Generative Recipes Using GPT-2

Chloe Valcourt and Isaac Chan

CS 505

ABSTRACT

This project addresses the challenge of using AI to generate coherent recipes, where we leveraged the GPT-2 model for its natural language processing capabilities. After custom data preparation, we fine-tuned the GPT-2 model with a curated subset of the RecipeNLG dataset with a focus on recipe structure, ingredient coherence, and direction coherence. The model initially struggled with producing reasonable and sensible recipes using the recipe title as an input, hence we changed our approach to use ingredients and recipe title as inputs for optimal results. Statistical evaluation was completed through BLEU (Bilingual Evaluation Understudy) scores and word cloud analysis, revealing a moderate level of success; however, it is evident through the discrepancies in ingredient representation and statistical results that there is a need for future refinement of the model to yield better results.

INTRODUCTION

With the boundaries of artificial intelligence (AI) technology being continually expanded through extensive research and technological advancements, the emergence of text-based AI-generated content seemed to lead the way with public interest, suggesting a new technological era in the form of an AI revolution. Upon discovering our shared passion for cooking and homemade food as university students, we wanted to build a natural language processing (NLP) model that generates recipes. We noticed that there was a notable gap in the use of AI tools like NLP models for innovative recipe creation, so in this project, we strived to bridge this gap.

MODEL CHOICE

Our first step was to choose an appropriate model to fine tune, to which we came to the conclusion of using GPT-2. Two main factors were taken into account behind this choice: GPT-2 is a proven model with advanced NLP capabilities and it is also easily accessible with ample documentation online. We found a hugging face dataset, RecipeNLG, that had over two million

recipes. Given the size of the dataset and limitations of google colab, we decided not to train our model on the entire dataset.

When looking for a resource that was similar to the scope of recipe generation and fine-tuning GPT-2, we were able to find an article that fine-tuned a GPT-2 model to generate song lyrics. We decided that this was the perfect starting point for our project and the direction and code in the article provided a skeleton for our work.

We first observed how GPT-2 performed without further training on our recipe datasets. It seemed to have a fair performance; however, it would occasionally produce strange ingredients and inconsistent formats. For instance, one of the recipes suggested cooking a chocolate cake on a baking sheet, which is typically used for cookies rather than cakes. The same recipe also suggested the use of peppers in a chocolate cake, a highly unusual ingredient. Additionally, it did not include rising ingredients like baking powder or baking soda. We concluded that this did not comply with the structure of a typical recipe, where ingredients are listed, then directions are provided step-by-step. Our goal was to improve GPT-2's models to have a more structured and sensible output for recipes.

DATA PREPARATION AND TRAINING

We extracted the data and concatenated the ingredients and directions for each recipe into a single formatted string; this was crucial in maintaining the structure and sequence of recipes. We then created a list of 5000 of these recipes as the basis of our training, with a max length of 500 characters. We saved 1000 recipes for testing work. We wanted our recipes to be structured as follows:

```
Recipe: Jewell Ball'S Chicken
Ingredients:
1 small jar chipped beef, cut up
4 boned chicken breasts
1 can cream of mushroom soup
1 carton sour cream

Directions:
- Place chipped beef on bottom of baking dish.
- Place chicken on top of beef.
- Mix soup and cream together; pour over chicken. Bake, uncovered, at 275° for 3 hours.
```

Following that we created a custom dataset class called CookBook, which we tailored to handle the recipe data for the GPT-2 model's training. The following are complete with the creation of this class: tokenizer initialization, encoding and padding recipes, converting

sequences to tensors (required format for model input), and dataset truncation (where we set the maximum recipe length to 500 tokens).

OBSERVATIONAL ANALYSIS OF FINAL MODEL PERFORMANCE

We first experimented with the model by seeing how the model completed recipes it was not trained on, using the recipe name as the sole input. Ideally, we wanted the generated recipes to follow the intended physical structure without overfitting by copying the training exactly, or underfitting by generating nonsense. Note that we generated each recipe with a temperature of 0.8 to prevent overfitting.

