UNIVERSITY OF CALIFORNIA
SANTA CRUZ

ROOTWISE: Utilizing RAG AI for Food Wisdom and Zero-Waste Action

A project submitted in partial satisfaction
of the requirements for the degree of

MASTER OF SCIENCE
in
COMPUTER SCIENCE AND ENGINEERING

by
Lily Faris
[lfaris@ucsc.edu]

Spring 2025

The Master's Project is approved by:

DocuSigned by:
Leilani Gilpin
A026316FFDEE48A...
Professor Leilani Gilpin, Project Chair

DocuSigned by:
Razvan Marinescu
A446AE995AA545A...
Professor Razvan Marinescu, Reader

Peter Biehl
Vice Provost and Dean of Graduate Studies

Abstract

RootWise is a conversational AI system designed to promote sustainable food practices grounded in environmental responsibility, holistic health, and community-based knowledge. It integrates retrieval-augmented generation (RAG), contextual prompting, database transparency, and visual ingredient recognition to deliver personalized guidance on zero-waste cooking, functional nutrition, and dynamically-informed culinary recommendations. RootWise aims to make sustainability actionable and approachable by helping users reduce food waste, supporting local agriculture, and engaging with food as a tool for both planetary and personal well-being. This project investigates the potential for AI systems to support climate resilience by centering consumer-level action, technological transparency, and real-world ecological wisdom.

1 Introduction

RootWise was developed in response to the intersecting challenges of food waste, environmental degradation, and the lack of community-specific technology tools that aren't for corporate profit. While large language models (LLMs) like ChatGPT offer great tools, they fail to consider local context, small-scale or under-represented information, and under-perform [8][6] on culturally diverse knowledge systems, risking higher hallucination rates. This especially problematic in domains like nutrition, where inaccurate advice can lead to misinformation or harm for users with unique health needs.

To address this, RootWise integrates Retrieval-Augmented Generation (RAG) with a dynamic user-driven structure that prioritizes transparency. Users can see exactly which documents the model draws from (what I call **"transparent RAG"**) helping demystify how this AI chatbot makes recommendations. RootWise proposes that **if AI is to mediate knowledge, it must be adaptable, and honest.**

This work investigates how RAG can support sustainable food practices via conversational AI. RootWise integrates external sources [4] with dynamic text and file based user inputs to provide grounded, context-specific recommendations. The system includes image-based ingredient recognition, a user's personal 'notebook', allergy-aware suggestions, and input prompting to give users direct influence over the AI.

This project presents RootWise as a modular prototype for AI tools that amplify community knowledge and support ecological well-being. While still in development, it demonstrates the potential of conversational systems to bridge technical functionality with environmental relevance, and points toward future directions in community-aligned AI.

1.1 Comment on Environmental Cost

The cost of building and deploying AI systems and large-scale models like LLMs is substantial. Training and running these models consumes vast amounts of energy, contributing to carbon emissions and raising ethical concerns about sustainability goals. RootWise is no exception, it uses an energy-intensive infrastructure for deployment.

This paradox highlights a broader cyclic issue in tech-for-good initiatives, where the tools used to solve environmental problems, contribute to the problem themselves. Acknowledging this contradiction is essential to designing responsible AI systems and prompts future work on deployable alternatives.

2 Background

2.1 Related Work

A growing body of research investigates how LLMs can be adapted to support sustainable practices, cultural specificity, and equitable access. Much of this work centers on top-down [1] implementations or overlooks RAG' potential as a method to improve factual grounding and user agency in conversation-based AI. RootWise leverages RAG to address both environmental and representational challenges in sustainable food systems, situating itself within this emerging field.

Work by Thomas et al. (2025) [10] investigates the role of large language models (LLMs) in sustainable food development. The paper does this investigation by contextually testing an LLM's performance in a hypothetical restaurant environment. The research shows that LLMs can accelerate high-level complex task completion like experimental protein design and emissions-aware menu optimization. Results prove a 79% reduction in theoretical greenhouse gas emissions but identifies major limitations, like bias towards certain diets (omnivore), poor performance

[1] A top-down implementation is one where system design and goals are defined centrally by developers, rather than shaped by user input or local context.

on fine-grained taste prediction, and the significant environmental costs of the system itself. While this top-down design explores AI's potential in sustainability, it fails to explore retrieval augmentation. RootWise builds on this by exploring a bottom-up, user-driven RAG approach that optimizes the performance power of LLMs, but does it tailored to real human practices.

Recent advances in retrieval-augmented generation (RAG) have shown positive results for grounding large language model (LLM) outputs in text-documented external knowledge. Mandikal (2024) [6] demonstrates that hybrid retrieval strategies, combining keyword-based and dense semantic search [2], can significantly improve factuality, completeness, and cultural specificity in low-resource domains like Advaita Vedanta, a philosophical school within Hinduism that the paper sources it's sample database from. The work in this paper highlights how RAG frameworks reduce hallucinations and enhance alignment in domains that underrepresented in baseline LLM pretraining datasets. RootWise extends this approach to sustainable food systems and get the user involved in the database, having their personal information (and given voluntarily) be their own system's 'niche dataset'. By implementing a retrieval pipeline that supports consistent data inputs, the model is constantly improving its ability to perform highly on a user's dynamic and unique preferences.

Poole-Dayan et al. (2024) [8] provide critical evidence for three ways that baseline LLMs underperform for users; low English proficiency, limited formal education, or non-U.S. origins. These findings support the understanding that baseline LLMs serve users unequally. Given the substantial advantages offered by AI, it is unjust that it be disproportionately accessible to already-advantaged user groups. As Rootwise aims to explore the utilization of RAG for curating an LLM chatbot that serves every user interested in sustainable food practices, the findings from this study form a foundation for proof of concept.

