Deep Learning Final Report

Louis Etienne Kusno
Tsinghua University

David Qi
Tsinghua University

Spencer Xi Zhou
Tsinghua University

Abstract

As personalized nutrition becomes increasingly important, especially for individuals managing conditions like diabetes, there is a growing need for recipe generation systems that account for health-related constraints such as glycemic index. Existing models often prioritize fluency and creativity over nutritional suitability. In this work, we fine-tune a pre-trained GPT-2 model on a curated subset of the RecipeNLG dataset containing low-glycemic-index recipes to explore controlled generation of healthier meals. Our dataset is carefully filtered for nutritional quality and manageable token length, allowing the model to learn structured, coherent, and health-compliant outputs. Using conditional generation based on user-provided ingredients, our system produces recipes that are both nutritionally appropriate and syntactically fluent. While qualitative evaluation suggests promising results, challenges remain in ingredient pairing and generalization to rare items. We discuss possible solutions, including semantic constraints and nutritional validation mechanisms. This work highlights the potential of language models in health-aware recipe generation and sets the foundation for more robust, user-driven, and nutritionally validated applications.

1. Introduction

Home cooks and diet conscious individuals frequently rely on automated meal planning tools, yet these systems often fail to incorporate essential nutritional constraints such as low glycemic index (GI), low sugar, and balanced macronutrients. This gap is particularly problematic for users managing conditions like diabetes or seeking to maintain stable blood glucose levels.

In this work, we present a personalized recipe generation framework that transforms a large language model into a health aware assistant. Rather than computing exact nutritional values—which requires detailed ingredient data and complex modeling—we adopt a practical strategy: we fine tune GPT-2 on a carefully filtered subset of the RecipeNLG corpus containing only low GI recipes. This allows the model to internalize patterns of health compliant meal composition while retaining linguistic fluency and structural coherence.

By conditioning on user provided ingredient lists, our system generates recipes that align with dietary guidelines without demanding expert knowledge from end users. We demonstrate that a standard transformer decoder, once exposed exclusively to nutritious recipe examples, can output meal instructions that are both plausible in the kitchen and consistent with health goals. In doing so, we bridge the gap between cutting edge natural language generation and real world nutritional planning.

2. Related Work

Early research in automatic recipe generation centered around multimodal approaches, such as Recipe1M+ [6], which aligned food images and textual recipes through joint embedding spaces. More recently, the focus has shifted to purely text based generation using large scale language models. Kusupati et al. [4] introduced RecipeGPT, a transforme based model trained on a diverse corpus of cooking instructions, demonstrating the feasibility of generating syntactically and semantically coherent recipes.

Health aware recipe generation presents additional challenges. Yang et al. [9] explored optimizing nutritional content by substituting ingredients while preserving recipe structure. Similarly, Nutri-bot [7] integrated user specific dietary profiles and nutritional guidelines to personalize meal recommendations.

In the broader area of diet planning, machine learning has enabled the development of models that optimize meals based on caloric intake, macronutrient balance, and glycemic index [2]. These systems typically leverage optimization techniques such as reinforcement learning or graph based search to recommend nutritionally balanced plans.

Large language models (LLMs) have also begun to play a prominent role in health related applications. For example, recent work by Lee et al. [5] investigates how LLMs can be used for personalized dietary recommendations, capitalizing on their strength in contextual language understanding and integration of domain knowledge.

Building on these developments, our approach fine tunes GPT-2 on a filtered subset of the RecipeNLG dataset with explicit health related constraints. This bridges flexible text generation with nutritional awareness, enabling the model to produce recipes aligned with user provided ingredients and health goals.

3. Approach

3.1. Model Architecture

Our recipe generation system is built upon GPT-2, a transformer based causal language model developed by OpenAI. Specifically, we utilize the pre trained `mbien/recipenlg` model, which is a fine tuned variant of GPT-2 medium (176 million parameters) tailored for generating semi structured culinary text. This model originates from the RecipeNLG project, which aims to address the structured generation of cooking recipes using language models.

GPT-2 employs a transformer decoder architecture with self attention mechanisms that allow it to model long range dependencies in text. As a causal language model, it generates text autoregressively—predicting the next token based on all previously generated tokens. This property is especially beneficial for generating recipes, which require coherent sequencing from ingredients to step by step instructions.

