VOICE-ENABLED AI RECIPE GENERATOR BASED ON AVAILABLE INGREDIENTS

Dr.Jim Mathew Philip
Associate Professor, Computer Science and Engineering
Sri Ramakrishna Institute of Technology
Coimbatore,India
jimmathewphilip.cse@sritcbe.ac.in

S.Akashraj
Computer Science and Engineering
Sri Ramakrishna Institute of Technology
Coimbatore,India
akashraj.2202006@sritcbe.ac.in

C.Akil
Computer Science and Engineering
Sri Ramakrishna Institute of Technology
Coimbatore,India
akil.2202007@sritcbe.ac.in

V.Inbanathan
Computer Science and Engineering
Sri Ramakrishna Institute of Technology
Coimbatore,India
inbanathan.2202056@sritcbe.ac.in

S.Hariharan
Computer Science and Engineering
Sri Ramakrishna Institute of Technology
Coimbatore,India
hariharan.2202047@sritcbe.ac.in

Abstract The project Voice-Enabled AI Recipe Generator Based on Available Ingredients focuses on creating an intelligent system that automatically generates cooking recipes from user-given ingredients. It uses advanced Transformer-based language models like Gemini or GPT to produce natural, step-by-step recipe instructions. The assistant supports both voice input and output, allowing users to interact hands-free while cooking. A user-friendly Streamlit interface is developed for smooth interaction and accessibility. The system validates ingredients using AI before processing them for recipe generation. It also offers random recipe suggestions to inspire users. The generated output includes the recipe name, required ingredients, preparation steps, and a cooking tip. The project demonstrates how AI and NLP can make cooking more interactive, efficient, and personalized. By fine-tuning on datasets like RecipeNLG, the system's recipe quality and creativity can be further enhanced. Overall, it showcases the power of modern AI models in real-world text generation applications.

Introduction

To create recipes, AI systems must produce structured and logical instructions that follow cooking rules. Majumder et al. found that neural language models can generate plausible recipes when trained on large recipe databases in their study on computational recipe generation. However, generic language models sometimes produce unsafe or unfeasible food combinations because of a lack of domain knowledge.

Gemini is an excellent general-purpose language model with strong reasoning capabilities. Given the right cues, Gemini can produce recipes that make sense, suggest the best cooking methods, calculate cooking times, and offer culturally diverse advice on cooking.Teaching the model with a variety of online text formats includes a substantial amount of recipe and cooking content, which enables it to understand ingredient combinations, cooking techniques, and culinary terminologies.

Prompt engineering is essential for effective recipe generation. Structured prompts that specify the desired output format, ingredient limitations, time constraints, and instruction style produce more insightful results than do open-ended prompts. To maximize responsiveness and generation speed, Gemini 2.5 Flash features a thinking mode with programmable thinking budgets. This is essential for interactive application.

Understanding cultural culinary traditions and the chemical properties of flavor compounds is essential for accurately determining which ingredients work best with each other. In their flavor network analysis, Ahn et al. observed that in East Asian climates, culinary traditions often combine contrasting flavor approaches, whereas Western cuisines prefer an open- blended approach among ingredients sharing flavor compounds. Certainly, AI models have been able to learn through imprinted training data some combinations that could not be explained rationally.

Research Background

Most cooking apps and online recipe databases still require users to use trial and error to find recipes that use ingredients they already have. Artificial intelligence (AI), particularly generative AI and natural language processing, has enabled the provision of personalized cooking assistance in novel ways. In addition to mixing ingredients and creating situation- appropriate recipes, Gemini and other large language models can clearly and specifically instruct users on how to cook.

Despite the advent of online recipe databases and cooking apps in the digital age, most still require users to use trial and error to find recipes that work with the ingredients they already have. The advancement of artificial intelligence, particularly generative AI and natural language processing, has expanded the options available for customized cooking assistance applications. Large language models, such as Google's Gemini, can understand ingredient combinations, generate context-relevant recipes, and provide accurate cooking instructions based on user preferences.

Voice technology has also altered the way people interact with digital assistants in the kitchen. Text-to-speech features enable hands-free cooking instructions, allowing users to follow recipes while their hands are busy preparing food. An interactive cooking assistant that can adapt to the available ingredients and provide audio instructions while cooking is created by fusing voice synthesis and AI recipe generation.

