Projet Report

ChefMate: A chatbot for recipe recommendations

Hamdi Mokni

ESPRIM

INFO3

Online Website: https://chefmate.onrender.com/

GitHub Repository: https://github.com/Mk-1000/ChefMate

Overleaf Link: https://www.overleaf.com/read/bgnjgmwmtjpx425a51

October 21, 2024

Contents

0.1 Introduction

0.1.1 Purpose of the Report

This Report summarizes the objectives, implementation strategies, and accomplishments achieved during my project entitled NLP Recipe Suggestion Application. The work presented herein is an overview of the methodologies applied, used technologies, and how this innovative application currently changes people's behavior in searching for food and user experience in general.

0.1.2 Project Overview

The recipe suggestion natural language processing application leverages NLP for effective suggestions, given the ingredient list a user submits. In addition, it is a Flask web application that is perfectly integrated with the API developed by Spoonacular, thus presenting users with great opportunities to browse through recipes that match their ingredients at hand. This system is highly user-friendly and thus provides access via natural language queries to the users. Further, this application can be accessed online through the link
urlhttps://chefmate.onrender.com/ to inspire creativity in cooking and reduce food waste by utilizing the items present in one's kitchen to their full potential.

0.2 Background

0.2.1 NLP in Culinary Applications

It is a strong domain in the area of artificial intelligence that can extract semantically meaningful information from raw, unstructured text data. In culinary applications, it enables users to interact with the system in a more intuitive and conversational manner, hence increasing the level of engagement and satisfaction. Consequently, the application can go on to suggest recipes of choice by properly interpreting what the user is asking for, thereby again easing up the process of meal planning for a user.

0.2.2 Existing Solutions

Even though many applications have been developed for recipe recommendations, most of them are doing a poor job when it comes to correctly extracting or understanding the ingredients from various types of natural language inputs. Reported approaches in literature today either require very rigid input formats or employ error correction in an extremely limited way, ultimately discouraging users. This is an important gap that needs to be bridged with an effective NLP pipeline designed for culinary contexts. The system will adopt the most recent techniques of spelling correction, synonym recognition, and ingredient normalization, so that proper recipe suggestions can be made even if the users' inputs are informal or imprecise.

0.3 Project Objectives

General project objectives include the following:

- The aim is to develop a web-based application; this application will suggest recipes to users based on the inventory of ingredients provided.
- Parse the user's input using effective NLP to extract, interpret, and then normalize the ingredients correctly.
- It can measure the application's scalability and responsiveness for many users from performance onward.

0.4 Implementation

0.4.1 Used Technologies

The following technologies were utilized throughout the project:

- **Flask**: Is a lightweight Python web framework that makes it very easy to build this web application.
- **spaCy**: A more advanced library of NLP used for performing a variety of tasks around language processing.
- **Requests**: A library designed to make HTTP requests as smoothly as possible using APIs.
- **HTML/CSS**: Basic technologies for the creation of front-end development, thus enabling the creation of a responsive user interface.

0.4.2 Code Overview

Flask Application

The Flask application will be a server-side web application whose purpose is to obtain recipes based on the ingredients provided by the end user. It should feature routing mechanisms, API interactions, and error handling to make the experience better for the user.

Here is some simplified code illustrating how recipes are fetched:

```
@app.route('/get_recipes', methods=['POST'])
def get_recipes():
    ingredients = request.json.get('ingredients', '')

    if not ingredients:
        return jsonify({'error': 'No ingredients provided.'})

    response = requests.get(
        f'{BASE_URL}findByIngredients',
        params={'ingredients': ingredients, 'apiKey': API_KEY}
    )

    if response.ok:
        return jsonify(response.json())

    return jsonify({'error': 'Failed to fetch recipes.'})
```

0.5 NLP in Culinary Applications

NLP is a revolutionary technology that basically allows machines to understand and interpret human languages and generate them in useful ways. This contributes highly, especially in recipe applications, where intuitive communication is allowed with the incorporation of an application, hence making interaction easy, allowing users to project queries in natural language.

