RECIPE SUGGESTION SYSTEM BASED ON AVAILABLE INGREDIENTS

Bikram Chand
Tagliatela College of Engineering
University Of New Haven
West Haven, Connecticut, United States Of America
bchan9@unh.newhaven.edu.

Diwas Chaulagain
Tagliatela College of Engineering
University Of New Haven
West Haven, Connecticut, United States Of America
dchau5@unh.newhaven.edu

Priyanka Kalyani
Tagliatela College of Engineering
University Of New Haven
West Haven, Connecticut, United States Of America
pkaly2@unh.newhaven.edu

Abstract— **We face this challenge of what to eat, despite having the necessary ingredients. It may not feel like that big of a problem but considering there are billions of people all over the world, the wastage of food is in tons. The report paper is about introducing a Recipe Suggestion System developed to solve the problem of food wastage and to plan the recipes using some NLP tasks. The system uses a BART model that is derived from Hugging Face that is a transformer model, to determine the similarity. The data is initially split into training, testing and validation and fine-tuning is also performed. The main aim of the entire project is to provide the end user with a proven solution to the problem that we stated. The solution is provided at the end in the form of a bot that suggests to us what to cook based upon the ingredients we provide. Python script was used for deployment and the exact match and F1 scores were checked. This document provides insights into the program and how things have been implemented. We will also investigate the results and dissect different areas of potential future exploration.**

I. INTRODUCTION

In today's health-conscious world, many individuals are striving to make informed dietary choices by focusing on healthy ingredients. However, translating these ingredients into creative and suitable recipe titles can often be a challenge. Whether it's selecting ingredients from the pantry or shopping for groceries, people frequently find themselves uncertain about what to prepare. This dilemma is not limited to beginners; even professional chefs occasionally struggle to conceptualize recipes based on available ingredients. Consequently, the ability to generate recipe titles based on ingredients becomes a valuable tool for simplifying the cooking process and inspiring meal ideas. For this reason, we have proposed a system that takes ingredients as input and generates a suitable recipe title. To achieve this, we used a dataset from Kaggle and implemented a model based on Facebook's BART architecture using the Hugging Face Transformers library. Additionally, we developed a Telegram bot that allows users to input ingredients and instantly receive a recipe title. This system simplifies meal planning by providing quick and creative recipe suggestions, making cooking more convenient and enjoyable.

II. RELATED WORK

To get a good insight into what we need to do, we should take a look at what is already happening in this field. We start by looking at the paper titled 'RecipeBowl: A Cooking Recommender for Ingredients and Recipes Using Set Transformer' from 2021. This paper talks about RecipeBowl, an application which provides the end user with some necessary ingredients that are missing as well as what recipe can be made with the provided ingredients and the suggested ingredients. Compared to other shallow applications and papers, this paper dives deep into the data part of the product and uses almost 400,000 recipes to train and almost 50,000 for validation and testing each. From this paper we can say that the more the data, the better the model but we also should keep in mind the computational power of our personal computer as well as the time it would take to run the model. The concept that we used from the model was the 80-10-10 distribution of the training, testing and validation data.

Another paper that we researched was the 'Recipe Recommendation System using Machine Learning Models' that was published in 2019. They used tf-idf to calculate the cosine similarity between two ingredients. This way they paired the different ingredients with high similarity and suggested a recipe. "Each ingredient is treated as a single document and flavor components are treated as terms of documents. Tf-idf score is calculated for all ingredients on the basis of this matrix." (Maheshwari, 2019). Although working properly, what we felt was this approach lacked the depth of the neural networks and maybe the ability to handle complex situations and the increase in data size may cause some flexibility issues as well.

In 'A Cooking Recipe Recommendation System with Visual Recognition of Food Ingredients', the main goal was to recommend a few recipes that are video based, based on the ingredients that was detected by the model in spaces like homes, restaurants and grocery stores. Bag-of-features and linear-kernel SVM were used, and the recognition rate was also high but the object detection feature felt unnecessary and unwarranted as it only adds complications to the system and the text based systems are more preferred. Similar approach was also used in ‘RecipeIS—Recipe Recommendation System Based on Recognition of Food Ingredients’. Here, CNN was used to detect the ingredients followed by a three-layered approach with ResNet model. The data was broken down into training, testing and validation as well. The model accuracy was determined by how accurately it could predict the ingredient, instead of how close the recipe is to the ingredient.