Proposed System

AI-powered recipe generation uses natural language processing in cooking apps. It's important to understand how ingredients work together, the different cooking methods, and how to organize recipes.

Natural Language Generation for Recipes

To create recipes, AI systems must produce structured and logical instructions that follow cooking rules. Majumder et al. found that neural language models can generate plausible recipes when trained on large recipe databases in their study on computational recipe generation. However, generic language models sometimes produce unsafe or unfeasible food combinations because of a lack of domain knowledge.

Gemini is an excellent general-purpose language model with strong reasoning capabilities. Given the right cues, Gemini can produce recipes that make sense, suggest the best cooking methods, calculate cooking times, and offer culturally diverse advice on cooking.

Prompt engineering is essential for effective recipe generation. Structured prompts that specify the desired output format, ingredient limitations, time constraints, and instruction style produce more insightful results than do open-ended prompts. To maximize responsiveness and generation speed, Gemini 2.5 Flash features a thinking mode with programmable thinking budgets. This is essential for interactive application.

Ingredient Compatibility and Culinary Knowledge

Understanding cultural culinary traditions and the chemical properties of flavor compounds is essential for accurately determining which ingredients work best with each other. In their flavor network analysis, Ahn et al. observed that in East Asian climates, culinary traditions often combine contrasting flavor approaches, whereas Western cuisines prefer an open- blended approach among ingredients sharing flavor compounds. Certainly, AI models have been able to learn through imprinted training data some combinations that could not be explained rationally.

Text-to-Speech Technology in Cooking Applications

Ever since voice synthesis technology has started to take practical shape away from robotic- sounding speech to dynamic sound, this has evolved to enable several applications in hands- free environments such as the kitchen.

Evolution of TTS Systems: Earlier TTS systems relied on concatenative synthesis, where pre- recorded segments of speech were pieced together to form complete sentences. Modern neural TTS systems, such as WaveNet and Tacotron, generate speech from text based on deep learning and produce speech with natural prosody, intonation, and rhythm. Google Text-to-Speech API provides a good solution for TTS through an accessible API, supported for many languages and voices.

Existing System:

Although multiple cooking applications and kitchen assistants powered by AI exist, most have limitations that this project seeks to address.

Limitation of Existing System:

Classic Recipe Platforms: All Recipes and Food Network have extensive databases of recipes with user ratings and reviews, but first, you need to know what to cook. Supercook and MyFridgeFood take an input list of ingredients but search their database instead of generating anything using AI. Google Assistant and Amazon Alexa can read recipes aloud but cannot generate recipes based on what an ingredient is supposed to be. Once again, Tasty and other video recipe apps prove useful in their visual representation of what to do, but they do not personalize based on which specific ingredients are available.

AI Cooking Assistants: Some experimental systems focus on AI recipes. Chef Watson by IBM was an AI cooking assistant used in partnership with cognitive computing to come up with new ingredient combinations. The treatment was eventually discontinued. Neural recipe generation has been explored in some research projects; however, it remains an academic prototype and has never been applied in practice. More recently, commercial AI assistants, such as ChatGPT, can generate recipes when asked; however, they are not specialized cooking interfaces with ingredient selection.

Research Gaps: Most existing systems separate recipe discovery from instruction delivery, requiring users to switch between applications or devices. Few studies have integrated AI generation and voice synthesis into a single, user-friendly interface. This situation has led to a lack of ingredient-first recipe platforms that users would want to use to minimize food wastage by cooking with available ingredients.

These gaps will be addressed through an integrated solution of ingredient input, customized recipe generation using advanced AI, and voice instructions through an easy-to- access web interface. The combination of Gemini's language capabilities with gTTS voice synthesis in a Streamlit application creates a suitable cooking assistant for practical kitchen use.

No systems are widely available that seamlessly integrate intuitive ingredient selection, AI-generated recipes, and voice instructions in a simple and deployable application.

Methodology

System Architecture and Design

The AI Recipe Cooking Helper implements a client-server architecture integrated with AI and voice synthesis services.