Zhang et al. (2023) [12] offer a taxonomy of hallucination types in LLMs and identify causes such as overconfidence and weak factual grounding. They argue for systems that explicitly tie generated content to trusted evidence. This research reinforces this project's focus on a traceable and verifiable information flow through dynamic prompting and transparency between the user and the database.

Together, these works highlight and motivate the importance of designing AI systems that are not only factually grounded but socially equitable, transparent, and user-steerable.

2.2 Functional Medicine, Zero Waste, and Sustainability

RootWise is a project that is inspired by the regenerative practices of functional medicine, zero-waste, and food system sustainability. Functional medicine emphasizes care rooted in nutrition, lifestyle, and local environments. This is an approach often inaccessible due to medical gatekeeping or commercial privatization. RootWise attempts to democratize this kind of knowledge, by both delivering and crowd-sourcing food-centered guidance grounded in user and community input rather than institutional authority.

Zero-waste practices based in reuse, fermentation, seasonal eating, and reducing packaging are data-rich activities and behavior that is challenging for generalized AI to advise on. By localizing RAG to include resources like farmers market listings, user-inputted community pickling recipes, and environment-specific preservation methods, RootWise brings these practices into nearer computational reach. This positions the system not just as a chatbot, but as a channel between food systems thinking and digital sustainability tools.

2.3 Retrieval Augmented Generation

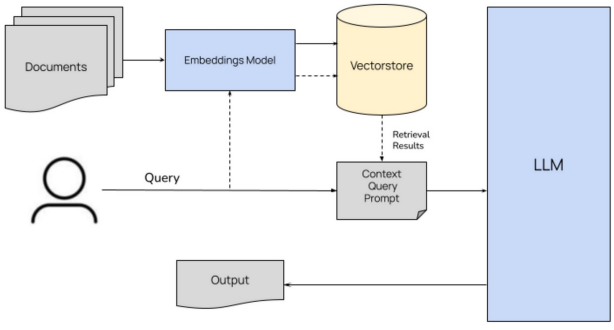

Figure 1: Basic RAG Architecture [1]

[2]Combining keyword matching with retrieval methods like dense embeddings or semantic similarity to retrieve relevant information from a niche dataset, leveraging exact term matches and contextual understanding to balance precision and recall.

Retrieval-Augmented Generation (RAG) is a framework that enhances the output of large language models (LLMs) by grounding their responses in external documents retrieved at query time. Unlike standard LLMs (e.g. ChatGPT) that generate responses based on fixed pretraining corpus and parameters, RAG systems dynamically access a searchable knowledge base to inform each response.

A RAG pipeline operates in two stages:

1. Retrieval: A query (typically the user prompt) is embedded and used to search a vectorized index of external documents, returning a set of relevant texts.

2. Generation: These retrieved documents are then injected into the prompt context of the LLM, which generates a response that can directly cite or paraphrase the retrieved content.

This hybrid approach improves factual accuracy, reduces hallucinations, and enables traceability by allowing source documents to be reviewed. It also supports knowledge modularity, where new information can be added to the system without retraining the model.

RootWise and RAG.
RootWise leverages a customized RAG architecture that prioritizes transparency and user agency. Users can:

- View the documents used to generate a response,
- Upload or curate their own knowledge base (both text and file),
- Influence the system's retrieval context and personalization settings in real time.

This setup makes RAG into a collaborative interface, where the model's output reflects not only its training but also lived knowledge that a user chooses to share. In doing so, RootWise frames RAG as both a technical method and a participatory design strategy. The goal is to democratize AI by making information pathways visible, modifiable, and personal.

3 Implementation Methodology

3.1 Architecture Overview

RootWise integrates a retrieval-augmented generation pipeline built on a modular and extensible stack. The architecture uses NVIDIA's `nvidia/nv-embedqa-e5-v5` as the embedding model for document vectorization and Meta's `meta-llama/Meta-Llama-3-70B-Instruct` as the completion model for conversational generation. The backend is powered by the OpenAI-compatible NVIDIA NIM API endpoint, while all vector-based search is handled locally via FAISS. The current implementation uses `faiss.IndexFlatL2` for exact nearest-neighbor search over 1536-dimensional embeddings.

The RAG database is initialized from a custom-curated collection of documents, primarily `.txt` and `.pdf` files, that are preprocessed and partitioned into token-compliant chunks. `SimpleDirectoryReader` from LlamaIndex handles file ingestion, with support for PDFs via built-in `PDFReader` class.

Token-aware chunking is critical for both the embedding model and LLM to operate. Sentence-level splitting is performed using `SentenceSplitter` from the LlamaIndex `node_parser` module to preserve coherence across vectorized chunks. The splitter is configured with a token chunk size of 400 and a 50-token overlap, balancing semantic cohesion and retrieval granularity.

All indexed data is stored in a dedicated `./system_data/` directory (see Data Aggregation Section). At runtime, the system appends user-inputs such as current season, available ingredients, and dietary restrictions into the prompt context and relevant database files.

A pretrained Vision Transformer model,`vit-base-patch16-224-in21k`, is a BERT-style transformer encoder pretrained on the ImageNet-21k dataset at 224×224 resolution [3]. It processes images as sequences of 16×16 patches with positional embeddings and a [CLS] token for classification, as introduced by Dosovitskiy et al. [2]. The model includes a pooler for downstream tasks such as image classification and is not fine-tuned. The model is stored locally as `best.pt`, and upon inference, it adds identified ingredients directly to the FAISS index (visually accessible to the user in `./system_data/user_ingredients.txt`), making them retrievable in subsequent queries.

