# BeefBot: Harnessing Advanced LLM and RAG Techniques for Providing Scientific and Technology Solutions to Beef Producers

**Zhihao Zhang[1], Carrie-Ann Wilson[1], Rachel Hay[1], Yvette Everingham[1], Usman Naseem[2]**
James Cook University, Australia[1], Macquarie University, Australia[2]
{zhihao.zhang, carrie.wilson1, rachel.hay, yvette.everingham}@jcu.edu.au
usman.naseem@mq.edu.au

## Abstract

We propose **BeefBot**, a LLM-powered chatbot designed for beef producers. It retrieves the latest agricultural technologies (AgTech), practices and scientific insights to provide rapid, domain-specific advice, helping to address on-farm challenges effectively. While generic Large Language Models (LLMs) like ChatGPT are useful for information retrieval, they often hallucinate and fall short in delivering tailored solutions to the specific needs of beef producers, including breed-specific strategies, operational practices, and regional adaptations.There are two common methods for incorporating domain-specific data in LLM applications: Retrieval-Augmented Generation (RAG) and fine-tuning. However, their respective advantages and disadvantages are not well understood. Therefore, we implement a pipeline to apply RAG and fine-tuning using an open-source LLM in BeefBot and evaluate the trade-offs. By doing so, we are able to select the best combination as the backend of BeefBot, delivering actionable recommendations that enhance productivity and sustainability for beef producers with fewer hallucinations. Key benefits of BeefBot include its accessibility as a web-based platform compatible with any browser, continuously updated knowledge through RAG, confidential assurance via local deployment, and a user-friendly experience facilitated by an interactive website. The demo of the BeefBot can be accessed at https://www.youtube.com/watch?v=r7mde1EOG4o.

## 1 Introduction

The latest development of Large Language Models (LLMs) has advanced the field of Natural Language Processing (NLP), delivering a strong foundation for a wide range of potential applications. However, applying generic LLMs to solve domain-specific problems presents several challenges, such as understanding domain objects' uniqueness, aligning domain's diversity of constraints, and producing consistent domain-related contents (Ling et al., 2023). In the context of the beef industry within the agricultural sector, these challenges are particularly pronounced. With a generational shift in farming, many younger producers may not be fully versed in traditional practices, underscoring the importance of accessible, digital platforms that offer instant access to a wealth of historical and cutting-edge knowledge. Furthermore, the unique challenge of the beef industry lies in the rapid pace of development in agricultural technologies and innovations, coupled with frequent updates to government regulations and guidelines, necessitating a tool that can provide up-to-date, reliable advice.

To address these challenges, recent studies have pursued two primary methods of knowledge injection (Wang et al., 2021; Chen et al., 2022) in LLMs including fine-tuning and Retrieval Augmentation Generation (RAG) (Ovadia et al., 2024). While both approaches can improve LLMs' responses with precision and concision, fine-tuning incorporates additional domain knowledge into the model, whereas RAG prompts the model with external data (Balaguer et al., 2024). Given the variability in different domains, these methods have been applied accordingly in several areas for LLM applications, including health (Singhal et al., 2023), finance (Yang et al., 2023) and agriculture (Arora et al., 2020). Although both methods can be utilised for adopting LLMs in new domains, most of the existing models tend to utilise either fine-tuning or retrieval augmentation with prompting, and their respective advantages and disadvantages are not well studied. Furthermore, agriculture includes sub-fields like horticulture, arable farming, animal husbandry and forestry, which is still too general for direct industry application purposes. Additionally, the lack of easily accessible platforms also prevents the adoption of domain-specific LLM in industrial settings, often due to the required expertise in deep-learning and programming.

*Proceedings of the 31st International Conference on Computational Linguistics: System Demonstrations*, pages 54–62
January 19–24, 2025. ©2025 Association for Computational Linguistics

