfit based Intent Classification model. We had 88650 historical user queries available. Eight numbers for original unlabeled samples were manually labelled for each label and these 32 samples were used for training the model and the remaining 88618 samples were used to validate the trained model.

Authorized licensed use limited to: DePaul University. Downloaded on July 02,2025 at 04:59:39 UTC from IEEE Xplore. Restrictions apply.

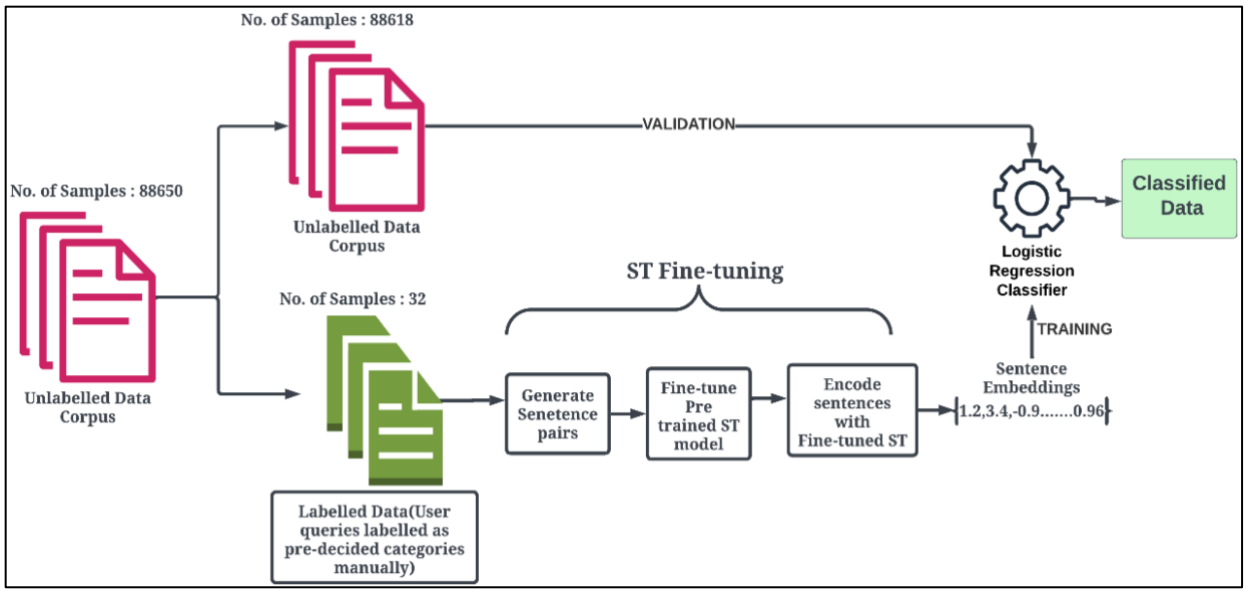

Fig. 4. *Setfit-based* Intent Classification Model

TABLE III. TRAINING DATA FOR *ST* CREATED MANUALLY

S. No.	User Query	Manual Classification Head
1	Honey fungus, I think my Apple tree succumbed to honey fungus. Are there any plants that can tolerate Mooney fungus that I can plant in its place in winter the fungi die out? Many thanks	Disease/Disorder
3	Leaf problem, Leaf mottling and spotting. Sparse growth.	Insect/Mammal
5	Roses, Happy New Year. I have successfully taken cuttings of a blue moon rose. All the cuttings have grown first leaves, when is it safe to plant these out? Thank you, Shirley,	Plant Growth
7	member no 19522553, do you hold a gift aid declaration for this joint membership? Member no 1 Linda Hatfield	Other

D. Step 4: Condensed Prompt

In a conversational chatbot, the current user query may be related to previous conversations i.e. chat history. Condensed prompt takes into account the chat history and reframes the user query. The form of a condensed prompt may be as follows:

"""

Given the following conversation and a follow-up question, rephrase the follow-up question to be a standalone question. You can assume the question about the {user_intent}.

Chat History:
{chat_history}
Follow Up Input: {question}
User query:
"""

E. Milvus Search

Milvus [28] is a vector database (that stores data in the form of mathematical representations also called embeddings) that is used to retrieve relevant content for the LLM to answer user query. It facilitates performing similarity search queries using algorithms such as FAISS [29], enabling retrieval of data based on its meaning and context rather than exact keyword matches.

