# 

# NU_RGB_Notched-N_wordmark_RB

# A Generative AI Tool for Recipe Creation Based on Available Household Ingredients

**Reducing Food Waste and Enhancing Meal Planning with Ingredient-Aware AI Recipes**

IE7374

Group Members:

Akshit Sharma

Anna Grondolsky

Allison Selwah

Karthik Veluchamy Sarguru

Table of Contents

[A Generative AI Tool for Recipe Creation Based on Available Household Ingredients 1](#_Toc1700357637)

[Project Title & Description 2](#_Toc717119646)

[Final Topic Area 3](#_Toc566830118)

[Dataset Description 3](#_Toc919290018)

[Model Selection 3](#_Toc1681524410)

[Research Questions 3](#_Toc1039480748)

[Plan of Action 4](#_Toc707016905)

[Team Contribution 4](#_Toc250297463)

## Project Title & Description

**A Generative AI Tool for Recipe Creation Based on Available Household Ingredients**

A generative AI-powered tool designed to help users create meal ideas from ingredients they already have at home. By inputting a list of grocery items, users receive one or more recipe suggestions that incorporate those items effectively. The system leverages natural language processing and generative models to craft complete and coherent recipes including ingredients lists, instructions, and preparation times. This tool aims to reduce food waste, save time, and inspire home cooking using artificial intelligence.

## **Final Topic Area**

We will be working in the field of Natural Language Processing (NLP) for our final project. Our chosen topic involves generating simple, easy-to-make recipes based on natural language ingredient inputs. The rationale behind this choice stems from our interest in how machines can understand and generate human language, particularly in a creative and practical domain like cooking.

This project emphasizes sequence generation and semantic understanding, both of which are core NLP tasks we’ve explored in this course. By converting a list of ingredients into coherent and useful recipe steps, we aim to apply NLP techniques such as tokenization, embedding, and potentially transformer-based models. The challenge and excitement lie in teaching a model to grasp not just words, but the contextual and procedural knowledge they imply—something we believe showcases the real-world potential of NLP.

## Dataset Description

For our project, we are using the Recipe1M+ dataset, developed by MIT CSAIL and available on Kaggle ([RecipeNLG Dataset](https://www.kaggle.com/datasets/saldenisov/recipenlg)). This dataset contains approximately one million recipes, each consisting of a title, a list of ingredients, and detailed cooking instructions. While the full dataset includes images, we will be focusing exclusively on the textual components, as our work centers on NLP.

The data is stored in JSON format, which makes it easy to parse and manipulate for training language models. Each recipe is cleanly structured, which is especially helpful for tasks like sequence generation. We are particularly interested in the ingredients-to-instructions relationship, as our goal is to generate simple recipes based on user-provided ingredient lists—mimicking the real-world scenario of deciding what to cook based on what's in the fridge.

One of the key advantages of this dataset is its scale and diversity, which will allow us to train or fine-tune a generative model to produce coherent and relevant cooking steps. We also plan to simulate user inputs by extracting subsets of ingredients from existing recipes, allowing the model to learn how to generate plausible instructions from partial ingredient lists.

Overall, Recipe1M+ offers the structure, volume, and content richness we need to support our project’s goals in natural language generation and semantic understanding.

## Model Selection

For our project, we plan to use a GPT-based generative language model, specifically **GPT-2,** which is a transformer-based decoder architecture developed by OpenAI.

GPT-2 is well-suited for our task because it excels at natural language generation and can produce coherent, contextually relevant text based on a given input prompt. Our research focuses on generating step-by-step recipe instructions from a list of ingredients, a classic sequence-to-sequence problem where understanding the semantics of the input and generating structured output is crucial. GPT-2's architecture, which is designed for left-to-right language modeling, aligns perfectly with the task of generating fluent, ordered recipe instructions.

We will be using a pre-trained version of GPT-2 (likely via the Hugging Face Transformers library) and fine-tuning it on the Recipe1M+ dataset. This approach is recommended because:

* It leverages existing linguistic knowledge already learned by GPT-2, reducing training time and computational cost.
* Fine-tuning allows us to adapt the model to the cooking domain, learning patterns specific to ingredients and cooking instructions.