The architecture is well uited to structured generation tasks, as it can learn and replicate the patterns typical of recipe text, such as consistent formatting of ingredients, procedural clarity in instructions, and contextual compatibility among ingredients. The model's training on a large corpus of recipes enables it to internalize domain specific language and conventions, making it an effective base for further fine-tuning on health-constrained recipe data.

4. Experiments

4.1. Dataset Preparation

To support personalized, nutritionally aware recipe generation, we defined a "healthy" meal using the following nutritional criteria:

Low Glycemic Index (GI) Ingredients must have a GI $\leq$ 55

Low Carbohydrates Total carbohydrates $\leq$ 45–60,g per meal (per adult dietary guidelines)

Low Added Sugars Added sugars contribute less than 10% of total daily caloric intake

Balanced Macronutrients Recipes contain adequate protein and healthy fats, and minimal saturated fats

Portion Control Servings are appropriately sized to meet average caloric needs

However, we only had time to implement the first criterion, focusing on glycemic index. This choice was made to simplify the dataset preparation process while still allowing for meaningful recipe generation. The glycemic index is a well established measure of how quickly foods raise blood sugar levels, making it a useful proxy for overall meal healthiness.

To construct a dataset aligned with these criteria, I first extracted glycemic index values for base ingredients using data from [3]. The RecipeNLG dataset [1] was then filtered to retain only those recipes where the majority of ingredients had a glycemic index $\leq$ 55. This process reduced the dataset from 2,231,142 recipes (`full_dataset.csv`) to 3,419 recipes (`h_recipes_50pct.csv`).

For efficiency, only recipes with $\leq$ 512 tokens were kept, further narrowing the dataset to 3,398 entries (`h_recipes_50pct_token_max512.csv`). This greatly improved training speed and memory usage, particularly given the use of a single 12.7GB GPU (Google Colab environment).

Additional data cleaning was necessary. Recipes with ingredients not listed in the glycemic index were removed. Ingredient duplication (e.g., both "apple" and "apples") was resolved by adding basic plural checks.

Each recipe was then tagged according to the format required by the RecipeNLG tokenizer:

`<RECIPE_START>`...`<RECIPE_END>` Encloses the entire recipe

`<INPUT_START>`...`<NEXT_INPUT>`...`<INPUT_END>` Tags ingredient names extracted via Named Entity Recognition (NER)

`<INGR_START>`...`<NEXT_INGR>`...`<INGR_END>` Tags full ingredient lines

`<INSTR_START>`...`<NEXT_INSTR>`...`<INSTR_END>` Tags individual cooking steps

`<TITLE_START>`...`<TITLE_END>` Tags the recipe title

Within each section, items were delimited using `<NEXT_INPUT>`, `<NEXT_INGR>`, or `<NEXT_INSTR>` tags as appropriate. The final preprocessed dataset was saved as `h_recipes_50pct_token_max512_tagged.csv`.

4.2. Model Training

The model was fine tuned using the Hugging Face `Trainer` API [8] with the following hyperparameters:

- **Learning Rate:** 5e-5

- **Batch Size:** 8 (per device, for both training and evaluation)
- **Epochs:** 10
- **Optimizer:** AdamW (betas = (0.9, 0.999), epsilon = 1e-8)
- **Scheduler:** Linear learning rate decay with 50 warmup steps
- **Mixed Precision:** Enabled (fp16=True) for faster training on GPU
- **Random Seed:** 42
- **Checkpointing:** Every 50 steps, retaining the two most recent checkpoints
- **Logging:** Every 100 steps (loss and learning rate output to TensorBoard)

For the complete training script, see `notebooks/retrain.ipynb`.

4.3. Model Run Parameters

Prompt Format. To generate recipes, the model was given prompts in the following structure:

> Give me a healthy recipe using the following ingredients (you don't have to use all of them): <RECIPE_START> <INPUT_START> chocolate <NEXT_INPUT> pickles <NEXT_INPUT> beef <NEXT_INPUT> pear <INPUT_END>

(Whitespace was minimized in practice to reduce token count.)

Tokenization Parameters.

- `truncation=True`
- `max_length=512`
- `return_attention_mask=True`

Generation Parameters.