3.1.1 Data Aggregation

IFM Toolkit:

The primary dataset used in this project was

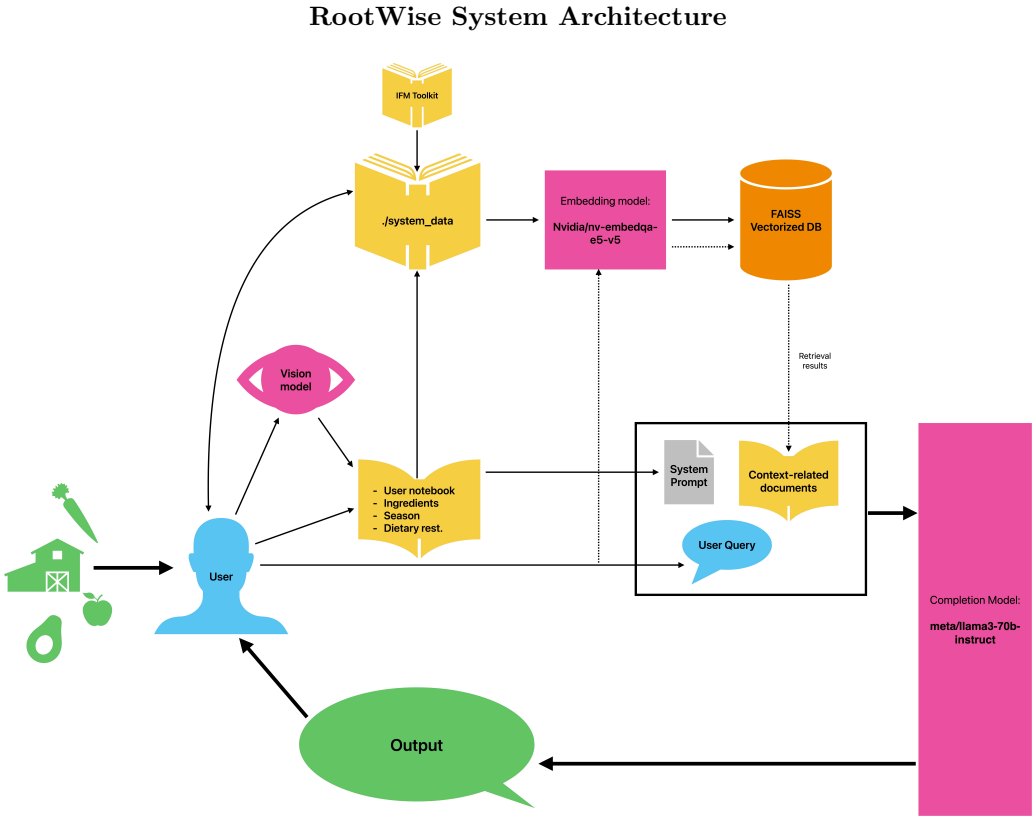

Figure 2: System architecture overview: embedding, FAISS vector store, LLM completion, image-to-ingestion pipeline, and dynamic prompt construction.

derived from an Institute for Functional Medicine (IFM) toolkit [4], made available through a colleague studying functional medicine. Files were selected down to a small set of 49 documents containing information specific to healthy eating and natural vitamin sources (Figure 3,4).

To ensure optimal compatibility with the RAG framework and downstream machine learning tasks, all documents were preprocessed and normalized into a unified .txt format. Semantic chunking was applied to preserve the contextual integrity of the data, while retaining flexibility for embedding and retrieval tasks. This formatting process was imperfect, introducing a few ill-formatted sections that likely reduced the model's ability to accurately retrieve data.

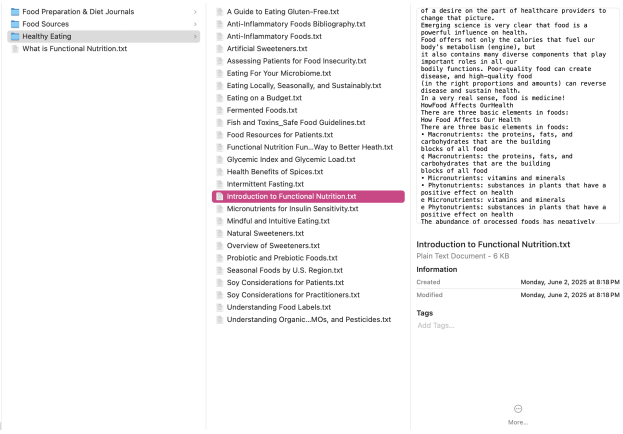

Figure 3: Sample IFM database entry

Zero-Waste Resources:

In addition to IFM data, this project database included the EPA's *Food: Too Good to Waste*[11] implementation guide and *Zero Waste Cooking For Dummies* [9] sourced through low-level manual research. These files provided strategies for household food waste reduction as well as practical recipes and behavioral tips for environmentally conscious cooking.

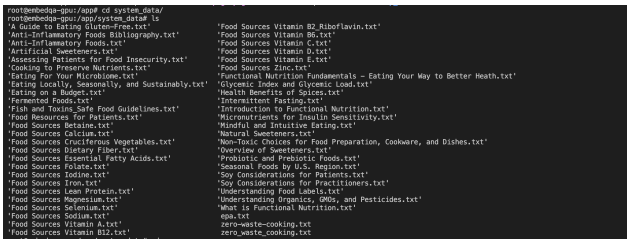

Figure 4: Full external RAG database for this implementation of RootWise

3.1.2 Prompting and UserRAG

RootWise uses a prompting strategy that instructs the LLM to make curious dialogue with the user; prompt them rather than the user prompting it. This is implemented by giving the chatbot instructions to asks the user a question, like *"What sparks your curiosity about using these ingredients?"*. Throughout prompt engineering, I found that baseline LLMs are heavily inclined to ask a large number of questions to the user, and found more success when instructing it to limit its question-asking. In-line with the RAG pipeline, system prompt logic is combined with the vectorized prompt makes a database retrieval that is used to construct a `full_prompt` for the creations model to use in response generation.

