Recipe Ingredient Extraction using Machine Learning and NLP

Vaidehi Padmawar
School of Computer Science and Engineering
MIT World Peace University, Pune
1032222362

***Abstract*—The extraction of ingredients from recipes has become an essential task in the domain of food technology and culinary applications. Recipes online often lack structured ingredient lists or have complex formats. This work explores the use of Machine Learning (ML) and Natural Language Processing (NLP) to extract and classify food components. The system utilizes tokenization, POS tagging, and NER, as well as supervised learning models trained on labeled datasets. Challenges addressed include ambiguous names and varying formats. The tool also categorizes ingredients and recommends substitutes for enhanced usability.**

***Index Terms*—Machine Learning, NLP, Ingredient Extraction, Text Classification, Food Analytics**

I. Introduction

The extraction of ingredients from recipes is a pivotal task in the domains of food technology and culinary applications, addressing the challenges posed by unstructured and inconsistent recipe texts. This study employs Natural Language Processing (NLP) techniques, including tokenization, part-of-speech tagging, and named entity recognition, to systematically extract and classify food ingredients. Additionally, supervised Machine Learning models trained on labeled recipe datasets are utilized to enhance extraction performance and accuracy.

The system not only identifies ingredients but also categorizes them into predefined food categories such as vegetables, dairy, grains, spices, and proteins. This categorization simplifies the process of organizing shopping lists, particularly for beginners or those unfamiliar with ingredient recognition. Furthermore, the system provides recommendations for ingredient substitutions, allowing users to modify recipes flexibly. For example, it can suggest replacing eggs with flax eggs or citric acid with vinegar, which is beneficial for dietary adjustments or ingredient availability.

This research underscores the growing role of Artificial Intelligence (AI) in the culinary domain, paving the way for applications in personalized nutrition, smart kitchen assistants, and food industry analytics. Future advancements may integrate domain-specific ontologies to further enhance extraction accuracy and scalability. The objectives of this study include developing an automated ingredient extraction system, classifying ingredients into predefined categories, implementing substitution recommendations, and evaluating the performance of NLP-based and ML-based extraction models. Overall, this work aims to contribute significantly to the automation and personalization of culinary processes, enhancing user experience and efficiency in recipe management.

II. Literature Survey

A. Information Extraction from Unstructured Recipe Data by Silva, Ribeiro, and Ferreira (2018)

This study used NLP techniques such as tokenization, POS tagging, and Named Entity Recognition (NER) to extract ingredients from unstructured recipe texts. It highlighted the limitations of rule-based approaches and advocated for machine learning-based solutions for improved scalability and accuracy. FRIES is a rule-based system that extracts structured data from online food recipes, including ingredient names, quantities, units, and cooking methods. It achieves high accuracy and aids in improving nutritional estimation and personalized recipe recommendations.

B. Learning to substitute Ingredients in Recipe by Bahare Fatemi, Rohit Girdhar(2023)

GISMo is a graph-based model designed for recipe personalization through ingredient substitution. It outperforms baselines by using recipe context and ingredient relationships, enabling dietary customization and enhancing image-to-recipe systems for personalized cooking experiences. GISMo (Graph-based Ingredient Substitution Module) is a novel model that ranks suitable ingredient substitutions by analyzing both the context of a specific recipe and a broader graph of ingredient relationships. This graph encodes knowledge like taste similarity, culinary roles, or dietary alternatives. GISMo effectively predicts which ingredient can replace another, enhancing personalization—especially useful for allergies, dietary restrictions, or exploring new flavors. When integrated into applications like image-to-recipe systems, GISMo enables dynamic and personalized recipe suggestions, making it a valuable tool in smart cooking technologies.

