LLM Based Recipe Generation with Nutritional Constraints

Ankit Kumar Pradhyumn Bhardwaj Omkar Dalvi Nikhil Rane Atharva Patil

Abstract

With the growing capabilities of Large Language Models (LLMs) across various domains, one promising application is in the field of automated recipe generation. Traditionally, people rely on cookbooks and manual methods to create recipes, but LLMs now enable the automatic generation of creative and coherent recipes based on user inputs. However, a gap remains in generating recipes tailored to specific nutritional requirements, which is a key concern for nutritionists and fitness enthusiasts. This project introduces a new task for recipe generation under nutritional constraints, such as target calorie limits and specific macronutrient distributions. We fine-tune a pre-trained state-of-the-art LLM for this task, optimizing training and inference pipelines for ef icient recipe generation. Additionally, we perform benchmarking and profiling to assess model performance and conduct thorough evaluations against baseline models, including of -the-shelf LLMs. This work contributes to the intersection of AI and nutrition, enabling automated, personalized Look at previous meal planning for health-conscious users.

1. Literature Review

The increasing availability of food-related content online has complicated meal decision-making, highlighting the need for advanced algorithms and recommendation systems. Traditional recommender systems often fail to address the complexities of food choices, which significantly impact health outcomes such as obesity and chronic diseases. Integrating health components into food recommendations can promote healthier eating behaviors among users. Recent studies have focused on utilizing Large Language Models (LLMs) for personalized recipe generation that meets nutritional needs. A study published in Nature Scientific Reports demonstrates the effectiveness of finetuning LLMs to create meal plans tailored to specific dietary goals, proving that LLMs can generate coherent and nutritionally relevant recipes [1]. Additionally, research utilizing LLMs in conjunction with knowledge graphs indicates that structured nutritional information can enhance the quality of recipe generation, outperformingstandard models [2]. Another significant contribution is the study titledPersonalized Health-Aware Recipe Recommendation, which introduces a novel approach for generatinghealthyrecipes based on a limited number of ingredients. Themodel consists of three main components: predictingpersonalized ingredients, fine-tuning the GPT-2model forrecipe generation, and recommending similar recipesfroma dataset. This approach effectively balances user preferences with health considerations, showcasingthepotential of personalized recipe generation to encouragehealthier eating habits [3]. Overall, these studies highlight the promise of LLMsinthefood recommendation domain, paving the wayfor systemsthat provide diverse meal options while promotingbetterhealth outcomes through personalized nutrition

2. Data Collection and Preprocessing

Data Cleaning The first step in the preprocessing pipeline is data cleaning, which involves identifying and handling inconsistencies, missing values, and incorrect data formats.

2.1 Handling Missing Values: Missing values are a common issue in real-world datasets. In the case of recipes, essential columns such as ingredients, descriptions, and instructions must be free from missing values. The following methods were employed to handle missing data:

Identifying Missing Values: The dataset is first examined to identify any null or missing values, particularly in columns crucial for recipe generation (e.g., ingredients, instructions). Filling Missing Values: In cases where ingredients or instructions are missing, imputation methods are applied. For ingredients, missing values may be filled with a generic placeholder (e.g., ”unknown ingredient”). For instructions, if the entire recipe is missing, the row may be dropped or flagged for review.

2.2 Cleaning and Standardizing Ingredient Lists: Recipes often contain lists of ingredients that may be inconsistent in formatting (e.g., different ways of writing quantities or ingredients). These lists are standardized by:

Splitting Ingredients: Ingredients are separated into individual items if they are listed as a single string (e.g.,

”chicken, tomatoes, onions” is split into ”chicken,” ”tomatoes,” ”onions”). Whitespace and Special Character Removal: Any extraneous whitespace, special characters, or unwanted symbols (e.g., commas, dashes) are removed from the ingredients list to ensure uniformity. Unit Normalization: Ingredients with quantities must be standardized. For example, “1 cup of rice” and “200 grams of rice” are converted to a uniform unit, such as grams, to ensure consistent analysis.

2.3 Verifying Nutritional Information: Nutrition columns may contain missing or inconsistent data. To ensure the dataset’s nutritional accuracy:

External APIs for Nutritional Augmentation: APIs such as CalorieNinja or Edamam are used to augment missing nutritional data, providing values for macronutrients (e.g., calories, proteins, fats, carbohydrates) where necessary. Nutritional Unit Conversion: Any units in the nutrition column (e.g., calories per serving, grams of protein) are standardized across the dataset.

2.4 Handling Outliers and Erroneous Data: Outliers or incorrect data, such as implausible cooking times or quantities, are flagged for review. Inconsistent or impossible values (e.g., a cooking time of 1000 minutes or negative quantities) are either corrected or removed.

3. EDA

Exploratory Data Analysis (EDA) is an essential step to understand the data’s structure, identify patterns, and uncover any hidden insights.