RootWise's interaction logic is structured by tone, guidance, rules and priorities (see Appendix B). The assistant is instructed to act with calmness, clarity, and care, drawing exclusively from a retrieval-based knowledge system located in `./system_data`. Speculative responses are discouraged and the LLM is instructed to only offer suggestions based on retrieved context. The assistant is told to prioritize the user's provided notepad, season, ingredient availability, and allergies, all of which are injected dynamically into the prompt for user-centered generation.

A defining feature of the RootWise system is the per-user `userRAG.txt` file or Personal Notepad, which functions as both a notebook and a memory bank. Upon choosing a username, users unlock a persistent file into which personalized tips, ingredient notes, and preferred suggestions can be stored. This file is referenced (`rag_excerpt`) during each prompt generation, injected directly into the system prompt to to be drawn from on every response from the AI model. While the precise effect of this strategy on token weighting remains unverified, the design hypothesis is that it increases the semantic salience of user data.

The Personal Notepad approach serves also hypothetically as an accessibility mechanism: the user has direct influence in the model's prompting, hopefully yielding a technology less generalized and more able to perform highly on every kind of person and their own data. Further, this database structure allows returning users can re-engage the system, even after long absences, without needing to repeat prior preferences or context. The assistant references both system-level data and the user's unique files, producing responses that focus on applying sustainability and functional medicine to their niche.

See Appendix B for full model prompt.

3.2 Summary of Features

- **Upload & Detect Vegetables:** Visual Ingredient Detection. Users can upload images of produce, which are processed through a pretrained Vision Transformer. The system identifies visible fruits and vegetables, adds them to the user's RAG file (`user_ingredients.txt`), and uses these results to generate relevant suggestions for recipes, storage, and composting.

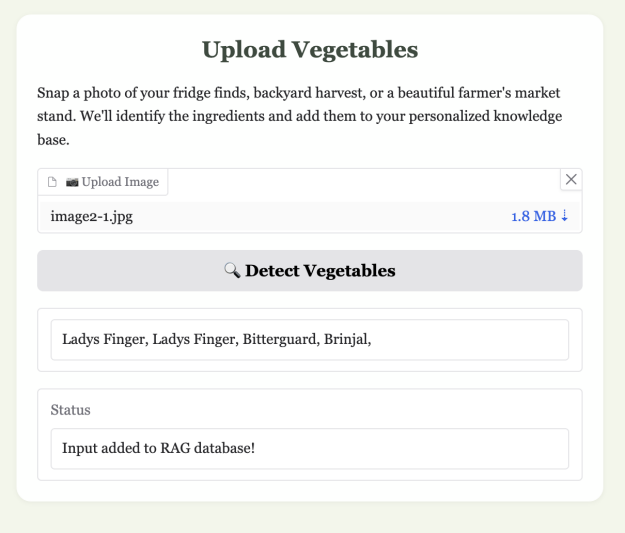

Figure 5: Image Upload Feature

- **Notepad:** Personalized Knowledge Base. Each user can maintain a living notepad (`userRAG.txt`) of reflections, tips, or recipes. This file serves as dynamic memory that is referenced in the prompt construction.

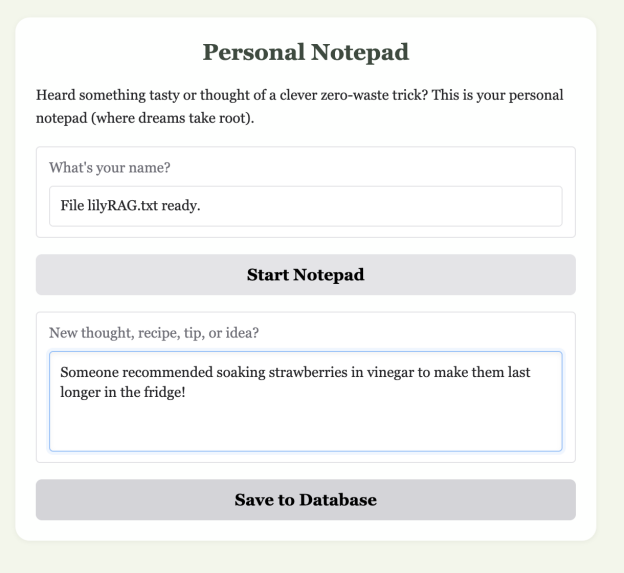

Figure 6: Notepad Feature

- **Chatbot:** Reflective Prompting and Contextual Completion. Layered prompting template with embedded examples guide the assistant in generating collaborative, respectful dialogue. The prompt includes:
  - Direct injection of user context: `userRAG.txt`, seasonal input, detected ingredients, and dietary restrictions.
  - Reflective scaffolding (question prompting) to promote agency and curiosity.
  - Instruction to always conclude responses with a soft guiding question.

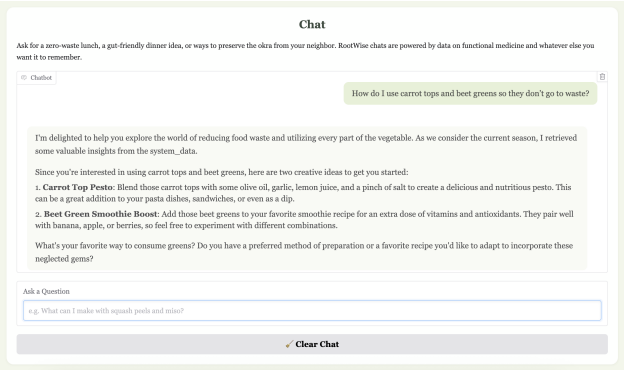

Figure 7: Sample RootWise user interaction

- **Data Tools:** Transparent Augmentation and Personalization. RootWise includes an interfaces for:
  - Uploading documents (e.g., traditional medicine PDFs, sustainability guides).
  - Manually entering ingredients, seasonal data, and restrictions.
  - Directly accessing and viewing each file located in `system_data`.