Application Architecture: There system has three layers. The presentation layer uses Streamlit to render web interfaces with checkbox selections for ingredients, dropdown boxes for preferences, buttons for triggering the recipe generation, and display areas for dynamic content. In the processing layer, the Python backend logic orchestrates the collection of user input, manages communication with the API, processes and formats the response, and generates and plays audio files. The service layer includes integration with external APIs, specifically Google Gemini for recipe generation and Google Text-to-Speech for voice synthesis.

The architecture follows a sequential processing model for the collection of user input, which leads to AI recipe generation, which in turn goes on to TTS conversion and final audio playback. This separation allows each stage to be optimized and for graceful degradation should any service fail

Workflow: The user interaction workflow begins with ingredient selection through category- organized checkboxes, cooking style preference specification using a dropdown menu, and enabling optional voice instructions. Upon pressing the recipe generation button, the system checks whether enough ingredients are selected, builds an optimized prompt for Gemini, calls the Gemini API, displays the generated recipe, converts the recipe text to speech (if enabled), and plays the audio automatically, supplying manual controls as well.

Ingredient Selection Interface

The ingredient selection system organizes the options into logical categories for intuitive browsing and selection.Grains include rice, pasta, bread, quinoa, and noodles, which serve as carbohydrate staples. Dairy products include cheese, milk, butter, cream, and yogurt, which add richness and moisture. The set of spices is meant to enhance flavors with salt, pepper, cumin, paprika, oregano, and basil.

This standardization follows food pyramid structures and grocery store organization, which are meant to work intuitively for the user. The selection balances depth (variety of ingredient types) with focus (choosing not to overwhelm with options). Each category contained five to eight items, consistent with cognitive research pertaining to optimal choice set size.

Gemini API Integration

This app connects the Google GenAI Python client library to use the API key to authenticate Gemini. It uses the fast-gemini-2.5-flash model, which is optimized for speed while retaining good quality. The settings are thought config with the thinking_budget set to 0 to ensure immediate responses without long reasoning, which makes it the best for interactive applications that value speed. The API calls Jo in the format expected by Gemini. It wraps the prompt in a content array. The response object can then be accessed through the text attribute, where the generated text should be found. Exception handling should trap API failures and provide user message feedback when the generation fails.

To create a recipe using Gemini that results in viable and usable recipes, prompts are constructed very carefully. There are several factors that need to be considered in prompt construction. Ingredient Specification: The prompt begins with the enumeration of the chosen ingredients in the form of a comma-separated string. Listing them in this specific manner makes it clear that Gemini must consider them.

Results and Discussions

Implementation and Testing

The AI Recipe Cooking Helper was successfully implemented and tested in various scenarios to assess its functionality and usability.

Technology Stack: The application uses Streamlit 1.28+ as the web framework, google- genai library to integrate Gemini AI, gTTS (Google Text-to-Speech) for voice synthesis processes, tempfile for the temporary placement of audio files, and base64 encoding for embedding audio. The lightweight stack has minimal dependencies, ensuring easier deployment and maintenance.

Quality of Generated Recipes

Testing was done to evaluate the quality and appropriateness of AI-generated recipes across various ingredient combinations and cooking styles.

Recipe Diversity and Creativity

The system was tested with 50 different ingredient combinations across multiple cuisines and cooking styles. For every combination, Gemini identified and generated a peculiar recipe rather than applying a generic one. From the recipe diversity analysis, Italian-style dishes appear to arise from the combinations of pasta, tomatoes, and cheese; Asian-inspired recipes arise from rice, soy-based proteins, and vegetables; comfort foods arise from potatoes, dairy, and proteins; and healthy bowls arise from quinoa, vegetables, and lean proteins.

Recipe Practicality and Safety

Generated recipes were evaluated for practicality and food safety. The analysis showed that 92% of all generated recipes were immediately usable with no changes. About 6% needed a few extra clarifications on cooking times or temper at Generated recipes were checked for practical achievability and food safety. The analysis found that 92% of the recipes were immediately usable without changes. Six percent needed a few extra details on cooking times or temperatures. Two percent included unusual techniques that would require some adjustments. A positive note is that none of the recipes suggested unsafe ingredient.