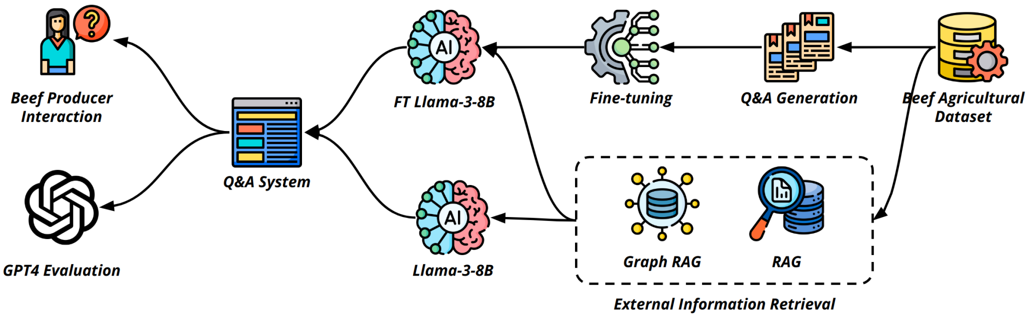


Figure 1: Overview of Beef Agriculture Pipeline for LLM

**Contribution.** With this in mind, we propose BeefBot, a LLM-powered web-based interactive chatbot, designed to support beef producers by providing immediate actionable recommendations and long-term strategies for their specific on-farm problems and goals. Its primary function is providing optimal solutions from the available knowledge in the beef industry, while taking into account farm-related variables such as economics, cattle breeds, grazing land management, and drought resilience strategies. More precisely, we first developed a pipeline to evaluate the impact of RAG and fine-tuning techniques on the performance of open-source LLM in the beef agriculture domain. The LLM was equipped with the optimal combination of these techniques, forming the complete architecture of BeefBot's backend. It can deliver answers through an interactive website similar to ChatGPT (Ouyang et al., 2022), with significantly reduced hallucination in out-of-knowledge response. This enhancement is designed to provide more precise and relevant responses for beef producers, thereby enabling them to focus more on implementation.

## 2 Beef Agriculture Pipeline for LLM

The pipeline is designed to utilize open-source LLMs to generate comprehensive responses for beef agriculture-specific questions. Its structure is shown in Figure 1. The beef agriculture dataset is gathered from trustworthy sources [1], comprising diverse content from text, podcasts, and videos. Following the data collection, we generate question-and-answer pairs for model fine-tuning and implement both original RAG and knowledge graph RAG. This aims to leverage different methods for improving the LLM responses in the beef agriculture domain.

### 2.1 Data Collection

We implemented a comprehensive data collection pipeline to extract and collect information from several trustworthy websites. This process includes two primary components: a web parsing algorithm and a resource downloading subroutine. The web parsing algorithm aims to scrape raw text data from source websites while removing any sensitive information, such as participants' names and business information in interviews or case studies. During web scraping, the algorithm identifies the webpage structure, removes trivial information such as web headers and social media links, but captures multimodal resources, including podcast and video content. The resource downloading subroutine targets those available multimodal contents and utilises open-source tools [2] [3] to download them. The downloaded multimodal content was further transcribed into text by the latest speech-to-text model Whisper (Radford et al., 2023). Both scrapped and transcript raw text are stored into plain text documents and indexed by their titles and source URLs. Together, this formed a comprehensive data collection, enabling us to provide the LLM agriculture pipeline with information and resources from the multimodal context in the beef industry domain.

### 2.2 Fine-tuning QA Generation

High-quality and contextually grounded questions that comprehensively reflect the collected text are essential for language model fine-tuning. Inspired by Alpaca (Taori et al., 2023), we utilize Llama-index [4] to transform the plain text into instructive question-answer pairs as the fine-tuning dataset. To

[1] Details on the specific websites utilised are withheld due to intellectual property concerns. It is important to note that all data collection and processing activities were conducted in compliance with ethical standards and considerations.

[2] https://github.com/yt-dlp/yt-dlp

[3] https://github.com/spotDL/spotify-downloader

[4] https://docs.llamaindex.ai

GPT-4

```
You are a Teacher/ Professor. Your task is to setup a quiz/examination. Using the provided
context, formulate 5 question-answer pairs that captures an important fact from the context.
<content> text chunk </content>

You MUST obey the following criteria:
- Restrict the question to the context information provided.
- Do NOT create a question that cannot be answered from the context.
- Phrase the question so that it does NOT refer to specific context and person. For instance, do
NOT put phrases like "given provided context" or "in this work" "How Jerry deal with" in the
question, because if the question is asked elsewhere it wouldn't be provided specific context.
Replace these terms with specific details.

- BAD questions:
What did the author do in his childhood
What were the main findings in this report
- GOOD questions:
What did the farmer do in his farm
What were the main findings in the original Transformers paper.

- Return your response in JSON format
Generate question:
Generate answer:
```

Figure 2: Fine-tuning QA Generation

achieve this, we split the long text documents into small text chunks of 2,000 characters and combine them alongside a carefully crafted prompt, following the Guidance framework [5]. For each text chunk, we utilise GPT-4 with the complete prompt to generate five specific question-answer pairs. This singular and unified process ensures the relevance and coherence of each question-answer pair given the source text. The prompt is shown in Figure 2, and the question-answer pairs are saved as instructive data instances in a JSON file. We divided the collected 24,057 instances into two sets: 19,245 for training and 4,812 for testing.