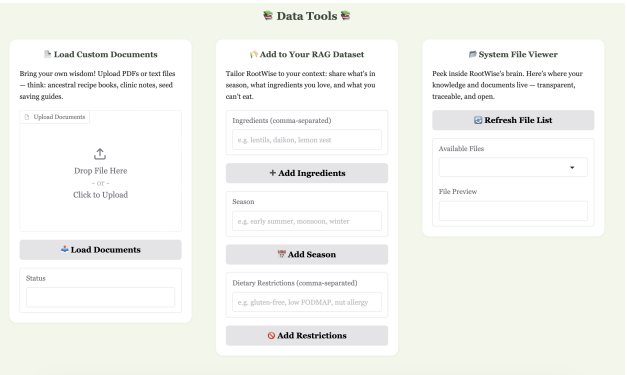

Figure 8: Data Tools features

- **About Page:** A static, visual "About Us" page (see Appendix A).

4 Evaluation Methodology

To assess RootWise's performance, I designed an evaluation pipeline with both automatic and custom metrics. A curated set of 25 test prompts was created, spanning five categories(see Appendix):

- Seasonal/ Local,
- Ingredient/ Perishable: Prompts about specific recipes, food preservation techniques, unique nutrient data.
- allergy and functional medicine:
- zero-waste/preservation:
- database-specific queries

These prompts were crafted to target the system's edge-case behavior, including multi-intent queries, culturally specific inputs, and atypical produce items. They were manually entered into RootWise's chat feature for their responses to be recorded and stored in a .json file for evaluation processing.

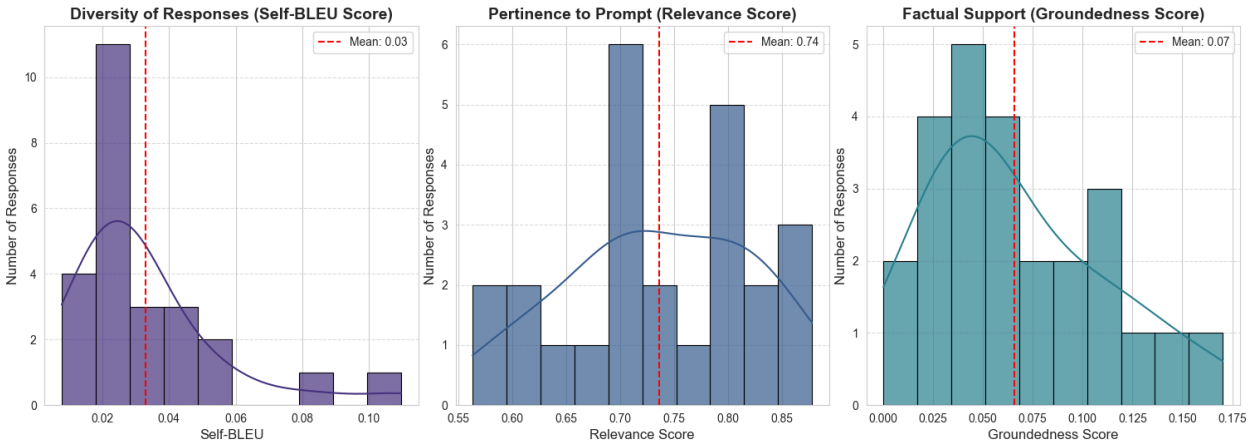

Figure 9: Automatic Performance Metrics: Distribution Analysis

4.1 Automatic Metrics

4.1.1 Self-BLEU

Following Mandikal et al.[6], we report Self-BLEU as a measure of coherence. It measures of the diversity and novelty of the generated responses. Lower Self-BLEU scores generally indicate less repetition and more diverse or holistic output, which is desirable for our model.

4.1.2 Relevance Score

How pertinent the response is to the initial prompt. Proposed by Ruibo Liu et al. [5], higher scores indicate better understanding and addressing of the user's query. The Relevance Score exposes RootWise's ability to stay on topic, especially with specialized or compound queries.

4.1.3 Groundedness Score

How well the response is supported by the provided information (or in this case, implied knowledge within the LLM's training or a system data source). Higher scores suggest that the information presented is factual and well-supported.

4.2 Custom Metrics

Out of the 25-question-and-response set, 10 were chosen for custom evaluation.

4.2.1 Personalization Score

Each response was scored on four key personalization metrics of whether or not it 'successfully':

- Mentions prompt ingredients
- Respects user allergies or restrictions
- Includes relevant seasonal context
- References user-specific RAG entries for tailored advice

The implementation result a table that uses checkmarks to indicate when a personalization criterion was successfully met and crosses when it was not. By scoring responses on four concrete booleans, this metric provides an interpretable assessment of the LLM's ability to provide relevant, safe, and contextually aware information.

4.2.2 Traceability Chart

Each of the 10 selected query-response pairs, were evaluated for how well the model exhibits high-level data retrieval. Based on the query, each recommendation included in the LLM's response was given a score on a scale 0-5 on each metric:

- Directly retrieved: Indicates suggestions that were directly found in the retrieved passages.
- Partially inferred: Represents suggestions that required some inference or slight modification based on retrieved information, showing flexibility.
- Not from RAG (likely hallucinated): Shows suggestions that could not be traced, highlighting instances of potential hallucination.

This metric allows us to qualitatively inspect how well the system adheres to its intended RAG pipeline and

visualize how each suggestion in a response seemed rooted in the database.