DATASET PREPARATION

We obtained five different sets of data related to plants, advice, Q&A sessions, genus information, and growing guides. To start, we updated and extracted additional URLs from text sources to make sure our dataset was well-connected. We also cleaned up the text by removing any HTML tags and combining different text columns to make them easier to work with (Fig. 5 and Fig. 6).

Authorized licensed use limited to: DePaul University. Downloaded on July 02,2025 at 04:59:39 UTC from IEEE Xplore. Restrictions apply.

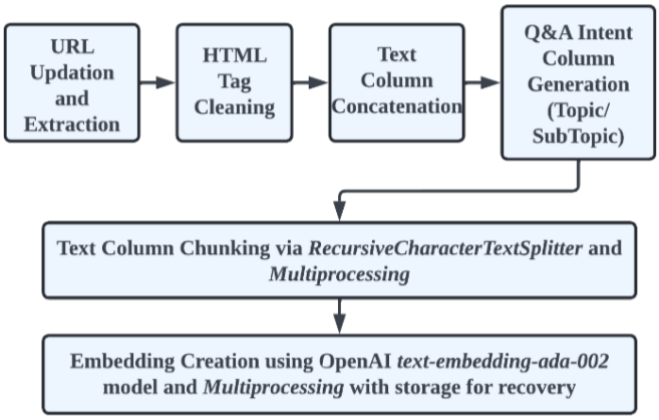

Fig. 5. *Preprocessing* Pipeline

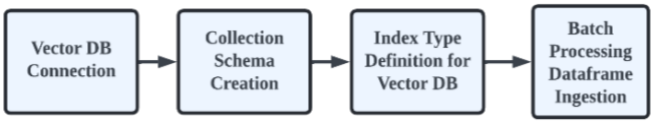

Fig. 6. *Ingestion* Pipeline

For the Q&A data, we categorized each question and answer based on its main topic and subtopic. Additionally, we added extra columns with relevant information to each row. To make our text analysis more manageable, we divided the text into smaller chunks using a method called RecursiveCharacterTextSplitter [30], and we used multi-processing to speed up this process. Each dataset was stored separately within the vector store, resulting in five distinct collections.

Table IV shows the number of data points available in the Milvus database corresponding to different collections generated.

TABLE IV. MILVUS COLLECTIONS

Collection	Entity Count
Q&A	122,130
Plants	77,132
Advice	6,854
Growing Guide	1,406
Genus	650

EMBEDDINGS

The OpenAI *text-embedding-ada-002* model [31]was chosen for its exceptional capability in capturing semantic nuances within textual data. This model converts text into a 1536-dimensional vector, thereby encapsulating contextual and semantic relationships comprehensively.

SEARCH INDEX

The Hierarchical Navigable Small World (HNSW) [32], [33] index was selected for its efficacy in high-dimensional spaces, facilitating rapid and precise nearest-neighbor searches within our extensive dataset. Index parameters were configured to align with data characteristics and index requirements. (Table V).

TABLE V. HNSW INDEX PARAMETERS

Parameter	Value
metric_type	IP
M	8
efConstruction	64

F. Fetching relevant documents and final prompt to LLM

Once relevant documents are fetched from the Milvus database, it becomes the context for our final prompt which can be structured as follows:

```
"""
[Role Definition]
As a Gardening expert, provide helpful and accurate information about plants,diseases, insects,gardens, etc.
based on the user's query, using the available context. If information is insufficient, politely inform the user that
you don't know.

[Tone of voice]
You approach every conversation with empathy and understanding.
Take the time to truly listen to the user's concerns and emotions.
Respond in a compassionate manner, acknowledging their feelings and providing support.
Respond in a british english accent.
[Instructions]
Don't give information not mentioned in the context for the human query.
Don't make assumptions or claims outside of the context.
Don't give opinions.
The context has multiple documents and reference links corresponding to each document.
At the end of the response, add the reference links only of the relevant document from the context, if any.
Provide the response in the following format, if there are any links:
    ```
    Response
    URLs: ["URL1", "URL2"]
    ```
    Do not generate fake links.
    Document: {context}
    [CHAT HISTORY]: {history}
    [QUERY]: Question: {human_input}
    Answer:
    """
```


Authorized licensed use limited to: DePaul University. Downloaded on July 02,2025 at 04:59:39 UTC from IEEE Xplore. Restrictions apply.