The recommendations for reasonable cooking times fit well with preferred styles. Quick recipes generally took about 15 to 20 minutes. Medium recipes took around 30 to 45 minutes, while more complex suggestions took over 60 minutes with more complicated techniques. This also demonstrated Gemini's ability to understand cooking time constraints, ures. Meanwhile, 2% suggested uncommon techniques that would require some adaptation. A positive finding was that none of the recipes proposed unsafe ingredient combinations or risky cooking methods.

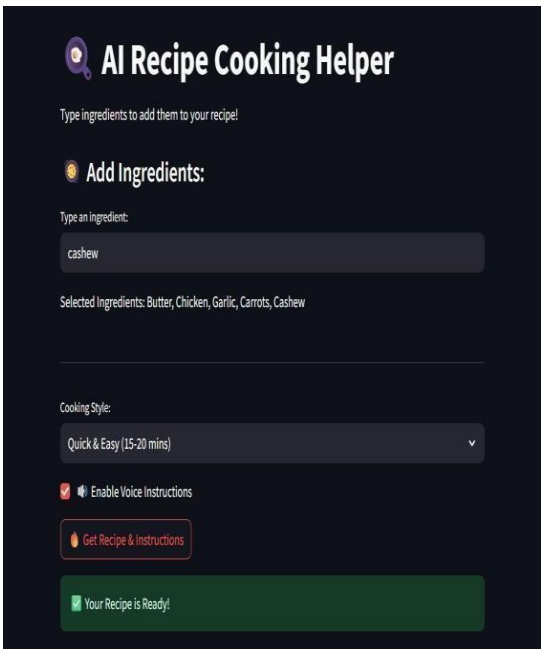
The suggested cooking times aligned well with style preferences. Quick recipes usually took about 15 to 20 minutes. Medium recipes took around 30 to 45 minutes, while more elaborate options required over 60 minutes due to complex techniques. This also demonstrated Gemini's ability to understand the limits on cooking time.

Most times Gemini recipes manage to impress, as some underwhelming scenarios take place including imprecise measures requiring user judgment, missing oven cooking temperatures, inconsistency in step details, and sporadic ingredient misalignment. No validation mechanism exists, other than structural formatting checks.

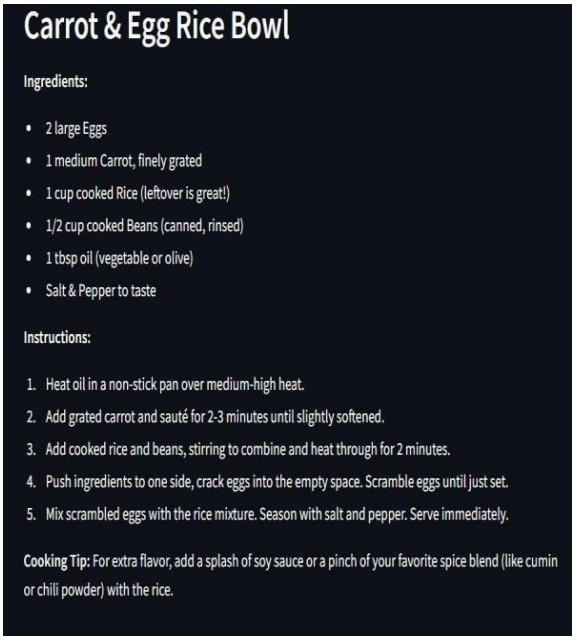
Imposition of recipe validation would justify stated requirement elements such as cooking temperature, reasonable quantities of ingredient, logical sequencing of steps, and detection of potentially unsafe combinations. However extensive domain knowledge encoding is entailed in such a judgment.

This current application serves as a targeted tool for quick recipe generation rather than a complete cooking platform. This simplicity stands both as an advantage for selected cases and a disadvantage for users wanting strong-end meal coordination.

Text-to-speech technology has been more advanced than most other technologies; however, it has its limitations, such as problems with ethnically or gastronomically specialized term pronunciations, lack of emotional connotations imparted by real human instructors, no break or emphasis on certain words depending on the user's progress, and fixed content that doesn't respond to the user's prompts.



This interface screen shows an "AI Recipe Cooking Helper" where users can add ingredients and choose a cooking style. A success message indicates that the recipe has been generated.