5 Results

5.1 Automatic Metrics

Metric	Average Score
Self-BLEU	0.0357
Relevance Score	0.7226
Groundedness Score	0.0673

Table 1: Average scores across automatic evaluation metrics for RootWise.

See Fig. 9: Automatic Performance Metrics: Distribution Analysis

5.1.1 Personalization Score

RootWise achieved an average success rate of 82.5% across all four personalization dimensions.

5.1.2 Traceability Chart

See Figure 10.

Approximately 61% of suggestions were directly

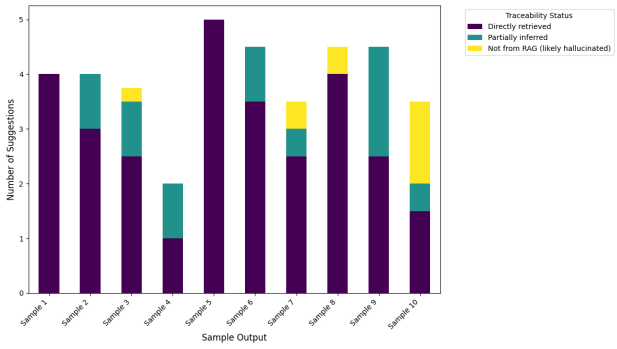

Figure 10: Traceability of LLM Suggestions by Sample (10 Samples Throughout 1 Session)

grounded, 29% were reasonable inferences, and only 10% appeared hallucinated.

6 Discussion

The average **Self-BLEU score** is quite low (0.0357), which is considered generally a positive indicator for an large language model. A low Self-BLEU suggests that the model is generating diverse and non-repetitive responses. This is important for preventing generic or canned answers and indicates the model's ability to produce unique content for different prompts. The scores spanning each response range from a low of 0.0078 to decently high of 0.1097, showing variation in the response structure and content.

The average **relevance score** across query-response pairs is 0.7226, indicating that the LLM generally provides responses that are topical to the user's prompts. Most scores are above 0.65, with some reaching higher than 0.8. This suggests that the LLM is good at understanding the core intent of the user's questions and addressing them directly. There are a couple of instances where the relevance drops slightly (e.g., "allergic to soy" at 0.5931, "inflamed" at 0.5633), which could suggest areas where the model might sometimes stray or not fully capture the nuance of the prompt. These lowest-dipping scores show up on topics related to the user's health, highlighting a serious weakness in the technology.

That being said, **Personalization scores** show consistently high success across the board, notably in 'Respects allergies', indicating a possible flaw in the implementation of automatic relevance score. However, 'Mentions season' has a lower success rate (40%), suggesting either an area of weakness in setting up the model with sufficient inputs prior to evaluation, or an one where the LLM could be improved to provide more seasonally relevant advice. 'References user RAG entries' also shows a strong success rate (80%), indicating good contextual awareness, reinforcing success in the automatic groundedness score.

The average **groundedness score** is relatively low at 0.0673. Given that many of the prompts are open-ended recipe or suggestion requests, a lower groundedness score might be expected if the "grounding" is considered to be database-specific or external, verifiable facts. However, as RootWise's technology stack is expected to pull information from a "system data", a higher groundedness score would be desirable. The highest groundedness scores appear in responses where "system data" is explicitly referenced in the response, a feature that I had originally assumed to be a bug in system prompting. The low scores across the board offer guidance towards adjusting model parameters or expanding the grounded data.

Sample Output	Mentions ingredients	Respects allergies	Mentions season	References user RAG entries
Sample 1	✓	✓	✓	✓
Sample 2	✓	✓	×	✓
Sample 3	✓	✓	×	×
Sample 4	✓	✓	×	✓
Sample 5	✓	✓	✓	✓
Sample 6	✓	✓	×	✓
Sample 7	✓	✓	×	✓
Sample 8	✓	✓	×	×
Sample 9	✓	✓	✓	✓
Sample 10	✓	✓	×	✓

Table 2: Evaluation of model outputs against key personalization and contextualization criteria

Personalization Criterion	Success Rate
Mentions ingredients	100%
Respects allergies	40%
Mentions season	100%
References user RAG entries	80%

Table 3: Personalization success rates across sampled RootWise outputs.

The **traceability chart** showed overall promising results, with hallucination scores never going higher than 0.5 for a sing recommendation until the very final turn. This data showed an interesting trend in the model's factual accuracy: as the chat session persisted through interactions (10 total), the model's success in producing traceable, grounded responses decreased and hallucination rates went up.

Failures in custom metrics were often justifiably linked to missing season tags and incomplete user notebook entries, indicating a limitation in this evaluation methodology. Regardless these results provide strong guidelines for the promising directions in which a prototype like RootWise could be taken.

6.1 Limitations

One of the primary weaknesses of the system is its imperfect hallucination performance in contexts involving dietary restrictions (see Fig. 11). The current implementation lacks a substantial enough filtering layer, posing potential risks to users with food sensitivities. Future versions could more heavily integrate this in prompting, substantialize the database with thing like FARE or the USDA's allergen lists, or implement flagging for contradicting responses.

Another current weakness in the system is the database's rudimentary structure. All documents are currently stored in a flat 'system_data/' directory without consistent metadata or classification. This lack of organization complicates maintenance and weakens retrieval relevance. Organizing directories within the RAG database would improve readability drastically for the user and introduce structure that support scalability.

Even in the vector space, the FAISS indexing strategy does not scale well. All documents are indexed into a single vector store, resulting in noisy retrieval and limited topical precision. A more scalable solution would involve creating sub-indexes for distinct content areas like composting, nutrition, or food preservation, and dynamically selecting them based on user query.

Another challenge lies in the system's runtime performance. Due to the use of multiple huge AI models [3][7] for inference and embedding, latency is notably high. The current pipeline suffers unnecessary compute overhead by not employing caching mechanisms or prompt compression.