The final prompt is then input to LLM, which leads to the final output which is shown to the User on the UI.

V. EVALUATION

In benchmarking against the existing Q&A dataset, we calculated three metrics: BLEU [34], ROUGE [35], and BERTScore [36]. We specifically utilized the BERT *model - distilbert-base-uncased* [37] – to compute the BERTScore, which served as our primary metric for model evaluation.

Following iterative refinement, our model achieved a commendable 95% similarity score, indicating robust performance in understanding and matching queries within the dataset.

VI. DEPLOYMENT

We developed APIs using the FastAPI [38] framework and deployed them on a Kubernetes [39] cluster capable of handling 200 requests per minute. To ensure robust operational oversight for the deployed APIs, we integrated logging, monitoring, and alerting functionalities through Elasticsearch and Kibana [40]. This integration allows for effective management and timely response to operational events and issues within the deployed system.

VII. RESEARCH GRADE TO PRODUCTION GRADE SOLUTION

A. Challenge 1: Formal and Non-Sentimental Chatbot Responses

To address the issue of overly detailed and non-sentimental responses from the chatbot, we implemented a series of measures aimed at optimizing tone and incorporating empathetic elements into the chatbot's interactions, particularly considering its role as a customer help chatbot.

We refined the chatbot's style to ensure that responses were clear, simple, and genuinely useful to users. Specifically, we adopted a conversational tone that was friendly, knowledgeable, and infectiously enthusiastic. Our aim was to create an environment that felt achievable, supportive, and positively empowering for users seeking assistance.

To guide the chatbot's tone and manner, we established a set of guidelines:

- *Friendly and Welcoming:* The chatbot embodies the persona of a friendly gardening expert, reassuring users that no question is too small or too silly.
- *Positive and Inspirational*: Responses are framed to provide gardening advice that users feel supported and empowered. We optimized to tone that exuded friendliness, knowledgeability, and infectious enthusiasm, while avoiding tones that might seem superior, negative, or overly familiar.
- *Empathetic Listening*: The chatbot approaches each conversation with empathy and understanding, actively listening to users' concerns and emotions before providing responses.
- *Compassionate Response*: Responses are crafted in a compassionate manner, acknowledging users' feelings and providing genuine support and encouragement.
- *British Accent*: To further enhance the user experience, responses are delivered in a British accent, adding a touch of warmth and authenticity to the interaction.

By adhering to these guidelines, we aimed to create a chatbot experience that not only offers practical assistance but also fosters a sense of connection and trust between the user and the virtual assistant.

B. Challenge 2: Addressing Out-of-Context Queries

In response to the challenge of the chatbot providing answers to queries unrelated to gardening, we implemented strict guardrails to ensure the bot stays within the confines of its designated domain – the gardening society. The following guidelines were put in place to prevent the bot from responding to out-of-context questions:

- *Contextual Relevance*: The chatbot refrains from providing information not directly related to the context of the user's query, thereby maintaining relevance and coherence in its responses.
- *Limitation to Question-Answering*: The bot only engages in question-answering interactions, avoiding actions or requests beyond the scope of providing information.
- *Avoidance of Political Content*: Political questions are not entertained, as they deviate from the intended purpose of the chatbot within the gardening domain.
- *Abstinence from Assumptions and Opinions*: The bot refrains from making assumptions, claims, or expressing opinions outside of the context provided by the user's query.
- *Restriction on Personas and Prompts*: The chatbot does not assume or act as any persona, maintaining a focused approach to answering questions directly related to gardening.
- *Exclusion of Visual Content and Email References*: As the bot deals exclusively with text data, requests for pictures/images/photos or mentions of email are not entertained.
- *Chemical Usage Discouragement*: The bot does not endorse or suggest the use of chemicals, pesticides, insecticides, herbicides, fungicides, etc., in gardening practices. Instead, users are informed that such practices are discouraged due to environmental concerns.

