Università degli Studi di Milano
FACOLTÀ DI SCIENZE E TECNOLOGIE

DIPARTIMENTO DI INFORMATICA
GIOVANNI DEGLI ANTONI

CORSO DI LAUREA MAGISTRALE IN
INFORMATICA

REPORT FINALE
NATURAL LANGUAGE PROCESSING

Report Finale by:
Daniele Boerio
Matr. No. 09627A

Academic Year 2023-2024

Contents

Contents

Chapter 1

Introduction

The following is a report of the project developed for Natural Language Processing course at the University of Studies of Milan, during the academic year 2023-2024. The project aims to develop a system capable of generating culinary recipes based on a provided list of ingredients. This system will use large language models (LLMs) and statistical methods to compose recipes which are the best possible. By utilizing pre-trained LLMs fine-tuned on a recipe dataset, the system learns to combine user-provided ingredients with appropriate cooking methods. The generated recipes will be evaluated against real recipes to assess their quality, utilizing user ratings as a benchmark.

Chapter 2

Research question and methodology

2.1 Goals of the project

The goal of this project is to develop a recipe generation system using a pre-trained Large Language Model (LLM). Specifically, we will use GPT-2, fine-tuned on a culinary dataset, to accomplish this task. The model will be capable of generating complete recipes based solely on a provided list of ingredients, producing detailed step-by-step instructions for each recipe. In addition, the system will evaluate the quality of the generated recipes by employing a statistical approach, comparing them to real recipes, and selecting the best one based on this analysis.

2.2 Proposed Approach

The project proposes to employ a combination of:

- Model: a pre-trained large language model (GPT2) fine-tuned on a recipe dataset
- Statistical methods: methods to analyze the correlation between ingredients, steps and ratings to select the best recipe generated from our LLM model
- Recipe generation: a model capable to generate a recipe integrating user-provided ingredients

2.3 Formal Definition of the Problem

Given a set of ingredients, the system aims to generate recipes that are not only feasible in terms of ingredient compatibility but also likely to be well-received based on existing recipe ratings.

Chapter 3

Experimental results

3.1 Dataset Used for Experiments

For this project, I chose to use a comprehensive dataset of culinary recipes that includes various cuisines, styles, and user reviews. This dataset, titled "Food.com Recipes and Interactions," is available on Kaggle. It contains over 180,000 recipes and more than 700,000 recipe reviews, spanning 18 years of user interactions and contributions on Food.com. The downloaded folder includes numerous files suitable for different types of projects. For my purposes, I opted to work with the raw dataset, specifically importing the following two files into my project:

- "RAW_recipes": This file contains detailed information about each recipe, including the recipe name, preparation time (in minutes), number of steps, step-by-step instructions, description, number of ingredients, and a list of ingredients.

- "RAW_interactions": This file provides data on user interactions with the recipes, including user IDs, recipe IDs, dates of interaction, ratings, and the text of user reviews.

3.2 Metrics for Evaluation

To assess the quality of the generated recipes, I employed a correlation matrix as the evaluation metric. The correlation matrix is constructed by analyzing the relationships between the ingredients, cooking steps, and the corresponding reviews or ratings provided by users. Each recipe is represented by its list of ingredients and cooking steps, and these elements are compared across the entire dataset. By calculating these correlations, the matrix highlights how specific combinations of ingredients and steps are associated with higher or lower ratings. This analysis helps identify the components that are more likely to contribute to a successful recipe, allowing us to select the most promising generated recipes based on patterns observed in real user feedback.

3.3 Experimental Methodology

The following steps outline the methodology I used to complete this project:

- Merging Datasets: Since both datasets contain crucial information, I merged them into a single Pandas DataFrame using the (id, recipe_id) as the unique key.

- Clear Dataset: Given that external datasets often contain null or blank values, I removed all rows with null, blank, or duplicate entries to ensure data quality.

- Listing ingredients and step: The ingredients and steps columns were originally represented as strings. I split these strings into lists of individual ingredients and steps to facilitate better management and processing.

- Formatting ingredients and step: Formatting the recipes was a critical step because the GPT model needs to be trained in the same format that will be used for generating text. The final format for the recipes was structured as "Ingredients: ing_1,ing_2,ing_3, ... Steps: 1.$step_1$, 2.$step_2$, 3.$step_3$, ..." This consistency ensures that the model performs effectively.

- Tokenizer: Tokenization involves breaking down text into smaller units called tokens, which can be words, phrases, or even individual characters. This step is crucial in natural language processing (NLP) as it converts text into a format that a machine learning model can understand and process. Tokenization simplifies and structures the text data, making it easier for the model to learn patterns and make accurate predictions.

- GPT Model Training: The GPT model was trained to understand and generate human-like text by being exposed to large amounts of data. During training, the model learns to predict the next word in a sentence based on the previous words, gradually improving its ability to generate coherent and contextually relevant text. This involves adjusting the model's parameters to minimize prediction errors, enabling it to effectively capture the patterns, nuances, and structures of language.

- Preprocessing And Correlation Matrix: I preprocessed the ingredient and step lists by removing punctuation and unnecessary spaces before tokenizing them. These steps were crucial for cleaning the data to achieve the best possible results. After tokenization, I combined the ingredients, steps, and ratings into a single DataFrame and then generated the correlation matrix.

- Recipe Generator: I used the GPT-2 model to generate five recipes based on a given input or "prompt." The model predicts the next word or token in a sequence, continuing this process until a specified length is reached or a stopping condition is met. I fine-tuned the generator using four key parameters to guide the output:
  - max_new_tokens: Maximum number of new tokens to generate.
  - temperature: Controls the randomness of predictions. Lower values make the text more deterministic, while higher values introduce more variety.
  - top_k: Limits model choices to the k best results, improving the quality of generations.
  - top_p: Sets the cumulative probability threshold for selecting tokens. Higher values increase variety and creativity by including less probable tokens, and vice versa.
- Preprocessing And Final Recipes: Similar to the Correlation Matrix step, I took the generated recipes, divided them into ingredients and steps, and then preprocessed them by removing punctuation and unnecessary spaces. The processed recipes were then evaluated using the correlation matrix. The recipe with the highest correlation value was selected as the best recipe and presented to the final user.

Chapter 4

Concluding remarks

4.1 Experimental Results

Rating	Real Recipe	Generated Recipe
0	-242645	-1438
1	-207138	-1
2	-150	0
3	-77809	0
4	-76454	-16
5	-7137	-1

This table presents the results of applying the correlation matrix to the combination of Ingredients and Steps. At first glance, the values suggest that the generated recipes outperform the real ones. However, this apparent superiority may be misleading. The issue lies in the generation of Ingredient/Step pairs that do not exist in the real dataset and therefore do not significantly impact the correlation matrix. As a result, many steps in the generated recipes do not influence the overall value of the recipe. In contrast, real recipes include specific Ingredient/Step pairs that may lower the recipe's value. Additionally, the value of real recipes was calculated based on the highest-rated recipe rather than the average rating. This approach explains why a real recipe with a 2-star rating might be considered more valuable than a 5-star recipe in this context.

4.2 Conclusions

The experimental results indicate that the correlation matrix is not an effective method for evaluating generated recipes against real ones, though it might be useful for comparing recipes within a generated set. A more robust approach could involve using a classification model based on neural networks, which could learn to independently rate recipes using real data and serve as a better evaluation tool. However, the current dataset, which is skewed with many 3- to 5-star ratings and fewer low ratings, might not be ideal for training such a neural network, as it could lead to biased results. In the future, using a more balanced dataset could significantly improve the effectiveness of this approach and enhance the overall project.