Using a pretrained model also enables us to focus on the design of input representations (e.g., formatting the ingredient list as a prompt) and the evaluation of output quality, rather than on training a large model from scratch.

Overall, GPT-2 offers the right balance of performance, accessibility, and flexibility for generating natural-sounding recipes from textual inputs.

## Research Questions

1. How effective is a fine-tuned language model (e.g., GPT-2) in generating complete, coherent recipes from a random list of grocery items?
   * This evaluates the model’s ability to structure logical and readable cooking instructions from unordered and variable ingredient inputs.
2. Can transformer-based models generalize to novel ingredient combinations not seen during training?
   * We want to test the model's creativity and semantic understanding when faced with unfamiliar inputs—mimicking real-world use cases where pantry contents are unpredictable.
3. How do different prompt structures (e.g., including cooking context, cuisine hints, or just raw ingredient lists) affect the quality and specificity of generated recipes?
   * By comparing prompt variations, we aim to identify which formats guide the model toward more useful and coherent outputs.
4. Does incorporating ingredient embeddings or clustering improve semantic matching and substitution (e.g., butter vs. margarine)?
   * This investigates whether embedding-based techniques can help the model recognize functionally similar ingredients, enhancing its flexibility and output quality.

## Plan of Action

1. Data Preprocessing

We will begin by cleaning and organizing the Recipe1M+ dataset:

* Extract and normalize ingredient lists and corresponding instructions
* Remove incomplete or poorly formatted entries
* Standardize units and ingredient naming to reduce variability
* Simulate user inputs by creating subsets of ingredients (e.g., 3-7 items per list) to reflect real-world grocery scenarios

1. Model Implementation

* Use a pretrained GPT-2 model from the Hugging Face Transformers library.
* Fine-tune the model on the cleaned dataset, using structured prompts to encourage coherent recipe generation.
* Experiment with prompt engineering, including variations like adding context or cuisine tags to guide outputs.

1. Experimental Design & Evaluation

We will conduct multiple experiments to test different configurations:

* **Baseline**: Raw ingredient list as prompt.
* **Context-enhanced**: Prompts with cuisine type or desired dish name.
* **Embedding-aided**: Using pretrained ingredient embeddings to improve semantic matching.

For evaluation, we will use:

* **BLEU** or **ROUGE** scores to measure similarity to ground truth instructions.
* **Perplexity** to assess model fluency.
* **Human evaluation** (if time allows) to assess readability and usefulness.

1. Analysis Techniques

* Compare performance across prompt types and ingredient subsets.
* Use clustering to analyze ingredient embedding quality.
* Visualize attention weights or embedding distances to understand model behavior.

## Team Contribution

Our team consists of five defined roles, with responsibilities distributed based on expertise and availability. Two members will lead the technical development, while others contribute to evaluation, testing, and documentation. All team members share responsibility for communication, planning, and final deliverables.

Akshit:

* Fine-tune the GPT-2 model using Hugging Face Transformers.
* Implement and experiment with prompt engineering techniques.
* Oversee model training and optimization processes.

Allison:

* Clean and preprocess the Recipe1M+ dataset.
* Normalize ingredients and create structured prompts.
* Generate training and test samples based on real-world input simulation.

Anna:

* Design and apply evaluation metrics (BLEU, ROUGE, perplexity).
* Conduct error analysis and identify model weaknesses.
* Help interpret both quantitative and qualitative results.

Karthik:

* Document all preprocessing, model, and evaluation steps.
* Assist with user simulation testing and prompt response validation.
* Lead preparation of the final report and visual materials for presentation.

All Members:

* Participate in weekly check-ins and milestone reviews.
* Review and test intermediate model outputs.
* Contribute to discussion of findings, final recommendations, and course-aligned reflections.
* Ensure code reproducibility, shared repositories, and version control practices are followed.