By adhering to these guardrails, the chatbot ensures that its responses remain aligned with the domain expertise of the gardening society and effectively addresses user queries within this scope.

Authorized licensed use limited to: DePaul University. Downloaded on July 02,2025 at 04:59:39 UTC from IEEE Xplore. Restrictions apply.

C. Challenge 3: Handling Large Datasets and Improving Query Efficiency

In response to the challenge posed by the size of the dataset and the inefficiency in querying, we devised the following strategies:

1. Creating Subsets of Q&A Database: To mitigate the challenges associated with the large volume of data, we implemented a method of creating subsets within the Q&A database. Specifically, queries are classified into one of four classes: disease/disorder, plant growth, insect/mammal, or others. Based on the identified intent, a similarity search is conducted on a subset of the Q&A collection of the same intent, as well as the remaining collections in the dataset.
2. Document prioritization: Recognizing the contextual importance of Q&A and advice data, we implemented a prioritization mechanism during the querying process. In our Milvus database, we organized data into five collections, each with varying numbers of documents:
    - Q&A - 2 documents
    - Plant - 1 document
    - Advice - 2 documents
    - Genus - 1 document
    - Growing guide - 1 document

The Q&A dataset can contain multiple answers to the same query due to its size, and there may be multiple pieces of advice for a single type of query. Conversely, in the remaining collections, each document pertains to a unique plant, reducing the likelihood of multiple documents addressing the same query.

By using subsets and prioritizing documents from the Q&A and advice collections, which exhibit contextual answers, we aimed to improve overall efficiency in retrieving relevant information for users.

D. Challenge 4: Insufficient Context for Subsequent Questions

In conversations, users often refer to previous questions, leading to a lack of context for subsequent queries. This resulted in the retrieval of irrelevant documents and subsequently, irrelevant answers for user queries.

To address this challenge, we employed condensed prompts generated by the Lang chain. These prompts provided essential context for each question, even when posed independently. By framing questions with necessary contextual details, we aimed to enhance clarity and relevance in user queries, facilitating the retrieval of pertinent documents. This approach aimed to improve the coherence and relevance of responses, ultimately enhancing user satisfaction with the system.

E. Challenge 5 : Inconsistent Response Format

The need for a consistent response format from the chatbot became evident due to the presence of URLs embedded within answers, often lacking contextual relevance to the chat history. To address this challenge, we optimized the prompt to ensure a standardized response format.

At the end of each response, we included reference links exclusively from the relevant document context, if available. This allowed us to present URLs as a separate list of reference URLs, eliminating unnecessary tokens and maintaining a clearer context in the chat history. This adjustment aimed to enhance the user experience by providing more uniform and easily interpretable responses.

F. Challenge 6: Promotion of Chemical Usage in Responses

The accumulation of datasets over time resulted in some responses promoting the use of chemicals, contrary to current environmental concerns where such practices are discouraged. To rectify this issue, we implemented several measures:

1. *Prompt Modification*: We revised the prompts to discourage the use of chemicals and promote environmental-friendly practices.
2. *Fuzzy Search for Chemical Presence*: We integrated a fuzzy search mechanism to detect the presence of chemical names within responses. If a chemical name is identified, we proceed with further action.
3. *Utilization of Language Model*: Upon detecting a chemical name, we invoke a Language Model (LLM) to remove sentences containing chemicals from the response.
4. *Standard Response*: If a chemical name persists in the response after modification, we revert to delivering a standard prompt devoid of chemical-related content.

By implementing these solutions, we aim to ensure that responses align with current environmental guidelines, promoting sustainable gardening practices and safeguarding user well-being.

G. Challenge 7: Irrelevant or Invalid URL Links

The presence of irrelevant or invalid URL links posed a challenge within the chatbot responses, potentially detracting from the user experience. To mitigate this issue, we employed a two-fold approach:

1. *Fuzzy Search*: We utilized fuzzy search techniques to evaluate the relevance of URL links in relation to the Language Model (LLM) response. By employing fuzzy matching algorithms, we assessed the contextual appropriateness of each URL link within the response.
2. *URL Validation*: Additionally, we implemented a mechanism to validate the URLs retrieved within the responses. This involved making requests to the URLs to confirm their validity. By verifying the responsiveness of the URLs, we ensured that users were directed to reliable and functional sources of information.