Below is a generated example:

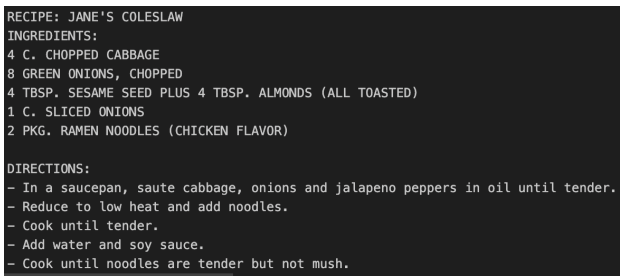
```
Generated Recipe
RECIPE: RICE SALAD
Ingredients:
2 lb. cooked rice
1 can cream of mushroom soup
1 can cream of chicken soup
1 can cream of celery soup
1 can cream of celery soup
1/2 c. melted butter
1/2 c. salad oil
1/2 c. vinegar
1/2 c. sugar
Directions:
- Cook rice. Drain off excess water.
- While hot, mix rice; add onion soup, celery soup and salad oil.
- Bring to boil.
- Turn heat down and simmer for 20 to 25 minutes or until tender.
```

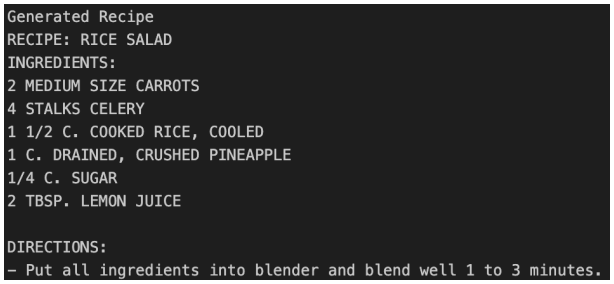
```
RECIPE: EGGPLANT AND TOMATO CASSEROLE
Ingredients:
4 large tomatoes
1/2 c. celery, chopped
3 medium onions, chopped
1 large onion, chopped
1/2 c. vinegar
1 Tbsp. sugar
1 tsp. salt
1/2 tsp. pepper
1 Tbsp. butter
1 Tbsp. oil

Directions:
- Kelvin:
- Boil tomatoes for about 20 minutes.
- Add celery, onions, carrots, tomatoes and tomato soup ingredients.
- Mix well.
- Bake in a preheated 350° oven for 1 1/2 hours.
- Top with cream cheese frosting and sprinkle with lemon juice.
```

It can be observed that when prompted with just the title of the recipe, the model has trouble outputting sensible recipes; however, it strongly follows the intended structure well. The measurements for each ingredient were also sensible, not outputting over measurements like one cup of salt for example, and most of the ingredients were mentioned in the directions. We wanted to improve the model through the following: preventing the redundant ingredients in the ingredients section, including ingredients from the title, including all ingredients in the directions, and distinguishing between sweet and savory dishes.

Due to the large disparity between the expectation of results and actual results, we decided to prompt the model with both the name and ingredients, observing the results of the directions. Below are two results of generated directions:





Here our model is prompted with the text in all caps and the lower case portion is what the model completed. It is evident from human analysis of our outputs that with such a small training set (due to Colab's limits) our model had difficulty extracting meaning from recipe titles alone. Typically, these are creative and subjective to the recipe writer. It would be difficult for the model to figure out that Jane's Coleslaw is very similar to John and everyone else's coleslaw on such a small training size.