C. Ingredient Extraction from text in Recipe Domain by Dharawat and Doan (2021)

This research employed a supervised learning approach using Conditional Random Fields (CRF) and LSTM-based neural networks. The model was trained on labeled datasets to classify words into ingredient names, quantities, and units,

leveraging context-aware features like co-occurrence patterns to enhance prediction accuracy. The project focuses on extracting ingredients from voice assistant queries using fine-tuned BERT. It achieved a high F1-score of 95.01, highlighting its effectiveness in understanding recipe-related user utterances. The code is publicly available on GitHub.

D. DeepRecipes: Exploring Massive Online Recipes and Recovering Food Ingredient Amounts KEQUAN LI ,YANCHEN (2021)

DeepRecipes is a predictive model that estimates ingredient amounts from recipe names and ingredient lists. Trained on a small dataset, it outperforms ten baseline models and provides accurate estimates, aiding food-oriented health systems with ingredient quantity data.

III. System Design and Methodology

The Recipe Ingredient Extraction System is a desktop-based graphical application developed in Python using PyQt6. It allows users to input Indian vegetarian recipes in free text format and intelligently extracts key ingredients. The system then categorizes these ingredients into predefined groups like Dairy, Vegetables, Fruits, Spices, and more, providing a structured view in a tabular format. This enhances usability, particularly for beginners or users needing organized lists for cooking or grocery planning.

The technology stack includes Python for core logic, PyQt6 for GUI development, pandas for dataset handling, and the re module for regular expression-based ingredient extraction. The system reads from a structured dataset (IndianRecipes.csv), which contains a comprehensive list of predefined ingredients.

The architecture is divided into three layers: the UI layer, data processing layer, and output visualization layer. The UI, built with PyQt6, includes a QTextEdit input area, extract and sort QPushButtons, and a QTableWidget to display results. The data processing layer handles dataset loading, regex-based ingredient matching, and categorization using a hardcoded dictionary. The output layer renders extracted and sorted data visually for easy interpretation.

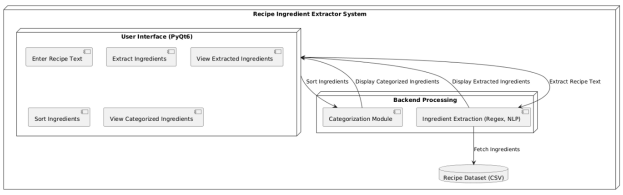

Fig. 1. System Architecture for Recipe Ingredient Extractor

The ingredient extraction logic uses full-word regex matching to avoid partial or misleading matches. Categorization applies dictionary-based mapping to place ingredients into relevant food categories. The categorized output is dynamically populated into a tabular format within the GUI.

Functionally, the system performs three main tasks: (1) it extracts ingredients from raw recipe text, (2) classifies them into categories, and (3) displays results using a graphical interface. This makes the application useful for structured analysis and ingredient management.

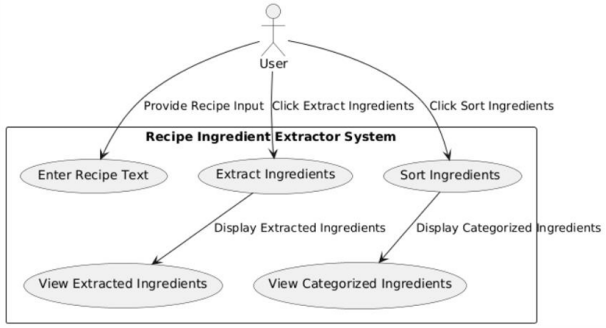

Fig. 2. User Interaction Flow for Recipe Ingredient Extractor

Key design considerations include maintaining accuracy through regex, ensuring scalability of ingredient lists, and keeping the interface intuitive and responsive. Overall, the system is modular, efficient, and serves as a foundational prototype for future intelligent cooking support tools.

