Intelligent Interactive Systems - Final Project Milestone 4

Matan Solomon 322853334
Nitay Suissa 209446640

Jan 2022

1 Project Summary

1.1 The problem

The problem we aim to solve is that people sometimes find it hard to translate their feelings about what they want to eat, to actual recipe. They know they want, for example, something sweet, cold, and quick, but they need suggestions of relevant options.

1.2 Our solution

Our solution is a retrieval system that takes from the user the properties we believe she will be able to give, and retrieve relevant recipes. The intelligence and challenge are expressed in the translation between the 'intuitive', or even 'abstract' description of the user, to specific food.

1.3 Unique approach

In our approach, we would like to project recipes into semantic vector space by generating embeddings for the recipes. The challenge is to create embeddings that will be able to capture these abstract properties. We would like to achieve this by creating predictors for the abstract properties and enrich them using graph neural networks.

2 The Interface

A prototype of our system is given in figure **??**.
While designing our interface, we made sure to follow several key-principle to make it as clear and interactive as possible. First our interface is consist of two parts. The right part is where the user fill his abstract concepts that he wants the recipe to meet. This part was designed to be minimal as possible but also very informative and helpful. The left part is where the recipes are retrieved. This two parts are designed to be next to each other and with the same size and dimensions for emphasizing the cooperation of the user and the system.

As for Microsoft AI Design guidlines, we decided that our system won't have many option that it can operate, but it has only one functionality that is to present the most relevant recipes to the user's ask. That way we made clear what the system can do (principle 1).Moreover, we decided to make our system live, in a way that the results change every time the user changes the input, such that it easy to invoke our system (principle 8). Furthermore, we added a button for retrieving other recipe for the inputted properties. These ideas increase the interactiveness, and support efficient correction (principle 9).

3 The Algorithm

3.1 Data Preprocessing

Our raw data was three data structure from the dataset Recipes1M+ that include more than 1 Million recipes.

1. First data structure was the original text of each ingredient and instruction for every recipe.

2. The second data structure save for every recipe its all ingredients outputted from an ingredients detector, built by the authors, that got as an input the raw text of the ingredients.

3. The third data structure is nutrition details for every recipe that had amounts for all the ingredients. This data structure include about 50,000 recipes.

We filter only recipes that has nutrition facts (for the health section) and that the ingredient detector in the second data structure succeeded to identify correctly all the ingredients in the recipe. After this filtering we ended up with a bit less then 50,000 recipes. Then we join all these data structures into one data frame called "data_df.pkl"

After arranging the data in a dataframe we needed to build the initial representation of the recipes. We will now cover how we create each type of features.

1. Flavors features - at first, we wanted to classify each ingredient to some flavor. However there was a bit less than 6000 unique ingredients. So we filter the least frequent ingredients i.e. the ingredients that appear in the smallest number of recipes. We filter those ingredients such that the total number of appearances of the ingredients that left is more than 95% of all the appearances of ingredients in all the recipes. Now we left with only about 1300 ingredients. We classify these into one of 5 classes, 'sweet', 'salty', 'sour', 'spicy', and 'other' using ChatGPT.
   After classify the ingredients, the flavors feature were the partial part of each flavor from the total flavors of one recipe.

2. Temperature feature - here we used ChatGPT for listing words that indicate high temperature (e.g. oven, bake), and words that indicate low temperature (e.g. refrigerate , freeze) both relate to food making and kitchen. After that for every recipe, we count the number of instructions that include cold words, hot words, and those who didn't include any of them. Then we normalize the quantities such that they will sum to one.

3. Time feature - Here we sum the time if there is in the instructions (e.g. bake it for 30 minutes) and we also add the number of instruction times five, as kind of heuristic that the average instruction last about 5 minutes. We normalize this feature using min-max normalization across all recipe.

4. Health feature - Here we took the unhealthy nutrition facts, energy, sugar, salt, and fat. For each recipe we average their index in each of the nutrition facts ,where index 1 is the recipe that has the least of the nutrition fact, using the harmonic average. Then we normalize this feature using min-max normalization across all recipe.

References