Some of the major NLP techniques implemented in the application are as follows:

- **Ingredient Name Identification/Extraction**: The user inputs are analyzed for correct identification and extraction of ingredient names.

```python
def extract_ingredients(nlp_text):
    """Enhanced ingredient extraction."""
    corrected_text = correct_spelling(nlp_text).lower()  # Correct spelling and normalize to lowercase
    print("Corrected Text:", corrected_text)

    doc = nlp(corrected_text)

    # Extract ingredients from NER and PhraseMatcher
    ingredients = [ent.text for ent in doc.ents if ent.label_ == 'FOOD']
    ingredients += extract_custom_ingredients(doc)

    print("Extracted Ingredients (before fallback):", ingredients)


    # If no ingredients, fall back to extracting common nouns
    if not ingredients:
        ingredients = [token.text for token in doc if token.pos_ == 'NOUN' and token.is_alpha]

    # Add synonyms for recognized ingredients
    all_ingredients = ingredients[:]
    for item in ingredients:
        all_ingredients += get_synonyms(item)

    # Extract quantities and add them to the ingredients list
    quantities = extract_quantities(corrected_text)
    ingredients_with_quantities = [f'{quantity} of {ingredient}' for quantity, ingredient in quantities]

    # Merge all extracted ingredients and remove duplicates
    all_ingredients += ingredients_with_quantities
    all_ingredients = list(set(all_ingredients))

    print("Final Ingredients:", all_ingredients)

    return ', '.join(all_ingredients)

```

- **Spell Check/Cleaning**: Spell checking is implemented as an algorithm to correct spelling mistakes in the user's input, which enhances the extraction process's robustness.

```python
def correct_spelling(text):
    """Correct spelling mistakes in user input."""
    corrected = [spell.correction(word) for word in text.split()]
    return ' '.join(corrected)

```

- **Synonym Identification**: This tool uses libraries such as WordNet, which make the process of finding synonyms and different variations of ingredient names easier.

```python
def get_synonyms(word):
    """Retrieve synonyms for a given word using WordNet."""
    synonyms = set()
    for synset in wn.synsets(word):
        for lemma in synset.lemmas():
            synonyms.add(lemma.name())
    return list(synonyms)

```

- **Contextual Understanding**: The application knows during what context the ingredient mentions pop up. This will lead to more valid recipe suggestions being returned due to such filtering.

```python
def contextual_understanding(nlp_text):
    """Analyze text context for ingredient recognition."""
    doc = nlp(nlp_text)
    context_data = [(token.text, token.dep_) for token in doc]
    return context_data

```

- **Phrase Correspondence**: Implementing phrase matching methodologies in the application enables it to recognize set phrases and patterns that denote ingredient lists.

```python
def extract_custom_ingredients(doc):
    """Extract ingredients using PhraseMatcher."""
    matches = matcher(doc)
    return [doc[start:end].text for match_id, start, end in matches
   ]

```

- **Quantity Extraction**: The recipe application can use regular expressions to find ingredients with quantities ("2 cups of sugar") and factor them into the suggested recipes.