3.1 Understanding Data Distribution: The dataset’s columns, including ingredients, nutritional information, and cooking times, are analyzed for distribution:

Ingredient Frequency: The frequency of ingredients is visualized to identify commonly used ingredients across recipes. Nutritional Range: Descriptive statistics are calculated for numerical columns (e.g., calories, fat, protein) to understand their distribution and identify any extreme outliers. **3.2 Identifying Relationships and Patterns:** The relationships between ingredients, cooking times, and nutritional values are explored using pairwise scatter plots, correlation matrices, and heatmaps:

Correlation of Nutritional Content: Nutrient correlations are examined to understand how ingredients affect nutritional outcomes (e.g., fat and protein content often correlating). Recipe Categories: Recipe types or tags (e.g., vegetarian, gluten-free) are analyzed to determine how different types of dishes influence nutritional content. **3.3 Detecting Categorical Data Issues:** In categorical columns like recipe tags (e.g., “low-fat,” “high-protein”), we use the following methods to analyze and clean the data:

Label Encoding: For categorical data that will be used as input features for machine learning models, label encoding or one-hot encoding is applied. For example, the tag ”vegetarian” could be transformed into a binary column (0 or 1), indicating whether a recipe is vegetarian. Tokenization and Text Preprocessing: Recipe descriptions and cooking instructions are tokenized for easier processing. Techniques such as lowercasing, removal of stop words, and stemming/lemmatization are applied to reduce noise in the text.

4. Data Preprocessing

After the data has been cleaned and analyzed, several preprocessing steps ensure that it is ready for modeling.

4.1 Feature Engineering: For features such as ingredients and instructions, additional processing is done to make them machine-readable:

One-Hot Encoding for Tags: Tags (e.g., ”vegetarian,” ”low-fat”) are converted into one-hot encoded vectors, turning categorical variables into binary features that a machine learning model can process. Ingredient Quantification: Ingredients are transformed into a more structured format, such as an ingredient count matrix (one-hot encoding for each ingredient), where each row represents a recipe and each column represents an ingredient.

4.2 Text Preprocessing for NLP Models: The recipe description and cooking steps are preprocessed for any Natural Language Processing (NLP) tasks. These steps include:

Tokenization: The text is split into smaller units (words or phrases) to make it easier for machine learning algorithms to process. Stopword Removal: Commonly used words that do not contribute significant meaning (e.g., “the,” “and,” “of”) are removed to improve model performance. Stemming/Lemmatization: Words are reduced to their root form (e.g., “cooking” becomes “cook”) to handle different word variations.

4.3 Nutritional Data Normalization: To standardize the numerical features like calories, fat, and protein, the following steps are performed:

Min-Max Normalization: Continuous features, such as calories and protein, are scaled to a common range (e.g., between 0 and 1) using min-max normalization. Z-Score Normalization: For some features, Z-score normalization is used to ensure that the features have a mean of 0 and a standard deviation of 1, which is important for models that rely on gradient-based optimization.

4.4 Time Feature Engineering: For columns related to cooking times, such as preparation and cooking durations, the following preprocessing steps are performed:

Time Conversion: Time-related features are converted into a standard format, such as splitting the total time into preparation time and cooking time. Handling Missing Time Values: Recipes with missing or implausible cooking times are flagged, removed, or imputed with average cooking times for similar recipes.

4.5 Data Splitting: Once all preprocessing tasks are complete, the dataset is split into training and testing sets to ensure model generalizability. The splitting process ensures:

Balanced Classes: Care is taken to ensure that the recipe categories (e.g., cuisine types, recipe tags) are evenly distributed between the training and testing datasets. Stratified Sampling: In cases of imbalanced classes, stratified sampling is used to ensure that the distribution of the target variable is similar across both training and testing datasets.

5. Evaluation

The evaluation of the GPT-3.5 Turbo model for recipe generation based on nutritional constraints showed promising results across several metrics. The model achieved an ingredient accuracy of 0.875 and a calorie accuracy of 0.923, indicating its ability to generate recipes that closely match the input ingredients and calorie specifications. The average calorie deviation was 25.4 kcal (± 15.2 kcal), demonstrating reasonable adherence to the target calorie values. In terms of structural correctness, the model showed strong performance with title extraction accuracy at 0.958, ingredients extraction accuracy at 0.912, and directions extraction accuracy at 0.894. These results suggest that the model can generate well-structured and nutritionally constrained recipes. For instance, when tasked with generating a recipe for chicken with a 400 kcal target, the model produced a recipe titled ”Lemon Rosemary Grilled Chicken” that closely matched the input ingredients and stayed within a 390 kcal range. Overall, the evaluation demonstrates that the model can generate accurate, structured, and nutritionally aligned recipes, making it effective for recipe generation tasks under specified constraints.

References

[1] Nature Scientific Reports, “Title of the Article,” *Nature Scientific Reports*, 2024. [Online]. Available: https://www.nature.com/articles/s41598-024-65438-x. [Accessed: Oct. 24, 2024].

[2] arXiv Preprint, “Title of the Paper,” *arXiv*, vol. 2312.08592v1, 2024. [Online]. Available: https://arxiv.org/html/2312.08592v1. [Accessed: Oct. 24, 2024].

[3] S. Author, “Personalized Health-Aware Recipe Recommendation: An Ensemble Topic Modeling Based Approach,” *ResearchGate*, 2019. [Online]. Available: https://www.researchgate.net/publication/334867462_Personalized_Health-Aware_Recipe_Recommendation_An_Ensemble_Topic_Modeling_Based_Approach. [Accessed: Oct. 24, 2024].