III. METHODOLOGY

A. Dataset

The dataset used for this study is the "RecipeNLG" dataset from Kaggle, which contains 2,231,142 entries, including recipe titles, ingredients, steps, sources, and Named Entity Recognition (NER) tags. The reason behind choosing this dataset was it has enough as well as includes the exact columns we need, such as recipe titles and ingredients, making it perfect for training the model.

B. Data Preprocessing

Since the original dataset was quite large, we selected a subset of 25,000 entries to make the training process manageable due to resource constraints. We then dropped unnecessary columns and removed duplicate recipe titles, as having the same title multiple times could confuse the model. Other redundant entries were also removed to ensure the data quality. After this cleaning process, the dataset was reduced to 15,870 entries. Since the dataset was already in a relatively clean state, we did not perform additional preprocessing steps like removing stop words, sentece segmentation etc. as they were not necessary for our project.

C. Feature Engineering

For feature engineering, we utilized the BART tokenizer from the Hugging Face Transformers library to preprocess the text data. The tokenizer was used to convert the recipe titles and ingredients into tokenized format, making it compatible for input into the model. After tokenizing the data, we verified the results by checking if the tokenized text and the original text matched, ensuring that the tokenization process was accurate and did not lose any essential information. This step was crucial to maintaining the integrity of the input data for the model training.

D. Modeling

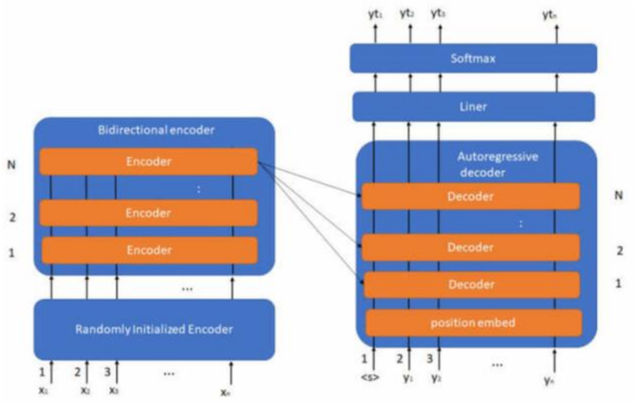
For the model, we used the BART (Bidirectional and Auto-Regressive Transformers) architecture, which is well-suited for sequence-to-sequence tasks like generating recipe titles based on ingredient lists. The dataset was first tokenized using the BartTokenizer from Hugging Face, transforming the text into tokens that the model can process. Following tokenization, the data was split into three subsets: training (80%), validation (10%), and testing (10%). This split ensures that the model is trained on a large portion of the data while being validated and tested on separate subsets to assess its performance.

To train the model, we used PyTorch's DataLoader to handle batching and shuffling of the training data, which helps the model generalize better. The model was initialized with the pre-trained facebook/bart-base version and moved to the available device (GPU or CPU). During training, we used the AdamW optimizer with a learning rate of 5e-5. The model was trained for 3 epochs, where each epoch consisted of a training phase followed by a validation phase. The training phase involved computing the loss, performing backpropagation, and updating the model weights, while the validation phase checked how well the model was generalizing to unseen data.

Throughout training, we monitored both the training and validation losses to ensure the model was learning effectively without overfitting. The model architecture itself consists of multiple layers of bidirectional transformers, with the encoder processing the input (ingredients) and the decoder generating the output (recipe titles). The final output layer of the decoder generates the predicted recipe title based on the tokenized ingredients.

Model Architecture

BART is a sequence-to-sequence model that combines a bidirectional encoder (like BERT) and a left-to-right decoder (like GPT). It is pretrained using a combination of text shuffling and in-filling tasks, where parts of the text are replaced with a mask token. This architecture allows BART to excel in text generation tasks, as well as comprehension tasks, outperforming models like RoBERTa on benchmarks such as GLUE and SQuAD. It achieves state-of-the-art results in areas like abstractive summarization, question answering, and dialogue generation (HuggingFace, 2019).