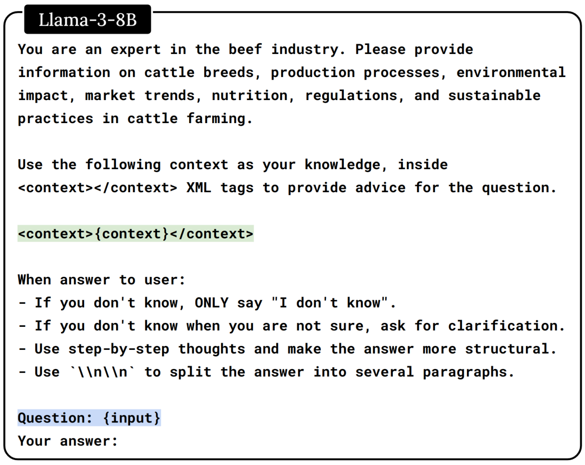
## 2.3 Model Fine-tuning

Model fine-tuning can inject factual knowledge into LLM parameters and provide promising results for completing in-domain tasks. We fine-tune the Llama-3 which is the latest generation of Llama model family. Llama (Touvron et al., 2023a) is an open-source autoregressive LLM based on the transformer architecture (Vaswani et al., 2023), comparable to GPT-3 (Brown et al., 2020). It leverages three main improvements over prior proposed models, including pre-normalisation (Brown et al., 2020), SwiGLU activation function (Chowdhery et al., 2023) and rotary embedding (Black et al., 2022). Llama-3 (Llama Team, 2024) is the third generation of Llama, which competes with ChatGPT (Ouyang et al., 2022). It features with twice context windows and more training data compared to the Llama-2 (Touvron et al., 2023b). These enhancements, along with grouped-query attention, enable Llama-3 to outperform many open-source LLMs such as Mistral and Gemma on reasoning, coding, and knowledge tests, indicating its capability in diverse tasks (Llama Team, 2024). Therefore, we fine-tune and validate the Llama-3-8B (Llama Team, 2024) model with the collected instruct data. The entire model is trained using paged AdamW for a single epoch, with a warm-up step of 100 and learning rate of 1e-5. Our implementation is based on HuggingFace Transformers (Wolf et al., 2020), following the instructions from Alpaca (Taori et al., 2023). To optimize the fine-tuning process, we deployed it with Fully Sharded Data Parallelism (FSDP) (Zhao et al., 2023), which allows the sharding of model weights, optimizer states, and gradients, enabling the efficient use of multiple GPUs in parallel. The entire fine-tuning process utilised 3 NVIDIA H100 GPUs over a duration of 5 hours.

## 2.4 Retrieval Augmentation Generation

Another method to improve the response from LLM is Retrieval Augmentation Generation (RAG) (Lewis et al., 2020). This method aims to extend LLM capability to precisely manipulate knowledge and handle out-of-knowledge queries to reduce hallucination in knowledge-intensive tasks. To prompt engineer the Llama-3 model with RAG system, there are three components involved in establishing: 1) vector database for knowledge con-

[5] https://github.com/guidance-ai/guidance



Llama-3-8B

```
You are an expert in the beef industry. Please provide
information on cattle breeds, production processes, environmental
impact, market trends, nutrition, regulations, and sustainable
practices in cattle farming.

Use the following context as your knowledge, inside
<context></context> XML tags to provide advice for the question.

<context>{context}</context>

When answer to user:
- If you don't know, ONLY say "I don't know".
- If you don't know when you are not sure, ask for clarification.
- Use step-by-step thoughts and make the answer more structural.
- Use `\\n\\n` to split the answer into several paragraphs.

Question: {input}
Your answer:
```

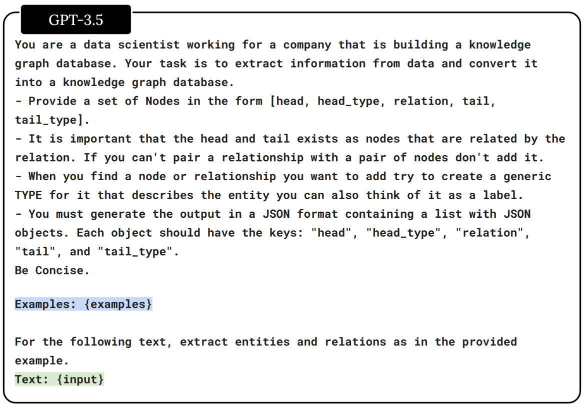
Figure 3: RAG Prompt

text reference, 2) model serving for instant inference, and 3) prompt designing for hallucination reduction.

**Vector Database.** We utilise Chromadb [6] to build a large-scale vector database with all the available documents collected from trustworthy sources. This vector database is constructed by truncating and embedding the collected document from Section 2.1 into text blocks, with a maximum of 500 tokens per block for context-related reference. The vector database can be continuously updated with the latest external resources, and we index all documents with unique IDs along with their original source URLs in the vector database to maintain the traceability of each text block.