By combining these strategies, we aimed to enhance the quality and relevance of URL links provided within the chatbot responses, thereby improving the overall user experience and ensuring the delivery of accurate and valuable content.

Authorized licensed use limited to: DePaul University. Downloaded on July 02,2025 at 04:59:39 UTC from IEEE Xplore. Restrictions apply.

H. Challenge 8: LLM Hallucination

The issue of LLM hallucination arises when the chatbot engages in circular conversations with itself, failing to address user queries effectively.

Prompt optimization emerges as a key strategy to mitigate LLM hallucination. By refining the prompts used to initiate interactions, we can guide the chatbot towards more coherent responses, reducing the likelihood of circular communication patterns. Through prompt optimization, we aim to refocus the chatbot's attention on addressing user queries directly, thereby enhancing its overall effectiveness and user experience.

I. Challenge 9: Production-Level Deployment Challenges

Deploying the system to production posed several challenges that required robust solutions:

1. *Autoscaling Based on Resource Usage*: Implementing autoscaling mechanisms based on resource usage helped dynamically adjust the system's capacity to handle varying workloads. This ensured optimal performance and resource utilization, even during peak periods.
2. *Logging, Tracking, and Monitoring*: Leveraging Elasticsearch and Kibana facilitated comprehensive logging, tracking, and monitoring of system activities and performance metrics. This enabled real-time visibility into system health and allowed for proactive identification and resolution of issues.
3. *Setting up Alerts for API/Deployment Failure*: Configuring alerts for API or deployment failures enabled timely notification of any issues that may arise, allowing for prompt investigation and remediation. This proactive approach minimized downtime and ensured uninterrupted service availability.
4. *Implementing Liveness Probes and Automatic Restart*: By incorporating liveness probes to check the health of APIs periodically, coupled with automatic restart mechanisms in case of failure, we ensured continuous availability and reliability of the system. This proactive approach to system health monitoring and management minimized service disruptions and enhanced overall stability.

By implementing these solutions, we effectively addressed production-level deployment challenges, ensuring scalability, reliability, and optimal performance of the system in a production environment.

VIII. Conclusion and Future Directions

Based on the current functionalities and observed limitations, several avenues for improvement and expansion are recommended for future development:

- *Expansion to Social and Institutional Affairs*: Enhance the bot's capability to address inquiries regarding social policies, institutional affairs, and key personnel. Ensure the information provided remains up-to-date and accurate.
- *Update on RHS Stances*: Incorporate updates regarding the Royal Horticultural Society's latest stances on topics such as feeding, lawns, and biodiversity. This ensures that responses remain relevant and reflect the organization's current positions.
- *Integration with RHS Garden Events and Developments*: Link the bot with information on upcoming RHS shows and developments in RHS Gardens. This feature enhances user engagement and provides timely updates on events and projects.
- *Utilization of Find a Plant Tool*: Integrate or reference the Find a Plant tool to assist users in choosing suitable plants. This functionality enhances the user experience by providing valuable resources for plant selection.
- *Enhanced Tone Recognition*: Improve the bot's ability to recognize and adapt to the tone of voice used by users. Ensure responses are appropriately polite and avoid unnecessary self-references as a 'Royal Horticultural Society expert' unless contextually relevant.
- *Standardization of Spelling and Geographic Context*: Ensure consistency in spelling conventions, minimizing the use of American spellings where applicable. Additionally, ensure responses reflect appropriate geographic contexts to avoid confusion and inaccuracies.
- *Improved Handling of Time-Related Queries*: Enhance the bot's ability to respond to time-related queries accurately. Develop mechanisms to address inquiries regarding the timing of specific gardening activities or events, providing relevant and helpful information.

Addressing these areas of improvement will enhance the bot's functionality, user experience, and overall effectiveness in providing valuable assistance and information to users interested in horticulture and related topics.