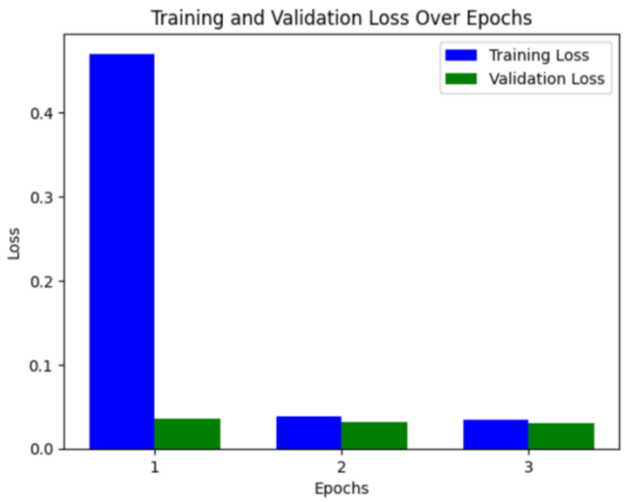


IV. RESULTS AND EVALUATION

The model was trained for three epochs, and both training and validation losses were recorded after each epoch. During the training process, the loss steadily decreased, showing that the model was effectively learning from the data. The validation loss also decreased consistently, indicating good generalization to unseen data.

Here are the results:

- **Epoch 1**: Training Loss = 0.4699, Validation Loss = 0.0359
- **Epoch 2**: Training Loss = 0.0386, Validation Loss = 0.0322
- **Epoch 3**: Training Loss = 0.0337, Validation Loss = 0.0310



This consistent decrease in both training and validation loss demonstrates that the model was fine-tuned effectively.

Additionally, the model's performance was evaluated using Exact Match and F1 Score metrics, achieving the following scores:

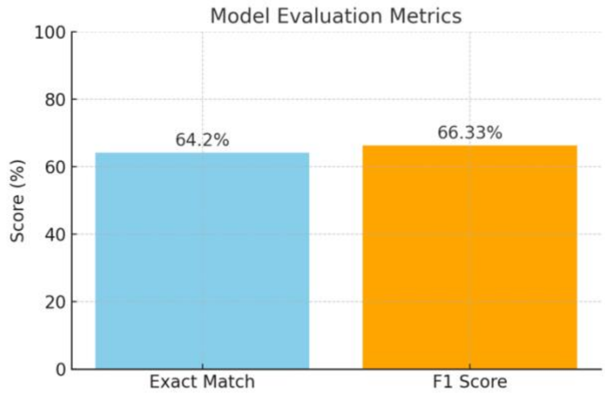


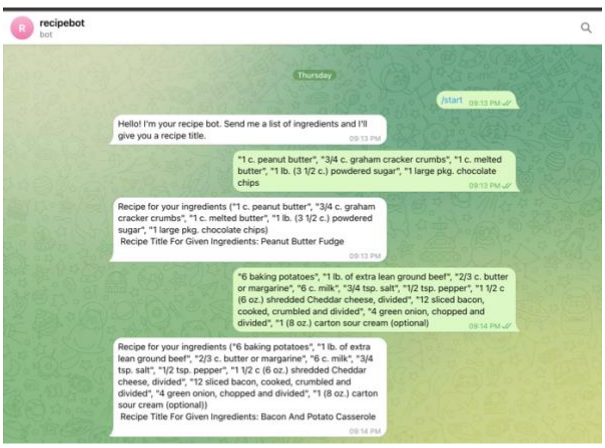
Fig. 1. Exact match and F1 Score Comparison