**Model Serving.** The fine-tuned model is served via Ollama [7], with a temperature of 0.1 and a repeat penalty of 1.15, to respond to in-coming queries with external context. Unlike traditional deep-learning pipelines that require complex dependencies and initial model loading for the first launch, Ollama serves the model as a system-wide service via a Docker-like container, making model inference simpler and faster.

**Prompt Designing.** As shown in Figure 3, we integrate LangChain [8] prompt templates within the RAG system by sending the most relevant text excerpts from the vector database, along with the queries, to the model. This approach allows the model's responses to include both its internal knowledge and external in-domain knowledge with proper references. To minimise hallucination, we prompt the model with a static response "I don't know" for queries beyond its knowledge scope.



GPT-3.5

```
You are a data scientist working for a company that is building a knowledge
graph database. Your task is to extract information from data and convert it
into a knowledge graph database.
- Provide a set of Nodes in the form [head, head_type, relation, tail,
tail_type].
- It is important that the head and tail exists as nodes that are related by the
relation. If you can't pair a relationship with a pair of nodes don't add it.
- When you find a node or relationship you want to add try to create a generic
TYPE for it that describes the entity you can also think of it as a label.
- You must generate the output in a JSON format containing a list with JSON
objects. Each object should have the keys: "head", "head_type", "relation",
"tail", and "tail_type".
Be Concise.

Examples: {examples}

For the following text, extract entities and relations as in the provided
example.
Text: {input}
```

Figure 4: Graph RAG Prompt

## 2.5 RAG with Knowledge Graph

RAG with knowledge graph or Graph RAG (Edge et al., 2024) is a updated version of RAG, which enhances the model capability to answer global questions requiring the understanding of an entire document. Based on the original RAG, there are two extra stages involved: 1) deriving an entity-based knowledge graph from source document and 2) related entities' community summaries pre-generation. Apart from these changes, the model serving remains the same for consistency.

**Knowledge Graph Derivation.** Collected documents are spited into manageable text chunks and each text chunk is further processed to identify and extract their entities and relationships. To ensure a comprehensive extraction while maintaining cost-effectiveness, we utilise GPT-3.5-turbo with multipart prompts, demonstrated in Figure 4. The extracted entities and relationships are then summarised into single descriptive blocks for each graph element. We incorporate Neo4j [9] to store the graph elements and build an undirected weighted graph, where entities are transformed as nodes and relationships are transformed as edges.

**Community Summaries Pre-generation.** For each community in the Neo4j graph database, report-like summaries are generated, which provides an overview of communities' semantics. When a question is received, relevant community summaries are retrieved for answering the question based on the relevance. The final answer is then generated by summarising all the summaries to provide a comprehensive response.

## 2.6 Pipeline Evaluation

To better understand the benefits of each method for LLM in the beef agriculture domain, we evaluated

[6] https://www.trychroma.com/

[7] https://ollama.ai/

[8] https://www.langchain.com/

[9] https://neo4j.com

| Llama3-8B | Relevance | Groundedness | Helpfulness |
|---|---|---|---|
| OG | 75.12 | 78.74 | **74.49** |
| OG-RAG | 76.14 | 78.35 | 72.76 |
| OG-GRAG | **76.46** | **79.37** | 73.31 |
| FT | **78.19** | **80.39** | **74.49** |
| FT-RAG | 68.11 | 73.54 | 63.78 |
| FT-GRAG | 69.76 | 76.14 | 66.46 |

Table 1: Evaluation results. "OG" represents Original Llama3-8B model. "FT" represents Fine-Tuned Llama3-8B model. "GRAG" represents Graph RAG.

different combinations using the same evaluation metric. The combinations include both the original Llama-3 and fine-tuned Llama-3, with and without RAG and Graph-RAG systems. Since human evaluation is expensive and non-experts cannot determine the correctness of the technical answers, we utilised GPT-4 as an evaluator by providing the ground-truth answers as guidance.

**Evaluation Setup.** Following the similar idea in Section 2.2, we applied GPT-4 to generate 200 question-answer pairs from the collected beef agriculture documents. The detailed evaluation generation prompts is shown in Appendix. After filtering out the duplicate topics, there is a total number of 127 question-answer pairs that can represent the ground-truth dataset. We prompt the models with the questions and provide GPT-4 with the ground truth answers and model generated answers for evaluation.

**Evalution Metrics.** To better reflect the application in the industry domain, we introduce three different evaluation metrics: **1) Relevance**: How closely the model answer addresses the specific question. **2) Groundedness**: The correctness of the answer compared with the ground truth. **3) Helpfulness**: The usefulness the answer can be utilised or implemented by a beef farmer. For each metric, GPT-4 will provide a score from 1 to 10, where 1 is the worst and 10 is the best. We take the mean value of each combination and linearly scale up the scores to 100 for evaluation.