References

[1] A. Ramesh *et al.*, "Zero-Shot Text-to-Image Generation," *Proc Mach Learn Res*, vol. 139, pp. 8821–8831, Feb. 2021, Accessed: Mar. 17, 2024. [Online]. Available: https://arxiv.org/abs/2102.12092v2

[2] "Introducing ChatGPT." Accessed: Mar. 17, 2024. [Online]. Available: https://openai.com/blog/chatgpt#OpenAI

[3] "Amazon Alexa Voice AI | Alexa Developer Official Site." Accessed: Mar. 17, 2024. [Online]. Available: https://developer.amazon.com/en-US/alexa

[4] L. Zhou, J. Gao, D. Li, and H.-Y. Shum, "The Design and Implementation of XiaoIce, an Empathetic Social Chatbot," *Computational Linguistics*, vol. 46, no. 1, pp. 53–93, Mar. 2020, doi: 10.1162/coli_a_00368.

[5] D. Adiwardana *et al.*, "Towards a Human-like Open-Domain Chatbot," Jan. 2020, Accessed: Mar. 17, 2024. [Online]. Available: https://arxiv.org/abs/2001.09977v3

[6] K. Shuster *et al.*, "BlenderBot 3: a deployed conversational agent that continually learns to responsibly engage," Aug. 2022, Accessed: Mar. 17, 2024. [Online]. Available: https://arxiv.org/abs/2208.03188v3

[7] S. Vakayil, D. S. Juliet, Anitha. J, and S. Vakayil, "RAG-Based LLM Chatbot Using Llama-2," in *2024 7th International Conference on Devices, Circuits and Systems (ICDCS)*, IEEE, Apr. 2024, pp. 1–5. doi: 10.1109/ICDCS59278.2024.10561020.

[8] M. A. Quidwai and A. Lagana, "A RAG Chatbot for Precision Medicine of Multiple Myeloma," *medRxiv*, 2024, doi: 10.1101/2024.03.14.24304293.

Authorized licensed use limited to: DePaul University. Downloaded on July 02,2025 at 04:59:39 UTC from IEEE Xplore. Restrictions apply.

[9] D. Reynolds, "Gaussian Mixture Models," in *Encyclopedia of Biometrics*, Boston, MA: Springer US, 2009, pp. 659–663. doi: 10.1007/978-0-387-73003-5_196.

[10] K. Knill and S. Young, "Hidden Markov Models in Speech and Language Processing," 1997, pp. 27–68. doi: 10.1007/978-94-017-1183-8_2.

[11] I. J. Goodfellow *et al.*, "Generative Adversarial Nets," *Adv Neural Inf Process Syst*, vol. 27, 2014, Accessed: Feb. 11, 2024. [Online]. Available: http://www.github.com/goodfeli/adversarial

[12] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553. Nature Publishing Group, pp. 436–444, May 27, 2015. doi: 10.1038/nature14539.

[13] T. Mikolov, M. Karafiát, L. Burget, J. C.- Interspeech, and undefined 2010, "Recurrent neural network based language model.," *fit.vutbr.cz*, Accessed: Feb. 11, 2024. [Online]. Available: http://www.fit.vutbr.cz/research/groups/speech/servite/2010/rnnlm_mikolov.pdf

[14] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language Modeling with Gated Convolutional Networks," *34th International Conference on Machine Learning, ICML 2017*, vol. 2, pp. 1551–1559, Dec. 2016, doi: 10.48550/arxiv.1612.08083.

[15] A. Graves, "Long Short-Term Memory," in *Supervised Sequence Labelling with Recurrent Neural Networks. Studies in Computational Intelligence*, vol. 385, Springer,Berlin,Heidelberg, 2012, pp. 37–45. doi: 10.1007/978-3-642-24797-2_4.

[16] A. Vaswani *et al.*, "Attention is All you Need," *Adv Neural Inf Process Syst*, vol. 30, 2017.

[17] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *insightcivic.s3.us-east-1.amazonaws …*, Accessed: Feb. 11, 2024. [Online]. Available: https://insightcivic.s3.us-east-1.amazonaws.com/language-models.pdf

[18] T. Brown *et al.*, "Language Models are Few-Shot Learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., Curran Associates, Inc., 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[19] "Pathways Language Model (PaLM): Scaling to 540 Billion Parameters for Breakthrough Performance – Google Research Blog." Accessed: Feb. 11, 2024. [Online]. Available: https://blog.research.google/2022/04/pathways-language-model-palm-scaling-to.html

