Hurry up, I'm hungry!

Stefano Corizzato[1]

[1]Dipartimento di Informatica "Giovanni degli Antoni", Università degli Studi di Milano, Via Giovanni Celoria - 18, Milano, 20133, Italia.

Contributing authors: stefano.corizzato@studenti.unimi.it;

Abstract

The project aims at developing a system that generates culinary recipes based on a given list of ingredients. The system will use large language models (LLMs) and statistical methods to compose ingredients and cooking methods. The generated recipes will be evaluated by comparing them to the ratings of similar real recipes.

The idea is to exploit a pre-trained language model like GPT-4, BERT or others. This model can be fine-tuned on the recipe dataset to enhance its understanding of culinary contexts. The system will then be capable of generating recipes by combining user-provided ingredients with cooking methods learned during training. To refine these recipes, statistical methods will be employed. By analyzing correlations between ingredients, cooking methods, and recipe ratings, the system can identify combinations that are typically well-received. This analysis will guide the generation process, making it more likely to produce high-quality recipes. The generated recipes will be evaluated against real recipes. This comparison will involve using existing user ratings of similar real recipes as a benchmark. By assessing the generated recipes against these ratings, the system's performance can be objectively measured.

Keywords: NLP, Text Generation, Recipes

1 Introduction

In this project, a pre-trained large language model (LLM) was fine-tuned to generate text. Specifically, the goal is to generate a recipe text using a list of ingredients as input.

After selecting an appropriate database for the purpose, three models were generated starting from an existing language model. The best model obtained was finally evaluated using both the original recipes and their rating as a benchmark.

The language used for this project is Python, the various models were fine-tuned starting from models available on the Hugging Face Transformers framework and using PyTorch a machine learning framework.

2 Database

The data was taken from Alvin "Food.com - Recipes and Reviews"[1] dataset, made available on the Keggle website. The dataset consists of approximately 520,000 recipes and 1.400.000 reviews. The data is organized in two folders "recipes" and "reviews", for this project only the "recipes" folder was used.

After downloading the data, it was inserted into a dataframe, where it was analyzed: the recipes dataset has 28 different fields that concern information about each recipe such as cooking times, servings, ingredients, nutrition, instructions, and more. Of this data only the following fields were kept:

- `RecipeId`: Recipe ID
- `RecipeIngredientParts`: Array of ingredients
- `AggregatedRating`: Average rating, i.e. average of votes
- `ReviewCount`: Number of votes
- `RecipeInstructions`: Instructions, the actual recipe

The resulting dataframe was filtered again selecting only the recipes with at least 1 review, and with the fields regarding the ingredients and the instructions to follow not null, thus obtaining a dataframe of 167,302 recipes

The fields containing the ingredients and the instructions were transformed from arrays to textual data for convenience during training. Finally, two datasets were created, of 10 thousand and 20 thousand recipes, selecting the best recipes for average score and number of votes.

The choice of dimensions is due to the computing power required for the fine-tuning of these models, and the choice of best recipes is to exclude any incomplete or incorrect recipes. This way the worst recipe in our dataset has 10 votes and average score of 5.

3 Training

The starting model chosen is GPT-2 (Generative Pre-trained Transformer 2), a model that uses causal attention (each token in the sequence can only refer to previous tokens, not future ones) and therefore preferable for text generation compared to models that use bidirectional attention[1].

GPT-2 pretrained models are available in different sizes: in this project `"gpt2"` with 110M parameters and `"gpt2-medium"` with 345M parameters were used.

[1] https://www.kaggle.com/datasets/irkaal/foodcom-recipes-and-reviews

3.1 Model 1

The first fine-tuned model was trained with the dataset of 10,000 recipes and using "gpt2", which is the lightest model.

The tokenizer used is that of GPT-2, which has a vocabulary of 50,260 tokens with 4 last indices reserved for special tokens such as `<|startoftext|>`, `<|endoftext|>`, `<|unknown|>` and `<|pad|>`. The special token `<|unknown|>` is used for words not in the GPT-2 vocabulary and `<|pad|>` is used at the end of short sequences to make sequences within a batch have the same length.

Each recipe was transformed by concatenating the ingredients and instructions, and adding the tokens `<|startoftext|>` and `<|endoftext|>` at the beginning and end of the text.

The recipes were then tokenized and divided into two datasets, one for training (90%) and one for validation (10%).The model was fine-tuned with the following parameters:

- batch size = 8
- epochs = 3
- optimizer = AdamW
- learning rate = 2e-5

The final validation loss is 1.65, below are the results divided by epoch:

Table 1 First model results

Epoch	Training Loss	Validation Loss	Training Time	Validation Time
1	2.333	1.701	0:07:42	0:00:17
2	1.710	1.663	0:07:58	0:00:17
3	1.666	1.651	0:07:58	0:00:17

3.2 Model 2

To obtain better results for the validation loss, for the second fine-tuned model gpt2-medium was used as a basis, keeping the dataset size constant, at 10.000 recipes.

The process used to train the model, as well as the hyperparameters are the same as those used in the previous model. Due to the increased number of parameters of gpt2-medium, the training time has increased considerably.

The final validation loss is 1.45, below are the results divided by epoch:

Table 2 Second model results

Epoch	Training Loss	Validation Loss	Training Time	Validation Time
1	2.011	1.485	0:21:30	0:00:43
2	1.490	1.455	0:21:35	0:00:43
3	1.446	1.447	0:21:35	0:00:43

3.3 Model 3

For the last model tested, a dataset of double the size (20,000 recipes) compared to the previous one was used, and "gpt2" was used as the starting model. As in the previous case, the procedures and hyperparameters remained unchanged compared to those of the models already seen.

