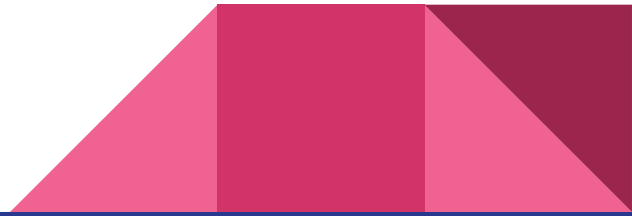
What’s That Recipe: Designing and Implementing a Recipe Recommendation System

By: Bryan Pritsker, Neil Godbole, Khalifa Afane

Overview

- Main Goal: Use the GPT2 model and NLP techniques to create a recipe recommendation system based on user prompts
- Overall Design:
  - Get recipe data using Tasty API
  - Use pre trained capabilities of Hugging Face's GPT2 model
  - Apply NLP techniques (semantic similarity, prompt engineering…) to process user inputs
  - Generate an appropriate output recipe
  - Make something yummy!

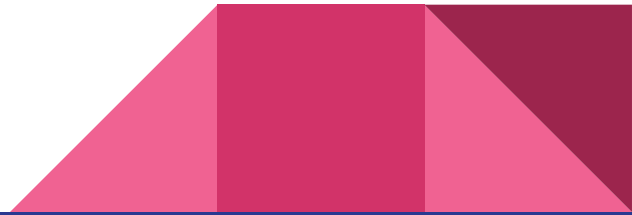


Collecting and Loading Data

- All recipe data collected from Tasty API
  - Free to use
  - Data includes information such as cuisine type, description, ingredients, instructions, etc.
- Current model pulls 775 recipes from 20 different cuisines (Thai, Chinese, Mexican etc.)
- Recipes saved into an excel file for reproducibility
- Model loads the file into a pandas dataframe
  - Combined and standardizes tags for cuisine labels
  - Explodes rows to handle multiple tags
  - Attaches cuisine to recipe description
  - Filter and normalize tags (Replace “North American” with “American”)

Data Preprocessing for GPT2

- Using GPT2 Model from Hugging Face
  - Loaded locally through HF's transformers library
  - Model is pre-trained on a large amount of text data → we did not train the model
- Model takes user input to generate output
  - Used nltk for processing: lowercasing, lemmatizing, tokenizing, removing stopwords
- Cleans/normalizes inputs → semantic similarity with recipe descriptions



Generating Recommendations: TF-IDF

- TF-IDF Similarity:
  - Term Frequency- Inverse Document Frequency
- User input and all recipe descriptions from dataframe converted into a text corpus
- Text converted into vector based on word frequencies
- Cosine similarity measured b/w user input vector and recipe description vector → ranking based on similarity

Generating Recommendations: NER

- Use NER (named entity recognition) in conjunction with TF-IF
  - using Spacy library
- Identifies and classifies key words in text
  - Model looks for 120 common food words: “beef”, “pork”,etc.
  - Look for cuisine types
- TF-IF calculates similarity score → model gives boost to recipes with extracted entities that are relevant
- Recipes are ranked by boosted score→ highest score recipe chosen

Model Example

- Example input: “Give me a mexican recipe with chicken”
- Example Recipe Descriptions:
  - 1) “This is a mexican taco recipe with chicken”
  - 2) “This is a mexican beef stew”
- Vector table:

Word	User Input	Recipe 1 (Taco)	Recipe 2 (Stew)
Mexican	High	High	High
Chicken	High	High	Low

- Similarity Score: Recipe 1→ 2 High:0.85, Recipe 2→ 1 high, 1 low: 0.75
- Boosting keywords: “Mexican”+0.12, “Chicken”+0.10
- Recipe 1 Score: 0.85+0.12(“Mexican”)+0.10(“Chicken”)= 1.07
- Recipe 2 Score: 0.75+ 0.12(“Mexican”) = 0.87
- **Recipe 1 chosen**

Prompt Engineering

- Used prompt engineering to help model results
- Used few shot learning→ gave model example inputs and matching recipe description
  - I want a healthy vegetarian meal” → “A fresh and vibrant quinoa salad with cherry tomatoes, cucumbers, and a tangy lemon dressing”
- Help lead model to recipe generation without fine tuning
- Leveraging the pre-trained capabilities of GPT2

Results

```
Recipe Similarity Scores Before Boosting:
Rank 1: Vietnamese Coffee - Score: 0.1795
Rank 2: One-Pot Mexican Quinoa - Score: 0.1593
Rank 3: Thai Iced Tea Caffeinated Smoothie - Score: 0.1541
Rank 4: Crispy Chili Beef - Score: 0.1522
Rank 5: Crispy Chili Beef - Score: 0.1522
Entities found in the user input: ['mexican', 'mexican', 'beef']


Top 5 Recipe Similarity Scores after Boosting:
Rank 1: Southwestern Taco Salad - Score: 0.3468
Rank 2: Cauliflower “Meat” Nachos - Score: 0.3468
Rank 3: Southwest Stuffed Acorn Squash - Score: 0.3463
Rank 4: One-Pot Taco Spaghetti - Score: 0.3386
Rank 5: Taco-Stuffed Peppers - Score: 0.3247
User Input: Give me a mexican beef recipe
Recommended Recipe: Southwestern Taco Salad
This recipe was chosen because it closely matches your input.
```

- Results for prompt “Give me an Italian recipe to impress my friends”
- Tested on 50+ inputs
- Generally gives relevant recipe but lacks in precision
- Picks up on cuisine tag very well, other keywords do not match as well

Conclusion

- Created a recipe recommendation LLM using GPT2 model
- Used several NLP techniques: TF-IF, NER, Prompt Eng, NLTK preprocessing
- Future Work:
  - Find a better way to evaluate model performance
  - Experiment with other pre-trained models: Llama3
  - Try embeddings for semantic similarity
  - Expand dataset