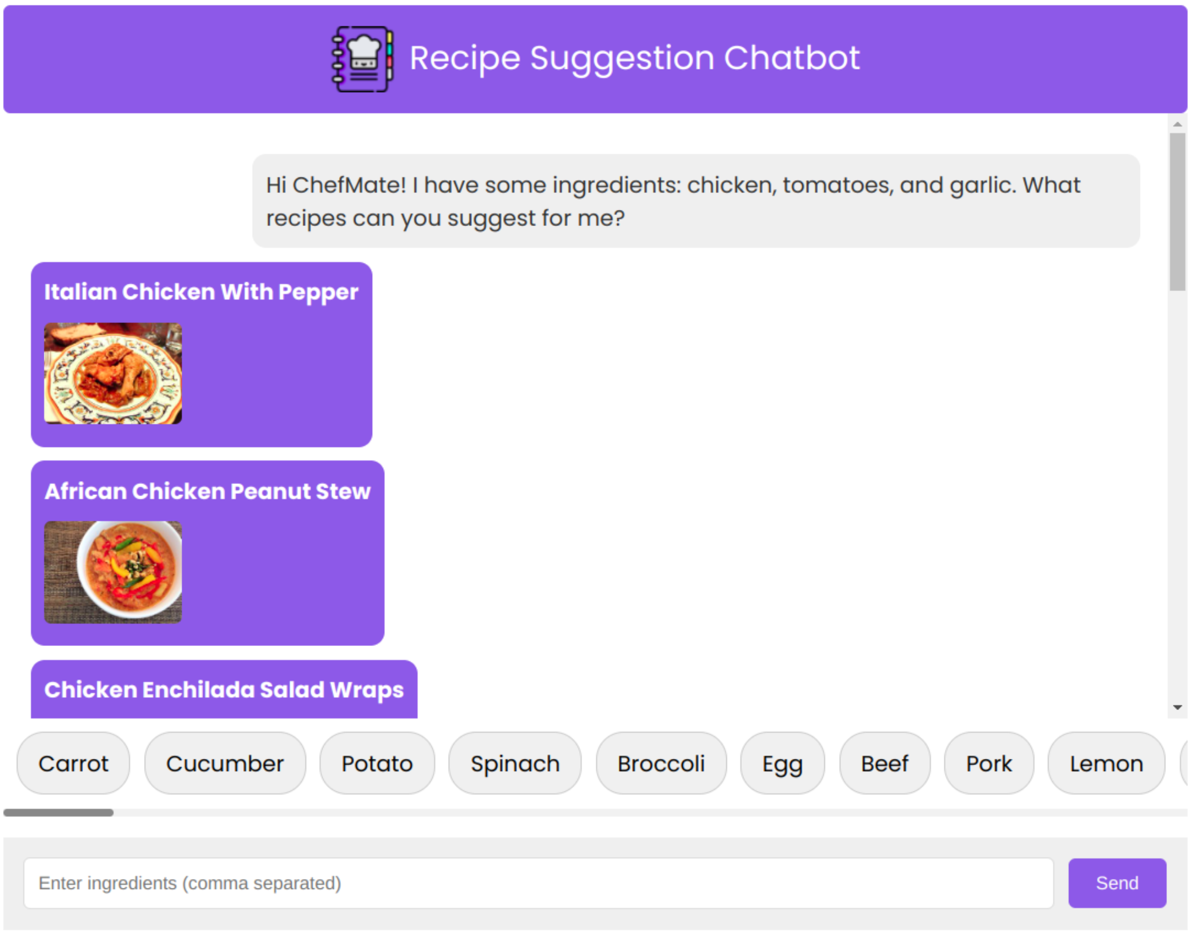
```python
def extract_quantities(text):
    """Extract quantities of ingredients from text using regex."""
    pattern = r'(\d+\s?(?:cups?|tsp|tbsp|grams?|kg|liters?|ml|
   ounces?|oz|pounds?|lbs?))\s(.+)'
    matches = re.findall(pattern, text)
    return matches

```

0.5.1 User Interface

The following screenshots illustrate the graphical user interface of the Recipe Suggestion NLP Application:

User Interface Screenshots

The following screenshots illustrate the user interface of the NLP Recipe Suggestion Application:


Recipe Suggestion Chatbot
Carrot
Cucumber
Potato
Spinach
Broccoli
Egg
Beef
Pork
Lemon
Enter ingredients (comma separated)
Send

Hi ChefMate! I have some ingredients: chicken, tomatoes, and garlic. What recipes can you suggest for me?