- `max_length=512`
- `num_beams=2` or `5` (beam search)
- `no_repeat_ngram_size=2`
- `early_stopping=True`
- `eos_token_id` set to the token ID for <RECIPE_END>

The `no_repeat_ngram_size=2` setting helped prevent repetitive ingredient or instruction phrases, a common issue in neural text generation. Beam search (with `num_beams=2` or `5`) allowed the model to explore multiple candidate sequences, improving recipe quality.

4.4. Results

Below are selected examples generated using the same base ingredients:

Melon and Strawberry Ice Cream

Base Ingredients: milk, cream, strawberry, melon, honey

Ingredients:

- 2 cups milk
- 1 cup half-and-half cream
- 12 cup strawberry, hulled and quartered
- 8 ounces cantaloupe, peeled, seeded, and cut into 1/2-inch cubes
- 2 tablespoons honey

Instructions:

1. In a medium saucepan, combine the milk and cream.
2. Bring to a boil over medium-high heat, then reduce the heat to low and simmer, stirring occasionally, until the mixture is thickened, about 5 minutes (do not let it boil).
3. Stir in the strawberries and melon; cover and refrigerate until chilled, at least 2 hours.

Melon-Jello Mold

Base Ingredients: milk, cream, strawberry, melon, honey

Ingredients:

- 1 1/2 cups milk
- 1 cup half-and-half cream
- 10 ounces strawberry Jell-O gelatin dessert
- 8 ounces cantaloupe melon, cubed
- 2 tablespoons honey

Instructions:

1. In a medium saucepan, combine milk and cream. Cook and stir over medium heat until mixture comes to a boil. Remove from heat; stir in gelatin until dissolved. Chill until slightly thickened, about 1 hour.
2. Stir in melons and honey. Pour into a 6-cup mold coated with cooking spray. Refrigerate until set, at least 4 hours. Unmold onto a serving platter.

Adding an incompatible ingredient like `salmon` yields a more creative, yet plausible recipe:

Grilled Salmon With Strawberry Sauce

Base Ingredients: milk, cream, strawberry, melon, honey, salmon

Ingredients:

- 1 1/2 cups milk
- 1 cup half-and-half cream
- 12 ounces strawberry, hulled and quartered
- 8 ounces cantaloupes, peeled, seeded and cubed
- 2 tablespoons honey
- 4 (6-ounce) salmon fillets

Instructions:

1. In a medium saucepan, combine the milk, half and half, strawberries and melon.
2. Bring to a boil over medium-high heat, then reduce the heat to medium and simmer until the fruit is soft, about 10 minutes. Stir in the honey and season with salt and pepper. Set aside to cool to room temperature, stirring occasionally to prevent discolouring. Place the salmon in a resealable plastic bag and pour the cooled fruit mixture over the top. Seal the bag, pressing out as much air as possible. Refrigerate for at least 4 hours or overnight, turning once. Remove from the refrigerator 30 minutes before grilling. Preheat an outdoor grill for medium heat (350° to 400°).
3. Grill salmon, skin side down, until just cooked through, 4 to 6 minutes, depending on thickness. Serve with the strawberry sauce.

However, ingredient pairing is not always optimal. For example:

Dried Beef And Pear Salad

Base Ingredients: chocolate, pickles, beef, pear

Ingredients:

- 1 (4 oz.) pkg. chocolate or butterscotch pudding mix
- 1/2 c. chopped pickles
- 2 oz. jar dried beef, chopped
- peel of 1/4 medium pear

Instructions:

1. Mix pudding and pickle in a bowl.
2. Add beef and pear. Chill.

5. Conclusion

This work demonstrates that an existing large language model (LLM), specifically GPT-2 medium, can be fine tuned on a health filtered subset of the RecipeNLG dataset to generate nutritionally conscious and coherent recipes. The selected dataset ensured adherence to health guidelines such as low glycemic index and balanced macronutrient content.

The training process involved meticulous dataset preparation, including tagging of recipe components and ingredient tokenization, enabling the model to learn both the structure and content of healthy recipes. While the current evaluation was primarily qualitative, future work should incorporate objective metrics such as BLEU scores, perplexity, or human evaluation surveys to assess fluency, relevance, and nutritional alignment more rigorously.