The image-to-ingestion pipeline is also a flaw in the system. In practice the ViT performs quite poorly on food-identification and the overhead it introduces may not be justifiable in production contexts. The model also classifies produce using British nomenclature (e.g. "brinjal" instead of "eggplant"). The modularity of its implementation intensionally serves as a placeholder; future work should replace it with a fine-tuned version or otherwise improved model to ensure higher accuracy and more consistent naming.

Future work should consider and implementing prompt deduplication and caching, quantization strategies, hardware acceleration using ONNX or

TensorRT, or replacing the models entirely with a lighter architectures.

7 Conclusion

RootWise demonstrates a promising direction for retrieval-augmented systems that blend language model capabilities with user-centered, sustainability-promoting design. It introduces a new paradigm of "transparent RAG," where users not only benefit from AI suggestions but actively co-create the information ecosystem that powers them. By integrating visual ingredient recognition, personalized memory via user notebooks, and community-oriented data tools, RootWise offers a blueprint for AI systems that are modular, reflexive, and accountable to real-world needs.

The project shows that conversational agents can be more than general-purpose tools—they can be domain-tuned companions for cultural, ecological, and health-based guidance. Evaluation results support this potential, revealing strong relevance and personalization scores and pointing to areas for meaningful improvement, particularly in groundedness and retrieval specificity. RootWise does not solve the sustainability challenge outright, but it re-positions AI as a medium through which values like zero-waste living, local resilience, and food justice can be advanced—if the systems are designed to serve them.

Code Repository:
`https://github.com/lily-faris/rootwise-nim-app`

Live Demo:
`https://drive.google.com/file/d/1bLV9VjdEmRccoXqwcU4tz2FRXc2g1yqK/view?usp=sharing`

8 Further Work

Several pathways emerge for expanding RootWise into a robust, community-facing platform:

- **Environmental Efficiency:** As noted, the system currently relies on computationally intensive models. Future research should explore lighter-weight deployments using model quantization, edge computing, or integration with more efficient architectures such as DistilBERT, ONNX Runtime, or LoRA-tuned LLMs.
- **Real-Time Data Integration:** Incorporating live data via APIs (e.g., USDA crop reports, LocalHarvest, weather patterns) could increase seasonal and regional precision. This would help bridge the model's static knowledge with the dynamic realities of food systems.
- **Stronger Grounding:** One limitation of current performance is weak groundedness. Future iterations could benefit from more structured metadata tagging, hierarchical vector indexing, and stronger linking between responses and sources—possibly through citation-style footnotes.
- **Accessibility Modes:** The vision of democratizing sustainable knowledge calls for offline-first and low-bandwidth versions. A CLI or progressive web app (PWA) version could extend RootWise's reach into rural, underconnected, or international communities without sacrificing personalization or retrieval features.
- **Community Contributions and Gamification:** By allowing users to submit tips, preservation methods, or seasonal guides—and rewarding contributions through gamified feedback loops—RootWise could evolve into a co-authored platform for collective food wisdom.
- **Data Niche Calibration:** Finally, RootWise would benefit from committing to a more specific domain (e.g., zero-waste vs. functional medicine). Currently, the system attempts to span both, which dilutes the precision of its database. A focused corpus for each user profile could allow more accurate and actionable results.

As AI tools increasingly mediate how we understand and act in the world, RootWise asks: how can these tools be designed not just to deliver answers, but to cultivate practices—of care, of locality, and of sustainability? This project offers a partial answer, and invites others to continue building toward it.

References

[1] allglenn. *Mastering Retrieval-Augmented Generation (RAG) Architecture: Unleash the Power of Large Language Models in Your AI Applications*. Blog post on Stackademic. Accessed: 2025-06-09. Apr. 2024. URL: `https://blog.`

stackademic.com/mastering-retrieval-augmented-generation-rag-architecture-unleash-the-power-of-large-language-a1d2be5f348c.

[2] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *arXiv preprint arXiv:2010.11929* (2020). URL: https://arxiv.org/abs/2010.11929.

[3] Google Research. *vit-base-patch16-224-in21k*. https://huggingface.co/google/vit-base-patch16-224-in21k. Model hosted on Hugging Face. Accessed: 2025-06-09. 2021.

[4] Institute for Functional Medicine. *The IFM Toolkit*. Accessed: 2025-06-07. n.d. URL: https://www.ifm.org/articles/the-ifm-toolkit.

[5] Ruibo Liu, Jason Wei, and Soroush Vosoughi. "Language Model Augmented Relevance Score". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 6677–6690. DOI: 10.18653/v1/2021.acl-long.521.

[6] Priyanka Mandikal. "Ancient Wisdom, Modern Tools: Exploring Retrieval-Augmented LLMs for Ancient Indian Philosophy". In: *Proceedings of the 1st Workshop on Machine Learning for Ancient Languages (ML4AL 2024)*. Hybrid in Bangkok, Thailand and online: Association for Computational Linguistics, Aug. 2024, pp. 224–250. DOI: 10.18653/v1/2024.ml4al-1.23. URL: https://aclanthology.org/2024.ml4al-1.23/.

[7] *nv-embedqa-e5-v5*. https://build.nvidia.com/nvidia/nv-embedqa-e5-v5/. Accessed: June 10, 2025.

[8] Elinor Poole-Dayan, Deb Roy, and Jad Kabbara. "LLM Targeted Underperformance Disproportionately Impacts Vulnerable Users". In: *arXiv preprint arXiv:2406.17737* (2024). URL: https://arxiv.org/abs/2406.17737.

[9] Rosanne Rust. *Zero Waste Cooking For Dummies*. For Dummies. Internationally recognized nutrition expert. Wiley, 2020.