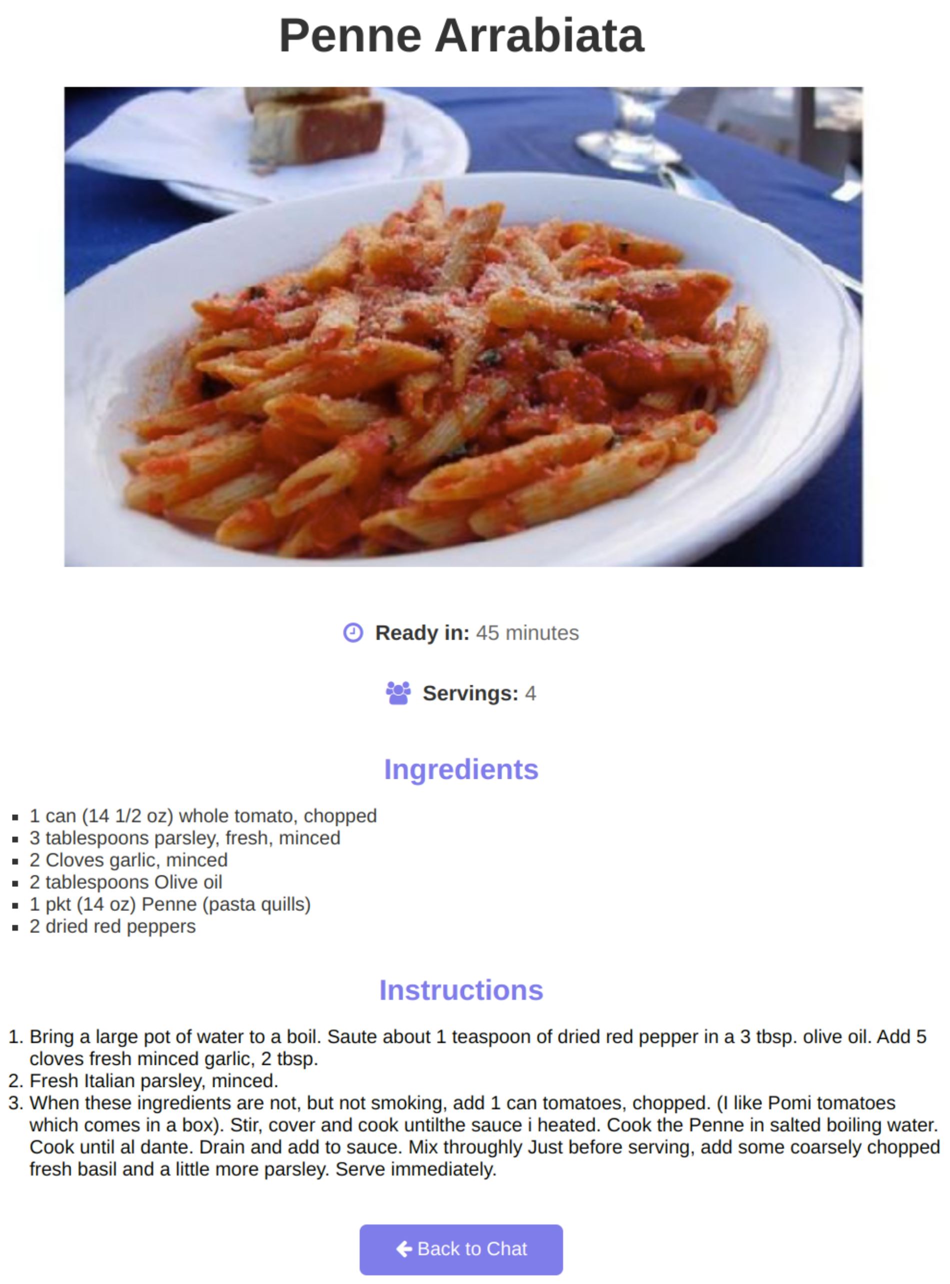
Italian Chicken With Pepper

African Chicken Peanut Stew

Chicken Enchilada Salad Wraps

Figure 1: Main Page for Ingredient Input

Description: This main interface allows users to input their ingredients to get recipe recommendations personalized to what the user has put in.



Penne Arrabiata

Ready in: 45 minutes

Servings: 4

Ingredients

- 1 can (14 1/2 oz) whole tomato, chopped
- 3 tablespoons parsley, fresh, minced
- 2 Cloves garlic, minced
- 2 tablespoons Olive oil
- 1 pkt (14 oz) Penne (pasta quills)
- 2 dried red peppers

Instructions

1. Bring a large pot of water to a boil. Saute about 1 teaspoon of dried red pepper in a 3 tbsp. olive oil. Add 5 cloves fresh minced garlic, 2 tbsp.
2. Fresh Italian parsley, minced.
3. When these ingredients are not, but not smoking, add 1 can tomatoes, chopped. (I like Pomi tomatoes which comes in a box). Stir, cover and cook untilthe sauce i heated. Cook the Penne in salted boiling water. Cook until al dante. Drain and add to sauce. Mix throughly Just before serving, add some coarsely chopped fresh basil and a little more parsley. Serve immediately.

Figure 2: Detailed Recipe Page

Description: This page shows selected recipe information, which includes ingredient details and the steps to be taken for cooking.