Lastly, here are some examples of recipe title generated by the model based on the input ingredients.

```
 'labels': torch.tensor(item['labels'], dtype=torch.long)
Sample 1 - Input: ["1 c. finely ground walnuts", "1/2 c. finely ground sunflower seeds", "1/2
Sample 1 - Generated Recipe: Brown Rice And Rice
--------------------------------------------------------------------------------
Sample 2 - Input: ["1 (9-inch) pie crust, baked and cooled", "1 c. sugar", "3 Tbsp. cornstarch
Sample 2 - Generated Recipe: Lemon Pie
--------------------------------------------------------------------------------
Sample 3 - Input: ["1/4 c. lemon juice", "2 tsp. cornstarch", "1/4 tsp. salt", "1/2 lb. each:
Sample 3 - Generated Recipe: Shrimp And Vegetable Soup
--------------------------------------------------------------------------------
Sample 4 - Input: ["8 large garlic cloves, pressed", "2 Tbsp. fresh chopped rosemary or 2 tsp.
Sample 4 - Generated Recipe: Beef And Lamb Casserole
--------------------------------------------------------------------------------
Sample 5 - Input: ["2 c. white sugar", "1 c. brown sugar", "1 c. evaporated milk", "1/2 c. ole
Sample 5 - Generated Recipe: Peanut Butter Fudge
--------------------------------------------------------------------------------
Sample 6 - Input: ["2 c. milk", "4 eggs", "1/3 c. oleo", "1/2 c. flour", "3/4 c. sugar", "1 c.
Sample 6 - Generated Recipe: Coconut Pudding
--------------------------------------------------------------------------------
Sample 7 - Input: ["2 c. low calorie cranberry juice", "2 c. natural apple juice", "1 1/2 c. o
Sample 7 - Generated Recipe: Cranberry Punch
--------------------------------------------------------------------------------
Sample 8 - Input: ["3 2/3 c. (1 lb. box) powdered sugar, sifted", "1/2 c. cocoa", "1/2 c. butt
Sample 8 - Generated Recipe: Chocolate Brownies
--------------------------------------------------------------------------------
Sample 9 - Input: ["1 c. cracker crumbs", "1 1/2 c. milk", "3 eggs, slightly beaten", "2 (6 1/
Sample 9 - Generated Recipe: Tuna Casserole
--------------------------------------------------------------------------------
Sample 10 - Input: ["4 (8 oz.) pkg. cream cheese, at room temperature", "1 1/2 tsp. vanilla",
Sample 10 - Generated Recipe: Cream Cheese Ball
--------------------------------------------------------------------------------
```

V. Deployment

The deployment of the fine-tuned BART model was implemented using a Telegram bot name **RecipeBot**, allowing users to interact with the model by providing a list of ingredients and receiving recipe title suggestions.



V. DISCUSSION AND FUTURE WORK

The model performed well overall, with relatively low training and validation losses. However, the Exact Match score of 64.20 and the F1 score of 66.33 suggest there is still room for improvement. These scores indicate that the model sometimes struggles to generate the exact recipe title. Possible improvements could include training on a larger and more diverse dataset, tuning hyperparameters, and applying data augmentation techniques.

In the future, we could enhance the system by allowing users to upload images of ingredients, which the model could use to generate recipe titles. This would make the application more interactive and practical. Expanding the model to support multiple languages and generating complete recipes, including instructions, is another potential improvement.

VI. Conclusion

In conclusion, this paper implemented a BART-based pretrained model and fine-tuned it to generate recipe titles from the provided ingredients.. The model achieved an Exact Match score of 64.20 and an F1 score of 66.33, indicating that the model performs well but could be improved. With better resources, such as higher GPU configurations, the model could have been trained for more epochs on a larger dataset, which might have improved its performance. Currently, the system generates recipe titles from text-based ingredient lists, but in the future, we aim to enhance it by enabling it to generate recipes from images of ingredients. This could make the system more interactive and practical for users.

VII. REFERENCES

Gim (2021). *RecipeBowl: A Cooking Recommender for Ingredients and Recipes Using Set Transformer*. Source - IEEE Xplore. https://ieeexplore.ieee.org/document/9570315

Maheshwari, Chourey (2019). *Recipe Recommendation System using Machine Learning Models*. Source - International Research Journal of Engineering and Technology (IRJET).

https://irjet .net/archives/V6/i9/IRJET-V6I946.pdf

Yanai . *A Cooking Recipe Recommendation System with Visual Recognition of Food Ingredients*. Source – Online Journal. https://online-journals.org/index.php/i-jim/article/view/3623/3116

Miguel (2023). *RecipeIS—Recipe Recommendation System Based on Recognition of Food Ingredients*. Source - MDPI. https://www.mdpi.com/2076-3417/13/13/7880

Vivek (2017). *Machine Learning Based Food Recipe Recommendation System*. Source - Springer. https://link.springer.com/chapter/10.1007/978-981-10-5146-3_2

Sharma (2021). *Food Recipe Recommendation System*. Source - IRJET. https://www.irjet.net/archives/V8/i4/IRJET-V8I4398.pdf