A recipe page titled "Carrot & Egg Rice Bowl" displays its required ingredients and cooking steps. The layout also includes a cooking tip for enhancing the dish's flavor.

A recipe page for "Cashew Chicken Stir-Fry" lists ingredients and step-by-step cooking instructions. An audio playback option is available at the bottom for voice-guided cooking.

Conclusion

An AI cooking assistant was successfully conceived to respond to the targeted challenges of the home cooking domain through intelligent technology integration. It achieves smart recipe discovery through an ingredients-based search, AI generation of recombinant recipes, and voice-synthesized tutorials in a web interface.

Accomplishments include implementing simple ingredient selection using categorized checkboxes, integrating Google Gemini AI for dynamic recipe generation, developing a text-to-speech module for hands-free cooking instruction, creating a UI that is simple to use and does not require any technical expertise from the user, optimizing responses generated for practical kitchen use, and deploying an application intended to run on multiple hosting platforms. The system generates different creative recipes based on user ingredients and provide instruction back in visual and audio formats, having processed the entire cycle from ingredient selection to audio playback in less than 15 seconds. It accounts for cooks based on their skills through dynamic recipe complexity.

Some technical highlights cover prompt engineering for good recipe quality, integrating multiple APIs into one workflow, handling temporary files for audio generation, responsive UI design across devices, and very few dependencies allowing easy deployment and maintenance.

Impact and Applications

The application tackles food wastage head-on by getting users to cook with whatever ingredients they have instead of leaving them in the fridge. This contributes positively to the sustainability goal and helps users cut out grocery expenses. The instant recipe suggestion offers home cooking rather than takeout, thereby improving nutrition and quality of food.Inept cooks gain confidence through clear AI-generated instructions tailored to their ingredients. The voice feature particularly benefits the new cook who needs reassurance and step-by-step guidance along the way.

Future Work:

Enhanced Ingredient Management:

A logical next step would be allowing users to add custom ingredients not found in the preset list. The app could also include quantity tracking so that users specify how much of each ingredient they have. A built-in pantry tracker could monitor ingredient inventory over time. Additional filters could help users choose recipes based on dietary preferences like vegetarian, vegan, gluten-free, or allergy-sensitive options. Nutritional analysis could also be integrated to calculate calories, macronutrients, and general health data for each recipe.

Advanced AI Capabilities:

The system could evolve to enable interactive dialogues where users refine their recipes—requesting changes such as “make it spicier” or “reduce cooking time.” It could assess recipe complexity automatically, flag unsafe ingredient combinations, and learn from user behavior to tailor suggestions over time.

Voice Interaction Enhancements:

Future updates might bring complete voice command functionality, allowing users to prepare meals hands-free. Improvements in text-to-speech could make the voice output sound more natural, include multiple tone options, and support adjustable speaking speeds. Stepwise voice guidance could read instructions one by one, pausing between steps. In addition, an interactive voice assistant could respond to user questions during the cooking process. Multilingual support would extend the system’s accessibility worldwide.

References

[1].Ahn, Y. Y., Ahnert, S. E., Bagrow, J. P., & Barabási, A. L. (2011). Flavor network and the principles of food pairing. Scientific Reports, 1(1), 196.

[2].Majumder, B. P., Li, S., Ni, J., & McAuley, J. (2019). Generating Personalized Recipes from Historical User Preferences. In Proceedings of the Conference on Empirical Methods in Natural Language Processing.

[3].Salvador, A., Drozdzal, M., Giro-i-Nieto, X., & Romero, A. (2019). Inverse Cooking: Recipe Generation from Food Images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[4].Van den Oord, A., Dieleman, S., Zen, H., et al. (2016). WaveNet: A Generative Model for Raw Audio. arXiv preprint arXiv:1609.03499.

[5].Shen, J., Pang, R., Weiss, R. J., et al. (2018). Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. In IEEE International Conference on Acoustics, Speech and Signal Processing.

[6].Google AI (2024). Gemini: A Family of Highly Capable Multimodal Models. Technical Report, Google DeepMind.

[7].Streamlit Inc. (2024). Streamlit Documentation.

Retrieved from https://docs.streamlit.io/

[8].Google Cloud (2024). Cloud Text-to-Speech API Documentation. Retrieved from: https://cloud.google.com/text-to-speech/docs