5.1. Discussion and Future Work

Model Limitations. Despite encouraging results, our fine tuned GPT-2 medium model exhibits several limitations. First, the healthy subset contains only 3,398 recipes, which is relatively small for a model of this size and raises concerns about overfitting. This limited data may explain repetitive phrasing—such as "combine all ingredients and serve"—and the generation of generic instructions. Second, the model sometimes proposes implausible ingredient pairings, for example combining chocolate pudding with pickles and beef, indicating insufficient semantic or culinary grounding. To mitigate these issues, one could employ semantic clustering to group ingredients by taste or culinary role, apply co occurrence penalties during beam search to disfavor rare or incompatible combinations, and introduce post processing filters—either handcrafted rules or learned classifiers—to eliminate any remaining implausible recipes before presentation to the user.

Nutritional Validation. While our dataset level filtering ensures that ingredients have known glycemic indices and that at least half of them are low GI, further validation of generated recipes is necessary. In future work, we plan to include a comparative analysis of glycemic index values for generated recipes versus those in the original corpus, presented in tabular form. Additionally, we will conduct a detailed macronutrient breakdown—reporting carbohydrates, sugars, proteins, and fats—for each recipe to provide greater dietary transparency and to confirm that the model's outputs truly meet health oriented nutrition goals.

Broader Context. For practical deployment, user interaction and system scalability must be carefully considered. A natural next step is to integrate the model into mobile or web based applications that accept ingredient inputs via manual text entry, barcode scanning, or even voice commands. Such interfaces could allow users to specify personalized dietary goals—like "low sugar" or "gluten free"—and receive recipe suggestions in real time. Moreover, because the model currently struggles with rare or underrepresented ingredients (e.g., jackfruit or quinoa), we recommend expanding the training corpus with external recipe sources, incorporating food specific embeddings (such as those from FlavorGraph) to relate unfamiliar ingredients to known ones, and exploring retrieval augmented generation techniques to bring in external nutritional or culinary context at inference time.

Community Feedback. Feedback from our in class presentation highlighted several promising directions for future enhancements. First, expanding the dataset to include a wider variety of global cuisines and dietary styles could improve model generalization. Second, integrating more stringent dietary filters—such as vegan or gluten free restrictions—would cater to specific user needs. Third, implementing a user feedback loop, where individuals rate generated recipes, could enable continuous, data driven refinement of the model. Finally, exploring advanced learning techniques, such as reinforcement learning from human feedback (RLHF), may further improve generation quality by directly optimizing for user satisfaction and nutritional accuracy.

5.2. Project Contributions

- **Method Design:** David & Spencer
- **Experiments:** Louis & David
- **Data Analysis:** Louis & Spencer
- **Report Writing:** Louis & Spencer

- **Poster Creation:** David & Louis & Spencer

References

[1] Tuan Bien, Kyunghyun Kim, and Sang-goo Kim. Recipenlg: A cooking recipes dataset for semi-structured text generation. In *LREC*, pages 7185–7192, 2020. 2

[2] Chao Fang, Dong Li, Ming Zhou, and et al. A personalized diet recommendation system based on health metrics. *IJCAI*, pages 234–240, 2020. 1

[3] FoodStruct. Glycemic index chart. Online; accessed 2024-06-09, 2024. https://foodstruct.com/glycemic-index-chart. 2

[4] Helena H. Lee, Ke Shu, Palakorn Achananuparp, Philips Kokoh Prasetyo, Yue Liu, Ee-Peng Lim, and Lav R. Varshney. Recipegpt: Generative pretraining based cooking recipe generation and evaluation system. In *The Web Conference 2020 Companion of the World Wide Web Conference (WWW 2020)*, pages 181–184. Association for Computing Machinery, 2020. Demo track. 1

[5] Jun Lee, Soo Kim, Yuna Park, and et al. Llms in healthcare: Applications for diet and lifestyle guidance. *NeurIPS Workshop*, 2023. 1

[6] Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(8):1932–1946, 2019. 13 million recipes & images. 1

[7] Priya Mehta, Rahul Singh, and Nisha Gupta. Nutri-bot: Personalized recipe recommendations using nutritional guidelines. In *IEEE International Conf. on Biomedical and Health Informatics (BHI)*, 2023. 1

[8] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. HuggingFace's transformers: State-of-the-art natural language processing. In *EMNLP Systems Demonstrations*, pages 38–45, 2019. 2

[9] Yixuan Yang, Minh Nguyen, Byron C. Wallace, and Xinghua Lu. Nutritional recipe generation using multi-objective reinforcement learning. In *Findings of the Assoc. for Computational Linguistics – EMNLP*, pages 471–483, 2022. 1