**Evaluation Result.** The evaluation results are summarise in Table 1. Compared with original Llama3-8B model, both RAG system and fine-tuned model have better performance in relevance and groundedness. The Helpfulness are slightly worse in RAG system and remains the same in fine-tuned models. This might due to the technical knowledge injection into the model, which lead to more technical language during question answering. For original Llama3-8B model, Graph RAG improve its answer in all three metrics comparing with original RAG. For fine-tuned Llama3-8B model, it's worth noting that the integration of RAG and Graph RAG decrease the performance of the model. This might be caused by catastrophic forgetting where model can loss its major reasoning capability while acquiring new domain knowledge (Luo et al., 2024). However, we observe that even under this circumstance, Graph RAG still outperform RAG in all three metrics. We also compare the performance with the proprietary models, although their performance are better than the open-source models, the concerns about privacy and cost-efficiency preventing deploying them into real-world application.

## 3 BeefBot Architecture

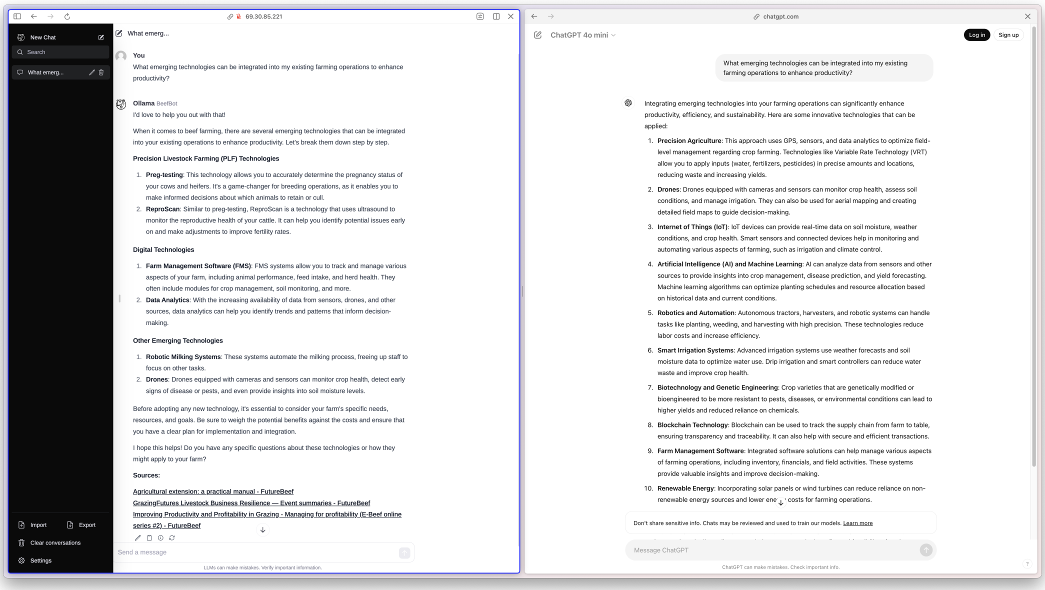
According to the observation from the beef agriculture pipeline, we propose a chatbot named BeefBot. It offers beef producers a well-designed, interactive Graphic User Interface (GUI) accessible through the web browser. Therefore, we designed both back-end and front-end of the system to ensure accuracy and efficiency of answering relevant beef production questions. This equips beef producers to tackle specific agricultural challenges with technology and scientific insights. Its design is shown in Figure 1, and the backend and front-end are described in the following sections.

### 3.1 Backend System

To streamline the application and simplify deployment, BeefBot's backend is constructed using a Flask HTTP server [10] with API calls. More specifically, the Llama3-8B model is hosted on RunPod [11] with Ollama API exposure. The Graph RAG database is stored on a Neo4j database, and the retrieval information is available during Ollama API calls. Langchain fuses the backend logic and provides a textual response to post via the Flask HTTP server. When there is a user query, Langchain retrieves the Graph RAG context and sends it to the Llama3-8B model along with the original query via an Ollama API call. The returned model response is extracted and formatted as a markdown text for the Flask HTTP server for front-end posting.

[10] https://flask.palletsprojects.com/

[11] https://www.runpod.io/



What emerg...

**You**

What emerging technologies can be integrated into my existing farming operations to enhance productivity?

**Ollama** BeefBot

I'd love to help you out with that!

When it comes to beef farming, there are several emerging technologies that can be integrated into your existing operations to enhance productivity. Let's break them down step by step.