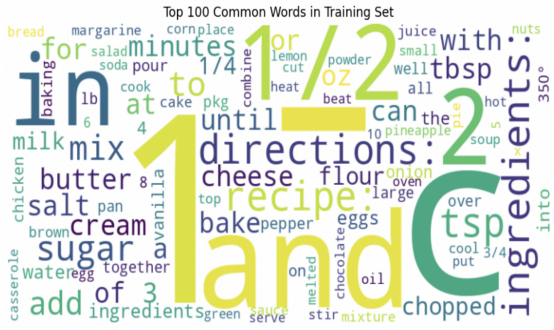
However, we were able to see that our model was able to learn from something more standardized and concrete to each recipe, the ingredients. If trained and fine tuned on a larger dataset this could be incredibly applicable to home usage. Say you have a list of ingredients that you are limited to and would like the model to output a new recipe for those ingredients, this model could be very useful and inventive in that case.

STATISTICAL ANALYSIS OF FINAL MODEL PERFORMANCE

Upon research, we decided to statistically analyze our model's performance using BLEU score and word cloud analysis. A BLEU (Bilingual Evaluation Understudy) score is a number [0,1] that measures the similarity between our outputted results and a reference of other recipes. It does so by measuring the overlap of n-grams (sequence of n items in the text) between our results and the reference texts. We fed the model the recipe titles and ingredients from our test set and analyzed its completion of the recipes in comparison to the real recipes on average.

We found that when generating BLEU scores for individual examples, we yielded incredibly low results, with some scores being well below 0.0001. This prompted us to instead retrieve a BLEU score from the average of 100 generated outputs. From this, we were able to yield a score of 0.0112, which is not ideal, but far better than the original. However, due to computational resources with the GPU in Colab and space restrictions with our datasets, it is understandable that we yielded a score that was this low, hence we decided that human analysis was more appropriate in evaluating our model's performance.

As mentioned above, we also utilized word clouds to find disparities between the training and testing set, allowing us to visualize the differences in vocabulary. By comparing these word clouds, we could identify the prevalent words in the training data that were underrepresented in the generated recipes. Below are the two word clouds:



We observed from above that specific ingredients like cheese and chicken are prominent in the training set, but less so in the testing set. This suggests a potential bias towards certain ingredients in the training data. It is clear that there are variances in the representation of cooking styles and methods, and there is evidence of overemphasis on terms like cheese and chicken in the dataset, which might explain the low BLEU score. However, we were pretty impressed with the consistency of words besides the savory and sweet word biases we noticed between test and training.

FUTURE OF THE PROJECT

It is evident that the yielded results were not perfect, albeit satisfactory considering the limitations in computational resources. From our results, we decided that we want to improve the model's ability to maintain coherence between recipe titles, ingredients, and directions, which could be achieved by the potential use of context-aware models that can comprehend the relationship between different parts of the recipe better. Additionally, we could benefit by adding additional features into the training process, labeling each recipe with features like general flavor, whether the dish is served hot or cold, and cooking method. We especially considered that future implementation may have to have a cleaning process for the titles of recipes to be less creative, for example rather than "Jane's Coleslaw" or "Crazy Mac n Cheese" a more standardized naming of "Coleslaw" and "Mac n Cheese", respectively. This would allow the model to learn more across different recipes of the same thing. We want to continue to refine this model to yield the optimal results that we want, which would also require potential allocation of

financial resources into expanded computational resources, which would enhance our model training process. Our ultimate goal for the future would be to develop a model that could generate sensible recipes based on recipe name only.

Google Drive Link to dataset and code:
https://drive.google.com/drive/folders/1oXJW5WuxlvwfKZ_NraYYm3aOZWIAC_mU?usp=drive_link

STATEMENT OF WORK:

Both Isaac and Chloe contributed to the development of this project. Chloe took the lead in developing the initial model and results, and both of us contributed to refine and improve the model. Both Isaac and Chloe contributed to the analysis, with Chloe's focus being the human analysis, while Isaac focused on the statistical analysis. Isaac managed the project deadlines to ensure that deadlines were met, and took the lead in the report writing process, though both members worked extensively on the report.

Signature (Chloe Valcourt): 

Signature (Isaac Chan):