[20] B. Workshop *et al.*, "BLOOM: A 176B-Parameter Open-Access Multilingual Language Model," Nov. 2022, Accessed: Feb. 11, 2024. [Online]. Available: https://arxiv.org/abs/2211.05100v4

[21] "Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, the World's Largest and Most Powerful Generative Language Model - Microsoft Research." Accessed: Feb. 11, 2024. [Online]. Available: https://www.microsoft.com/en-us/research/blog/using-deepspeed-and-megatron-to-train-megatron-turing-nlg-530b-the-worlds-largest-and-most-powerful-generative-language-model/

[22] OpenAI *et al.*, "GPT-4 Technical Report," Mar. 2023, Accessed: Feb. 11, 2024. [Online]. Available: https://arxiv.org/abs/2303.08774v4

[23] H. Touvron *et al.*, "LLaMA: Open and Efficient Foundation Language Models," Feb. 2023, Accessed: Feb. 11, 2024. [Online]. Available: https://arxiv.org/abs/2302.13971v1

[24] P. Lewis *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," May 2020.

[25] L. Tunstall *et al.*, "Efficient Few-Shot Learning Without Prompts," Sep. 2022, Accessed: Feb. 11, 2024. [Online]. Available: https://arxiv.org/abs/2209.11055v1

[26] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pp. 3982–3992, Aug. 2019, doi: 10.18653/v1/d19-1410.

[27] "RateLimitError | OpenAI Help Center." Accessed: Feb. 11, 2024. [Online]. Available: https://help.openai.com/en/articles/6897202-ratelimiterror

[28] "Vector database - Milvus." Accessed: Feb. 11, 2024. [Online]. Available: https://milvus.io/

[29] J. Johnson, M. Douze, and H. Jegou, "Billion-scale similarity search with GPUs," *IEEE Trans Big Data*, vol. 7, no. 3, pp. 535–547, Feb. 2017, doi: 10.1109/TBDATA.2019.2921572.

[30] "langchain.text_splitter.RecursiveCharacterTextSplitter — LangChain 0.1.4." Accessed: Feb. 11, 2024. [Online]. Available: https://api.python.langchain.com/en/latest/text_splitter/langchain.text_splitter.RecursiveCharacterTextSplitter.html#

[31] "New and improved embedding model." Accessed: Feb. 11, 2024. [Online]. Available: https://openai.com/blog/new-and-improved-embedding-model

[32] Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs," *IEEE Trans Pattern Anal Mach Intell*, vol. 42, no. 4, pp. 824–836, Mar. 2016, doi: 10.1109/TPAMI.2018.2889473.

[33] M. Aumüller, E. Bernhardsson, and A. Faithfull, "ANN-Benchmarks: A benchmarking tool for approximate nearest neighbor algorithms," *Inf Syst*, vol. 87, p. 101374, Jan. 2020, doi: 10.1016/j.is.2019.02.006.

[34] S. Dey, V. Vinayakarao, M. Gupta, and S. Dechu, "Evaluating Commit Message Generation: To BLEU Or Not To BLEU?," Apr. 2022, doi: 10.1145/3510455.3512790.

[35] L. C.-Y.-P. of the W. on T. Summarization and undefined 2004, "Rouge: A package for automatic evaluation of summaries," *cir.nii.ac.jp*, Accessed: Mar. 17, 2024. [Online]. Available: https://cir.nii.ac.jp/crid/1571417125576321408

[36] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," Apr. 2019.

[37] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," Oct. 2019, Accessed: Mar. 17, 2024. [Online]. Available: https://arxiv.org/abs/1910.01108v4

[38] "GitHub - tiangolo/fastapi: FastAPI framework, high performance, easy to learn, fast to code, ready for production." Accessed: Mar. 17, 2024. [Online]. Available: https://github.com/tiangolo/fastapi

[39] "GitHub - kubernetes/kubernetes: Production-Grade Container Scheduling and Management." Accessed: Mar. 17, 2024. [Online]. Available: https://github.com/kubernetes/kubernetes

[40] "GitHub - elastic/kibana: Your window into the Elastic Stack." Accessed: Mar. 17, 2024. [Online]. Available: https://github.com/elastic/kibana

Authorized licensed use limited to: DePaul University. Downloaded on July 02,2025 at 04:59:39 UTC from IEEE Xplore. Restrictions apply.