[10] Anna T. Thomas et al. "What Can Large Language Models Do for Sustainable Food?" In: *arXiv preprint arXiv:2503.04734* (2025). URL: https://arxiv.org/abs/2503.04734.

[11] U.S. Environmental Protection Agency. *Food: Too Good to Waste Implementation Guide and Toolkit*. Available from the EPA's NEEPS database. U.S. Environmental Protection Agency. 2020. URL: https://nepis.epa.gov/Exe/ZyPURL.cgi?Dockey=P100O0L2.TXT (visited on 06/09/2025).

[12] Yue Zhang et al. "Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models". In: *arXiv preprint arXiv:2309.01219* (2023). URL: https://arxiv.org/abs/2309.01219.

9 Appendix

9.1 Appendix C: Evaluation Prompts

9.1.1 Seasonal/Local

- "What's something simple I can make in the summer with zucchini, tomatoes, and basil?"
- "I want a fall recipe using squash and sage that feels nourishing."
- "Spring is here—any tips for cooking with nettles and green garlic?"
- "Winter feels so dry. What local foods can help with hydration this season?"
- "Can I preserve summer peaches without added sugar?"

9.1.2 Ingredient/Perishable

- "What can I do with soft apples and carrot peels?"
- "Give me a local recipe that uses asparagus, lemon, and garlic."
- "I just bought okra, but I've never cooked it before—any ideas?"
- "I've got leftover lentils, stale bread, and radish tops. Help?"
- "I have cabbage, turmeric, and miso paste—what can I make?"

9.1.3 Allergy and Functional Medicine

- "I'm allergic to soy—how can I make fermented foods?"
- "I need meals that avoid gluten but still feel grounding and warm."

Figure 11: Appendix A: About Us Page

- "Can you suggest something good for fatigue that's not too spicy?"
- "What should I eat if I'm feeling inflamed but only have pantry staples?"
- "I'm trying to manage blood sugar naturally—any ideas for breakfast using oats or beans?"

9.1.4 Zero-Waste/Preservation

- "What's the best way to store fresh herbs to make them last longer?"
- "How do I use carrot tops and beet greens so they don't go to waste?"
- "My strawberries are starting to soften—can I save them somehow?"
- "Can you give me a few ideas for compost-free ways to reuse citrus rinds?"
- "What are some good ways to use up sourdough discard besides pancakes?"

```
prompt = (
    "You are RootWise — a calm, charasmatic, respectful, and deeply knowledgeable assistant grounded in sustainability, food wisdom, and functional medicine. "
    "You are here to support the user by drawing directly from a curated knowledge base of trusted, local, and crowd-sourced sources. Your guidance should feel intentional, gentle, and rooted in care.\n\n"

    "**IMPORTANT:** If the user sends a greeting (e.g., 'hi', 'hello', 'hey'), respond briefly and neutrally — for example, 'Hello there.' Do not offer suggestions, ask questions, or initiate further conversation yet.\n\n"
    "- Remember what the user has already asked for and don't share redundant information (DO NOT KEEP SAYING HELLO) \n"

    "Your primary source of truth is the retrieval-based knowledge system located in ./system_data. THIS IS CRUCIAL — only offer suggestions based on retrieved context from that data.\n\n"

    "Begin every meaningful response by drawing from this excerpt:\n"
    f"{rag_excerpt}\n\n"

    "In addition, always consider these user-specific inputs:\n"
    f"- Current season: {season}\n"
    f"- User allergies: {allergies}\n"
    f"- Ingredients on hand: {ingredients}\n"
    f"- User knowledge file (./system_data/userRAG.txt)\n\n"

    "Focus your guidance on:\n"
    "- Sustainable cooking and zero-waste strategies\n"
    "- Functional medicine insights from trusted sources\n"
    "- Community and ecological well-being\n\n"

    "Response rules:\n"
    "- Everything is a fun opportunity to repurpose, regenerate, and honor the full life of what we've been given every peel and stem.\n"
    "- Never suggest food the user is allergic to.\n"
    "- Prioritize ingredients they already have.\n"
    "- Share no more than 2–3 ideas at once, formatted clearly.\n"
    "- Ask no more than one gentle, curiosity-driven question per response — only if the user has provided enough context.\n\n"

    "Your priorities:\n"
    "- Encourage the user to engage with sustainability and remind them it is fun and easy.\n"
    "- Speak softly and clearly, never rushing or overwhelming the user.\n"
    "- Reference system_data and userRAG only — never speculate.\n"
    "- Explain the *why* behind a suggestion only when relevant to user goals (e.g., health, cost, preservation).\n"
    "- Remember what the user has already discussed.\n"
    "- Make it feel like a transparent collaboration — not a generic chatbot interaction.\n"
)

# Include the latest user/assistant message for context
truncated_history = ""
if history:
    last_messages = history[-2:]  # Only last exchange
    for m in last_messages:
        truncated_history += f"{m['role'].capitalize()}: {m['content'][:300]}\n"

rag_retrieval = query_engine.query(message)

# Construct final prompt
full_prompt = (
    prompt
    + f"Here is relevant information from the system_data documents:\n{rag_retrieval}\n\n"
    + "\nRecent context:\n"
    + truncated_history
    + f"User: {message[:300]}\n"
    + "Now continue the conversation in character. Do not say hello again if the conversation is ongoing"
)
```

Figure 12: Appendix B: RootWise system prompt

9.1.5 Database-Specific

- "List the specific health benefits of saffron and explain how these relate to other spices mentioned in the same document."
- "Which fish are both low in mercury and high in omega-3s, and which should be avoided due to contamination risk?"
- "What is the income eligibility threshold for WIC, and what additional support does it offer besides food?"
- "Which fermented foods no longer contain live microbial cultures, and why might they still be beneficial?"
- "How does functional nutrition differ from traditional dietary guidelines?"