**Precision Livestock Farming (PLF) Technologies**

1. **Preg-testing**: This technology allows you to accurately determine the pregnancy status of your cows and heifers. It's a game-changer for breeding operations, as it enables you to make informed decisions about which animals to retain or cull.
2. **ReproScan**: Similar to preg-testing, ReproScan is a technology that uses ultrasound to monitor the reproductive health of your cattle. It can help you identify potential issues early on and make adjustments to improve fertility rates.

**Digital Technologies**

1. **Farm Management Software (FMS)**: FMS systems allow you to track and manage various aspects of your farm, including animal performance, feed intake, and herd health. They often include modules for crop management, soil monitoring, and more.
2. **Data Analytics**: With the increasing availability of data from sensors, drones, and other sources, data analytics can help you identify trends and patterns that inform decision-making.

**Other Emerging Technologies**

1. **Robotic Milking Systems**: These systems automate the milking process, freeing up staff to focus on other tasks.
2. **Drones**: Drones equipped with cameras and sensors can monitor crop health, detect early signs of disease or pests, and even provide insights into soil moisture levels.

Before adopting any new technology, it's essential to consider your farm's specific needs, resources, and goals. Be sure to weigh the potential benefits against the costs and ensure that you have a clear plan for implementation and integration.

I hope this helps! Do you have any specific questions about these technologies or how they might apply to your farm?

**Sources:**

Agricultural extension: a practical manual - FutureBeef
GrazingFutures Livestock Business Resilience — Event summaries - FutureBeef
Improving Productivity and Profitability in Grazing - Managing for profitability (E-Beef online series #2) - FutureBeef

What emerging technologies can be integrated into my existing farming operations to enhance productivity?

Integrating emerging technologies into your farming operations can significantly enhance productivity, efficiency, and sustainability. Here are some innovative technologies that can be applied:

1. **Precision Agriculture**: This approach uses GPS, sensors, and data analytics to optimize field-level management regarding crop farming. Technologies like Variable Rate Technology (VRT) allow you to apply inputs (water, fertilizers, pesticides) in precise amounts and locations, reducing waste and increasing yields.
2. **Drones**: Drones equipped with cameras and sensors can monitor crop health, assess soil conditions, and manage irrigation. They can also be used for aerial mapping and creating detailed field maps to guide decision-making.
3. **Internet of Things (IoT)**: IoT devices can provide real-time data on soil moisture, weather conditions, and crop health. Smart sensors and connected devices help in monitoring and automating various aspects of farming, such as irrigation and climate control.
4. **Artificial Intelligence (AI) and Machine Learning**: AI can analyze data from sensors and other sources to provide insights into crop management, disease prediction, and yield forecasting. Machine learning algorithms can optimize planting schedules and resource allocation based on historical data and current conditions.
5. **Robotics and Automation**: Autonomous tractors, harvesters, and robotic systems can handle tasks like planting, weeding, and harvesting with high precision. These technologies reduce labor costs and increase efficiency.
6. **Smart Irrigation Systems**: Advanced irrigation systems use weather forecasts and soil moisture data to optimize water use. Drip irrigation and smart controllers can reduce water waste and improve crop health.
7. **Biotechnology and Genetic Engineering**: Crop varieties that are genetically modified or bioengineered to be more resistant to pests, diseases, or environmental conditions can lead to higher yields and reduced reliance on chemicals.
8. **Blockchain Technology**: Blockchain can be used to track the supply chain from farm to table, ensuring transparency and traceability. It can also help with secure and efficient transactions.
9. **Farm Management Software**: Integrated software solutions can help manage various aspects of farming operations, including inventory, financials, and field activities. These systems provide valuable insights and improve decision-making.
10. **Renewable Energy**: Incorporating solar panels or wind turbines can reduce reliance on non-renewable energy sources and lower ene costs for farming operations.

Don't share sensitive info. Chats may be reviewed and used to train our models. Learn more

Figure 5: Comparison between front-end responses from BeefBot and ChatGPT for the same question. BeefBot (left) features responses tailored to the beef industry domain, while ChatGPT (right) offers generic responses.

### 3.2 Interactive Front-end Website

BeefBot offers an interactive web front-end built on the ollama-webui lite [12]. As shown on the left side of Figure 5, beef producers can interact with BeefBot by typing their questions into the text bar located at the bottom of the webpage. By integrating the BeefBot with external resources through the Graph RAG, its responses include all the source URLs referred to in the context. These clickable links guide beef producers to the websites of the mentioned techniques, helping them find the optimal solution for their specific problems without extensive web browsing. The conversations are searchable from the sidebar, which helps the user to find the previous information efficiently. To ensure privacy, all chat history is stored in the random access memory (RAM) of the host machine. Therefore, exiting each web session or clicking "Clear conversations" in the sidebar wipes out the entire conversation with BeefBot. We also provide a method that allows the user to export or import their conversations for continuous usage.