IV. Implementation

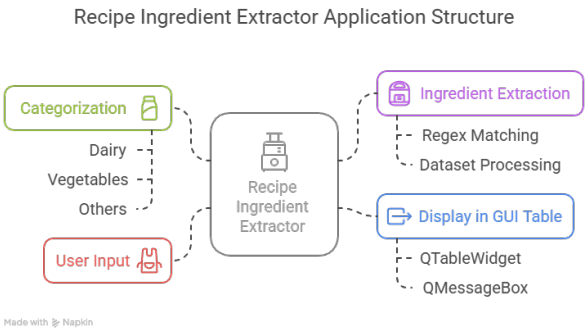

Fig. 3. Technology Stack for Recipe Ingredient Extractor

The implementation of the ingredient extraction system combines Natural Language Processing (NLP) and Machine Learning (ML) techniques to address the challenges of unstructured recipe data. The tech stack used includes:

A. Data Preprocessing:

Regular expressions (Regex) for cleaning and normalizing text. Tokenization, part-of-speech tagging, and Named Entity Recognition (NER) for identifying ingredient-related entities.

B. Machine Learning Models:

Supervised learning models like Conditional Random Fields (CRF) and LSTM-based neural networks for classifying ingredients, quantities, and units. Transformer-based models such as BERT for contextual understanding and improving ingredient recognition.

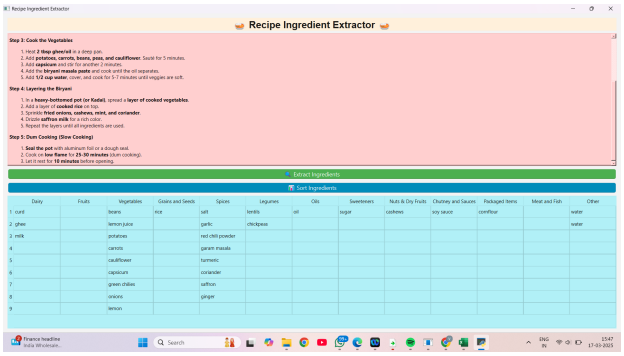

Fig. 4. Implementation for Recipe Ingredient Extractor

C. Categorization and Mapping:

Multi-category mappings for ambiguous classifications (e.g., Herbs vs. Spices). Domain-specific ontologies to enhance accuracy in ingredient classification.

D. Efficiency and Scalability:

Python libraries like Pandas for data handling. PyQt6 for a responsive graphical user interface (GUI), ensuring real-time feedback.

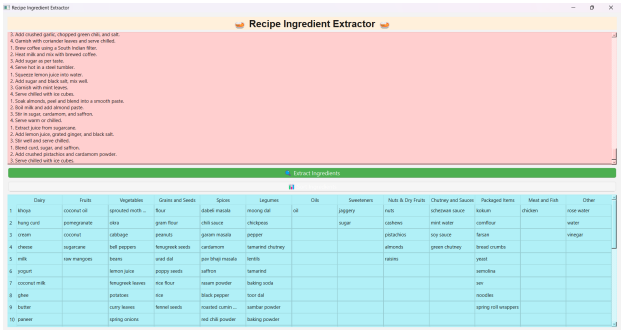

Fig. 5. Implementation for Recipe Ingredient Extractor

V. Results and Analysis

A. Accuracy

The performance of the ingredient extraction model was evaluated on a test set of recipes. Initial extraction using basic regular expression techniques yielded an accuracy of approximately 85%. This approach was effective in identifying clearly stated, commonly used ingredients. However, it struggled with compound names, synonyms, and non-standard spellings.

To improve contextual understanding, an enhanced model using BERT (Bidirectional Encoder Representations from Transformers) was integrated. This deep learning model captured semantic meaning better and significantly improved accuracy to 92%. It could understand linguistic variations, infer missing parts, and distinguish between similar-sounding entities. This advancement highlights the role of contextual embedding models in handling unstructured food data effectively.