The final validation loss is 1.55, below are the results divided by epoch:

Table 3 Third model results

Epoch	Training Loss	Validation Loss	Training Time	Validation Time
1	2.021	1.606	0:15:41	0:00:32
2	1.640	1.566	0:14:40	0:00:32
3	1.597	1.553	0:15:41	0:00:32

The best result was obtained in the second case, therefore in order to evaluate the recipes created, only the second model was taken into consideration.

4 Evaluation

Recipes are generated with a maximum length of 180 characters, considering only the 50 most probable words (top_k) and among these those with a cumulative probability of 85% (top_p = 0.85).

In order to evaluate the generated texts, user ratings on real recipes were used. The main idea is to select pairs of recipes with high and low scores that share the same ingredients. Then generate a recipe with the same ingredients and then measure the distance between the generated recipe and the 2 real recipes.

Furthermore, the BLEU algorithm was used on the fine-tuned model and on the original model to have an estimate of the goodness of the created recipes.

4.1 User Rating

First, 2 datasets of recipes were created that the model had never seen. One for low-scoring recipes, selecting recipes with an average score lower than 2.5, and one for high-scoring recipes, selecting recipes with a score of 5 and less than 10 votes, effectively excluding the recipes used for training.

The ingredients of the first 100 recipes in the low-scoring dataset were then compared with the recipes in the high-scoring dataset. Only recipe pairs with more than 75% of ingredients in common were considered, finding 14 "low-scoring" "high-scoring" pairs. These pairs were further filtered by eliminating recipes with less than 3 ingredients, resulting in a final set of 11 recipe pairs that share the same ingredients.

Using the common ingredients, a recipe was generated for each pair using the fine-tuned model. The generated recipe and the original recipes were embedded using the all-mpnet-base-v2 model, a sentence encoder based on the MPNet (Masked and Permuted Pre-training for Language Understanding) family of models developed by Microsoft.

Using PyTorch, the cosine similarity between the recipe embeddings was calculated, to check whether the generated recipe was closer to the high or low recipe. The average score obtained between the high and generated recipes is 0.733, while the average score between the generated and low recipes is 0.671.

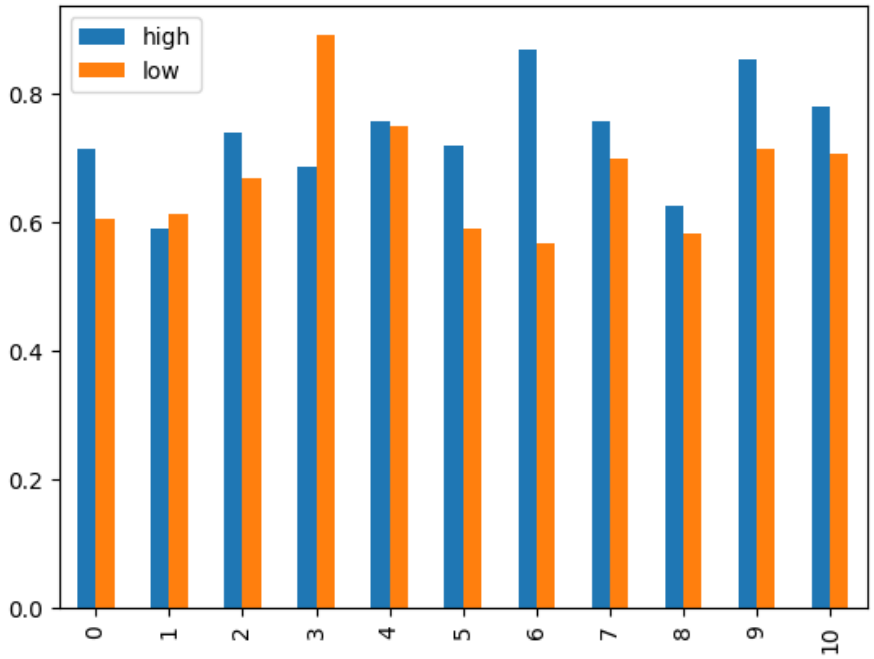

Fig. 1 Cosine similarity score between the generated recipe and high and low scoring recipes with the same ingredients.

In 9 out of 11 cases the generated recipe is more similar to the recipe with the high score.

4.2 BLEU

Using the BLEU algorithm via nltk's sentence_bleu it is possible to calculate how similar a recipe generated, starting from a list of ingredients of a real recipe, is to the original recipe. BLEU uses exact correspondences of n-grams and is usually used to verify the quality of the translation of texts[2]. The value obtained by measuring the fine-tuned model is 8.035e-232, while the original one is 8.55e-232. Such low values are too insignificant to indicate a real improvement.

For this reason, METEOR was also calculated, a metric for evaluating the generated text, which takes into account synonyms, stemming and word order. The values obtained are 0.212 for the trained model and 0.149 for the original model. Although both are low values, the difference between the models is more appreciable.

5 Conclusion

The cosine similarity results show that, in most cases, the generated recipes are more similar to high-scoring recipes, indicating that the fine-tuned model is successful in creating higher-quality outputs. The BLEU and METEOR scores reveal an improvement with the fine-tuned model compared to the original model, although the values are still very low. These results suggest that while the fine-tuned model shows some promise in generating higher quality recipes than the original model, there is still room for further improvement in terms of accuracy and fluency.

References

[1] Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, 3rd edn. (2025). Chap. 11. Online manuscript released January 12, 2025. https://web.stanford.edu/~jurafsky/slp3/

[2] Rao, D., McMahan, B.: Natural Language Processing with PyTorch: Build Intelligent Language Applications Using Deep Learning, (2019). Chap. 8. https://books.google.it/books?id=3m69tAEACAAJ