### 3.3 Case Study Comparison

As demonstrated in Figure 5, we compare the responses from BeefBot and ChatGPT to the same questions likely to be asked by beef producers. Even though the question does not specifically mention the beef industry, BeefBot tailors its response in that direction with actionable suggestions, such as Preg-testing technology, ReproScan technology, and Farm Management Software. These actionable suggestions align with our 'helpfulness' metric, as they reflect practical solutions that beef producers can implement, which evaluates the utility of answers in real-world applications, ensuring they are actionable and tailored to user needs. In contrast, ChatGPT tends to provide general answers within the broader agriculture domain, some of which are only high-level concepts, including artificial intelligence, robotics, and blockchain. Moreover, ChatGPT's responses rely solely on its internal knowledge without any external references, making it challenging to verify their correctness.

## 4 Conclusion

We propose BeefBot, a web-based interactive chatbot powered by a LLM, designed to offer precise, immediate, and long-term solutions to beef producers by leveraging the available knowledge in the beef industry. BeefBot's architecture, including its RAG system, model serving with Ollama, and frontend interface, is largely domain-independent and can be reused across sectors. Domain adaptation primarily requires collecting domain-specific data, generating high-quality Q&A datasets, and fine-tuning the LLM accordingly. This process would typically involve moderate effort depending on the data availability and complexity of domain-specific tasks.

[12] https://github.com/ollama-webui/ollama-webui-lite

## References

Bhavika Arora, Dheeraj Singh Chaudhary, Mahima Satsangi, Mahima Yadav, Lotika Singh, and Prem Sewak Sudhish. 2020. Agribot: A natural language generative neural networks engine for agricultural applications. In *2020 International Conference on Contemporary Computing and Applications (IC3A)*, pages 28–33.

Angels Balaguer, Vinamra Benara, Renato Luiz de Freitas Cunha, Roberto de M. Estevão Filho, Todd Hendry, Daniel Holstein, Jennifer Marsman, Nick Mecklenburg, Sara Malvar, Leonardo O. Nunes, Rafael Padilha, Morris Sharp, Bruno Silva, Swati Sharma, Vijay Aski, and Ranveer Chandra. 2024. Rag vs fine-tuning: Pipelines, tradeoffs, and a case study on agriculture. *Preprint*, arXiv:2401.08406.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. Gpt-neox-20b: An open-source autoregressive language model. *Preprint*, arXiv:2204.06745.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 2778–2788. ACM.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *Preprint*, arXiv:2404.16130.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Chen Ling, Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, Yun Li, Hejie Cui, Xuchao Zhang, Tianjiao Zhao, Amit Panalkar, Wei Cheng, Haoyu Wang, Yanchi Liu, Zhengzhang Chen, Haifeng Chen, Chris White, Quanquan Gu, Jian Pei, and Liang Zhao. 2023. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *Preprint*, arXiv:2305.18703.

AI@Meta Llama Team. 2024. The llama 3 herd of models. A detailed contributor list can be found in the appendix of this paper.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2024. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *Preprint*, arXiv:2308.08747.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. 2024. Fine-tuning or retrieval? comparing knowledge injection in llms. *Preprint*, arXiv:2312.05934.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, Mike Schaekermann, Amy Wang, Mohamed Amin, Sami Lachgar, Philip Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Aguera y Arcas, Nenad Tomasev, Yun Liu, Renee Wong, Christopher Semturs, S. Sara Mahdavi, Joelle Barral, Dale Webster, Greg S. Corrado, Yossi Matias, Shekoofeh Azizi, Alan Karthikesalingam, and Vivek Natarajan. 2023. Towards expert-level medical question answering with large language models. *Preprint*, arXiv:2305.09617.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