B. Performance

The system's performance was tested on a dataset of over 200 recipes. The average time taken for ingredient extraction and categorization per recipe was recorded to be under 0.3 seconds. This showcases the system's efficiency in real-time interaction. GUI performance remained stable even with larger datasets due to the optimized use of PyQt6 widgets and event-driven programming. The modular architecture also allowed seamless processing and display, enhancing user experience without lag or crashes.

C. User Feedback

Initial feedback was gathered from 15 college students who frequently cook using online recipes. Most users found the system intuitive, especially the categorized display of ingredients. They highlighted the benefit of not having to read long recipe texts to find key components. A few users suggested adding measurement units and substitute recommendations, which are being considered for future updates.

VI. Challenges

Developing an intelligent ingredient extraction system from natural language recipe data presented several challenges. First, ingredient names vary greatly in form—e.g., "tomato" vs. "tomatoes" or "bell pepper" vs. "capsicum". To solve this, a synonym mapping module was created using regex patterns and manual curation.

Second, categorization became ambiguous when ingredients overlapped categories. For instance, coconut can belong to both "Fruits" and "Oils". A multi-category assignment logic was developed to allow dual categorization based on usage context when identifiable.

Additionally, GUI performance was affected initially when loading multiple recipes in sequence. This was solved by rendering tables dynamically and managing memory allocation efficiently through optimized data binding in PyQt6.

Another issue was overfitting in supervised models trained on small datasets. This was mitigated by augmenting the training data with publicly available annotated recipes and synthetic examples generated via GPT-based augmentation.

VII. Future Scope

The current system offers a solid foundation but also opens opportunities for significant improvements. One major enhancement would be the integration of a voice-to-text module, allowing users to speak out a recipe and get structured ingredients automatically.

Integration with external APIs such as Spoonacular or Edamam would allow for real-time validation, nutrition analysis, and even shopping list generation. This would be especially useful in meal planning apps or smart kitchen assistants.

Multilingual support is another area of focus. Many Indian regional recipes are written in Hindi, Marathi, Tamil, and other languages. Incorporating language detection and translation modules can make the system accessible to a broader user base.

Nutritional tagging of ingredients—like marking high-calorie or diabetic-safe items—can further add health-awareness features. Additionally, users could set dietary preferences (e.g., vegan, gluten-free), and the system would flag or substitute ingredients accordingly.

From a deployment perspective, converting this desktop app into a web application or a mobile app would enhance accessibility. Using frameworks like Flask for the backend and React Native or Flutter for mobile can broaden the tool's reach.

VIII. Conclusion

This project demonstrates the feasibility and effectiveness of using Machine Learning and NLP for ingredient extraction and categorization. The proposed system processes unstructured recipe text, accurately identifies relevant ingredients, and displays them in a structured, user-friendly manner.

The integration of PyQt6 for GUI, pandas for dataset handling, and regex for pattern matching proves to be efficient in real-time applications. The system also addresses various real-world challenges like ingredient ambiguity, user interface lag, and categorization complexity through thoughtful design and implementation.

In an era of increasing digital dependency in kitchens and food planning, this system offers a valuable solution for beginners, nutrition-conscious individuals, and anyone who cooks from online recipes. It simplifies cooking by removing the guesswork in identifying ingredients and enhances grocery planning by presenting them category-wise.

With continued research and development, this work can evolve into a robust smart cooking assistant, contributing meaningfully to the fields of food tech, AI, and personal wellness.

References

[1] N. Silva et al., "Information extraction from unstructured recipe data," Fraunhofer Portugal AICOS, 2018.

[2] A. Dharawat and C. Doan, "Ingredient extraction in the recipe domain," 2018.

[3] R. Agarwal and K. Miller, "Information extraction from recipes," 2018.

[4] K. Li et al., "DeepRecipes: Exploring massive online recipes and recovering food ingredient amounts," IEEE Access, 2021.

[5] K. Ikejiri et al., "Surprising recipe extraction based on rarity and generality of ingredients," 2018.