0.6 Challenges and Solutions

0.6.1 Challenges Faced

The subsequent points represent several pivotal challenges encountered throughout the process of application development:

- **Extracting Ingredients**: The inconsistency of the data provided by the user made the task of extracting ingredients very difficult, both in terms of consistency and accuracy.
- **API Constraints and Operational Rates**: Most of the errors taken upon Spoonacluar API included limits on the rate and error in responses.
- **Validation of User Input**: Grammatical spelling and ambiguities with regard to ingredient names tested completeness of language and context understanding.

0.6.2 Solution Applied

To effectively address these challenges, the following solutions were implemented:

- **Improved NLP Pipelin**: A strong pipeline for NLP was set up that included spell correction, synonym extraction, and phrase matching. This hugely improved the accurate extraction of ingredients from user input provided in many different ways.
- **Complete Error Handling**: Complete Error Handling: Integration of complete error handling for API interaction, including graceful degradation of functionality. This improved user experience by providing informative error messages with fallbacks.
- **Normalization of User Input**: Worked on mechanisms that handle ingredient name and quantity input in various ways to enhance the accuracy of the ingredient recognition process. This included the use of regex to extract quantities and keep the ingredient list uniform.
- **Response Cachin**: A designed caching policy was used for highly accessed recipes in order to minimize redundant API calls and improve performance. By this way, the strategy kept the rate limit impact and improved responsiveness in the application.

0.7 Reflections

0.7.1 Learnings

Throughout the project, I acquired valuable skills in NLP techniques and API integration, which have significantly enhanced my technical proficiency.

0.7.2 Future Improvements

Potential enhancements for the application include:

- **User Authentication and Profiles**: A secure user authentication system can be provided, enabling users to create profiles whereby they can save their favorite recipes, handle their ingredient lists, and also suggest recipes to them tailored to their culinary tastes and dietary needs.

- **Enhanced NLP Techniques**: Implementing more advanced techniques of natural language processing, including deep learning models, would boost further the accuracy of ingredient extraction and overall meaning extraction of user input. Techniques such as NER and contextual embeddings could be useful in enhancing the identification of ingredients along with their actual quantities and cooking methods.

- **Recipe customization features**: Further development may focus on filtering mechanisms regarding dietary preference options-vegan and gluten-free-along with cooking time, for increased customer satisfaction. Additionally, a machine learning model could analyze user behavior to provide recipe suggestions that they are more likely to enjoy.

- **Integration with External APIs**: Looking into partnerships with grocery delivery services that allow users to order ingredients directly through the app may give an perfect cooking experience. The inclusion of the database with food-related blogs and websites that cater to cooking-related topics would further enhance its interaction with the user.

- **User Feedback Mechanism**: A feedback mechanism to be provided at the end will enable users to rate and comment on various recipes. These could then be used to fine-tune the recipe recommendations for better performances in the future.

0.8 Conclusion

It ended with the development of a novel application based on natural language processing and further integrated into enhancing the culinary experience. It can provide personalized suggestions of recipes based on the entered ingredients. Through advanced natural language processing and easy-to-use interfaces, the users of the app are imaginatively and effectively able to find new recipes. Capable of improvement, the system will take a strategic place in the kitchen as it serves users with not only convenience but interaction with the culinary experience.

Bibliography

[1] Flask. (n.d.). *Flask Documentation.* Retrieved from `https://flask.palletsprojects.com/en/2.3.x/`

[2] Requests. (n.d.). *Requests Documentation.* Retrieved from `https://docs.python-requests.org/en/latest/`

[3] spaCy. (n.d.). *spaCy Documentation.* Retrieved from `https://spacy.io/`

[4] NLTK. (n.d.). *NLTK Documentation.* Retrieved from `https://www.nltk.org/`

[5] WordNet. (n.d.). *WordNet Website.* Retrieved from `https://wordnet.princeton.edu/`

[6] PySpellChecker. (n.d.). *PySpellChecker Documentation.* Retrieved from `https://pyspellchecker.readthedocs.io/en/latest/`

[7] spaCy. (n.d.). *PhraseMatcher Documentation.* Retrieved from `https://spacy.io/api/matcher#phrase-matcher`

[8] Python Software Foundation. (n.d.). *re Module Documentation.* Retrieved from `https://docs.python.org/3/library/re.html`

[9] Spoonacular. (n.d.). *Spoonacular API Documentation.* Retrieved from `https://spoonacular.com/food-api/docs`