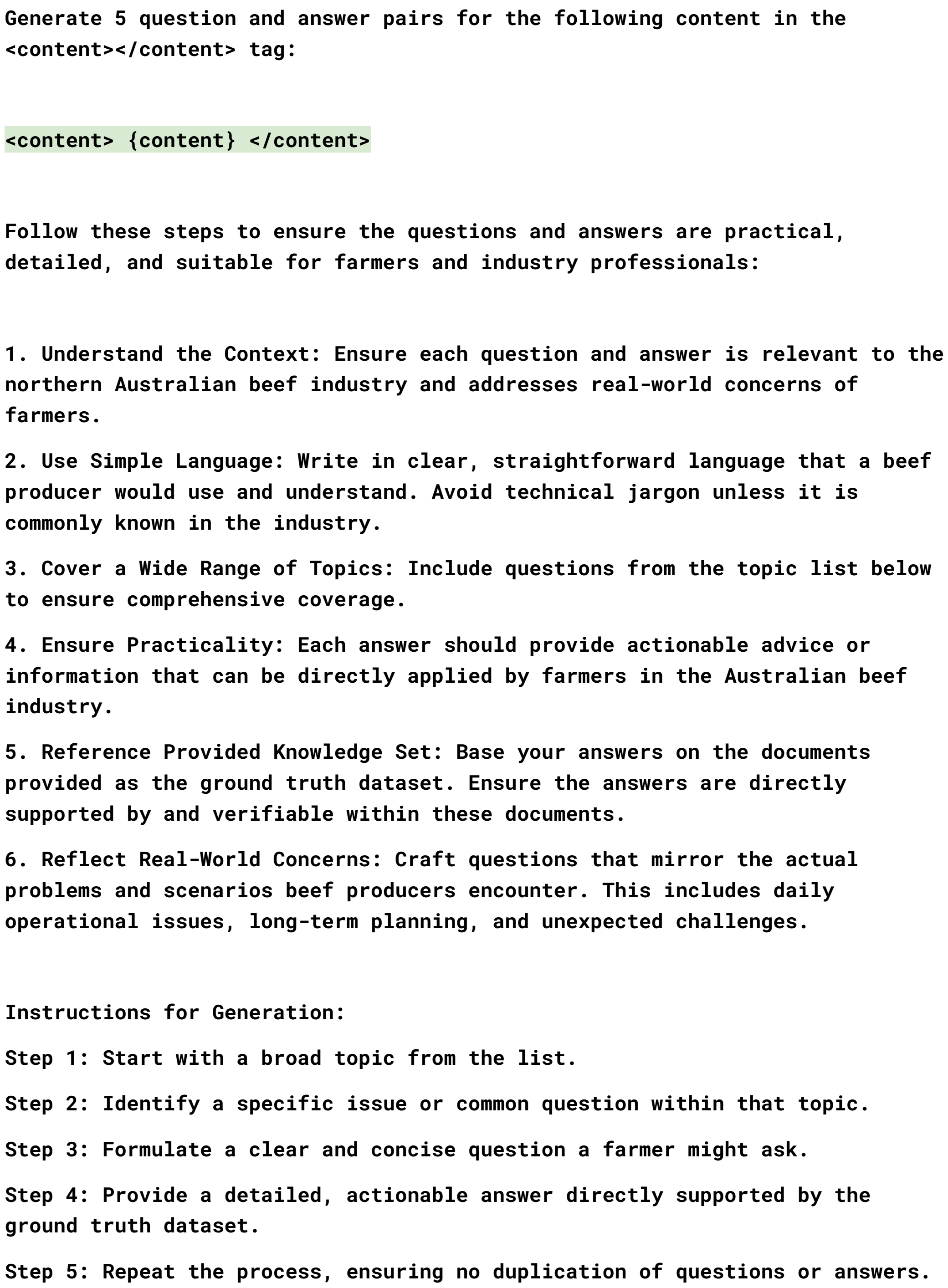
Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021. K-adapter: Infusing knowledge into pre-trained models with adapters. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1405–1418. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023. Fingpt: Open-source financial large language models. *Preprint*, arXiv:2306.06031.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. 2023. Pytorch fsdp: Experiences on scaling fully sharded data parallel. *Preprint*, arXiv:2304.11277.

# A Appendix 1



```
Generate 5 question and answer pairs for the following content in the
<content></content> tag:

<content> {content} </content>

Follow these steps to ensure the questions and answers are practical,
detailed, and suitable for farmers and industry professionals:

1. Understand the Context: Ensure each question and answer is relevant to the
northern Australian beef industry and addresses real-world concerns of
farmers.

2. Use Simple Language: Write in clear, straightforward language that a beef
producer would use and understand. Avoid technical jargon unless it is
commonly known in the industry.

3. Cover a Wide Range of Topics: Include questions from the topic list below
to ensure comprehensive coverage.

4. Ensure Practicality: Each answer should provide actionable advice or
information that can be directly applied by farmers in the Australian beef
industry.

5. Reference Provided Knowledge Set: Base your answers on the documents
provided as the ground truth dataset. Ensure the answers are directly
supported by and verifiable within these documents.

6. Reflect Real-World Concerns: Craft questions that mirror the actual
problems and scenarios beef producers encounter. This includes daily
operational issues, long-term planning, and unexpected challenges.

Instructions for Generation:

Step 1: Start with a broad topic from the list.

Step 2: Identify a specific issue or common question within that topic.

Step 3: Formulate a clear and concise question a farmer might ask.

Step 4: Provide a detailed, actionable answer directly supported by the
ground truth dataset.

Step 5: Repeat the process, ensuring no duplication of questions or answers.
```

Figure 6: Evaluation Prompt