2AMD20 – Knowledge Engineering

Ketogenic Diet Recipe Finder

M. van Alten, N. van den Brand, D. Kim, M. Laamoumi

Group 13

Research Question

Client: Founder of a cooking website trying to expand their business.

Goal: Create a tool that finds recipes that fit the ketogenic diet (consisting of < 10% carbs, ~20% protein, and ~70% fat), ideally allowing the user to select a specific taste.

Data Sources

RecipeNLG dataset: contains 2 million cooking recipes. For every recipe, a list of instructions and two lists of ingredients are provided. One list includes the quantities required. We only used the first 200,000 recipes to increase processing speed.

Taste, fat, and texture database: consists of 627 Dutch food products, for which a taste profile is provided. This taste profile is measured in salt, sweet, sour, bitter, fat, and umami.

Open Food Facts database: is a collection of 2.8 million products sold in supermarkets globally.

Its 203 attributes describe the product, location of origin and sale, brand, quantities, nutriscore, allergens, additives, ingredients, and about 30 nutritional values per 100 grams of the product. The values for fat, protein, and carbohydrates are among the nutrients listed.

Approaches

Data Integration Cleaning Analysis: We merged the taste and food facts database to match ingredient taste and nutrition. We discarded all irrelevant attributes, leaving 11 relevant ones. We removed unuseful entries manually, added our own 'Key Ingredient' column with the main ingredient of each item, and used that to merge. This column was used for the actual merge, which left 90 ingredients with taste and nutrition.

This column was then also used to match ingredients between this dataset and the recipe dataset. After cleaning the recipe data and discarding recipes for which we did not match at least 1 ingredient correctly, there were 13,000 recipes left, with 60 ingredients in them.

RDF: Python library 'RDFlib' is used to make an RDF graph. First, there are several classes and properties defined. Namely, the 'Recipe' and 'Ingredient' classes are defined as blank nodes. They are connected by asserting 'hasIngredient' property with 'Recipe' as its domain and `Ingredient' as its range. Also, a property such as 'hasSweetness' is defined in 'Namespace'. After defining a general scheme, actual matching happens. Dictionaries 'recipe_dict' and 'ingredient_dict' are created to keep track of recipe and ingredient nodes. Then each dataset is iterated by using the key (e.g. 'recipeID', or 'recipeID + ingredient_name'), and if the key does not exist in each dictionary, then a new recipe or ingredient node is created. Lastly, corresponding properties are added to each blank node as well.

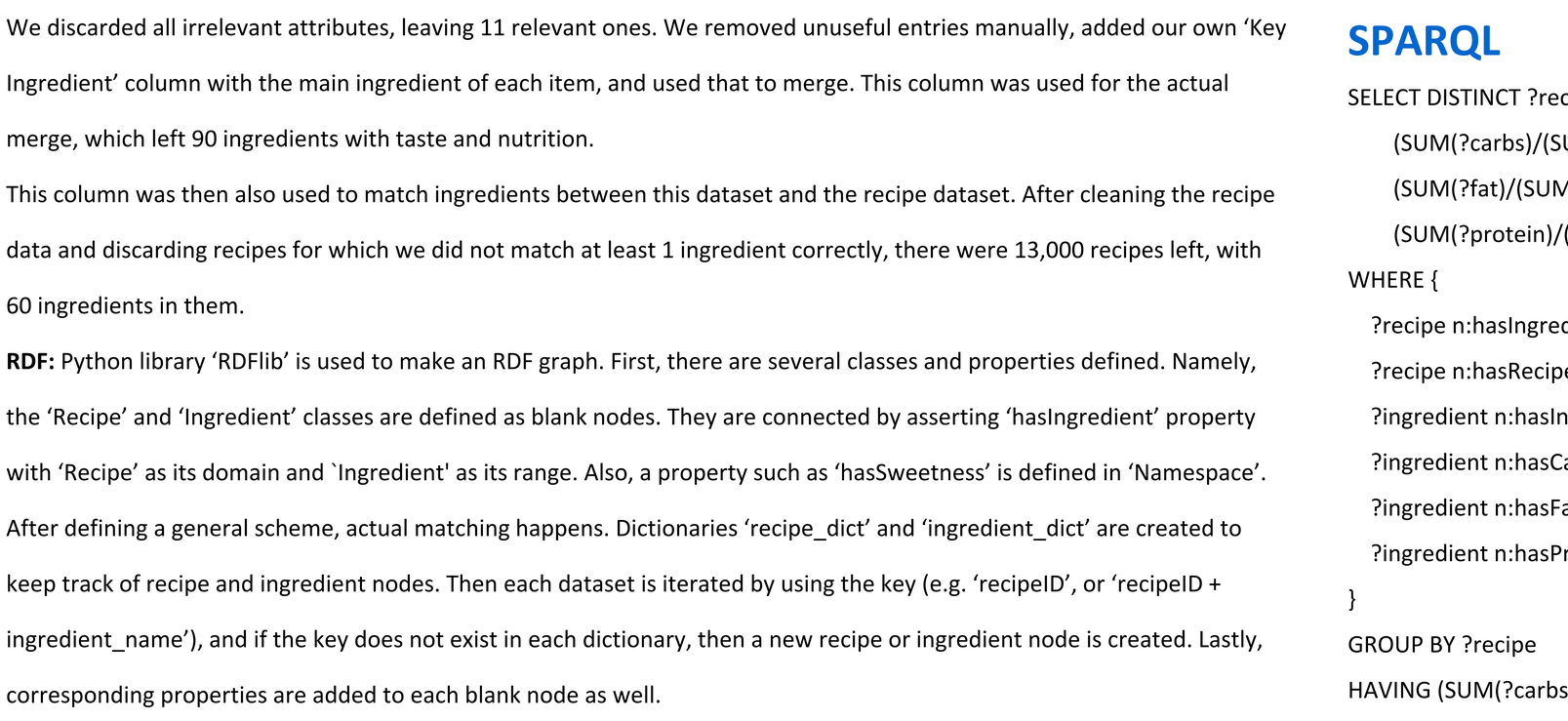

Figure 2: Visualisation of the knowledge graph of the recipe for plain flour dumplings.

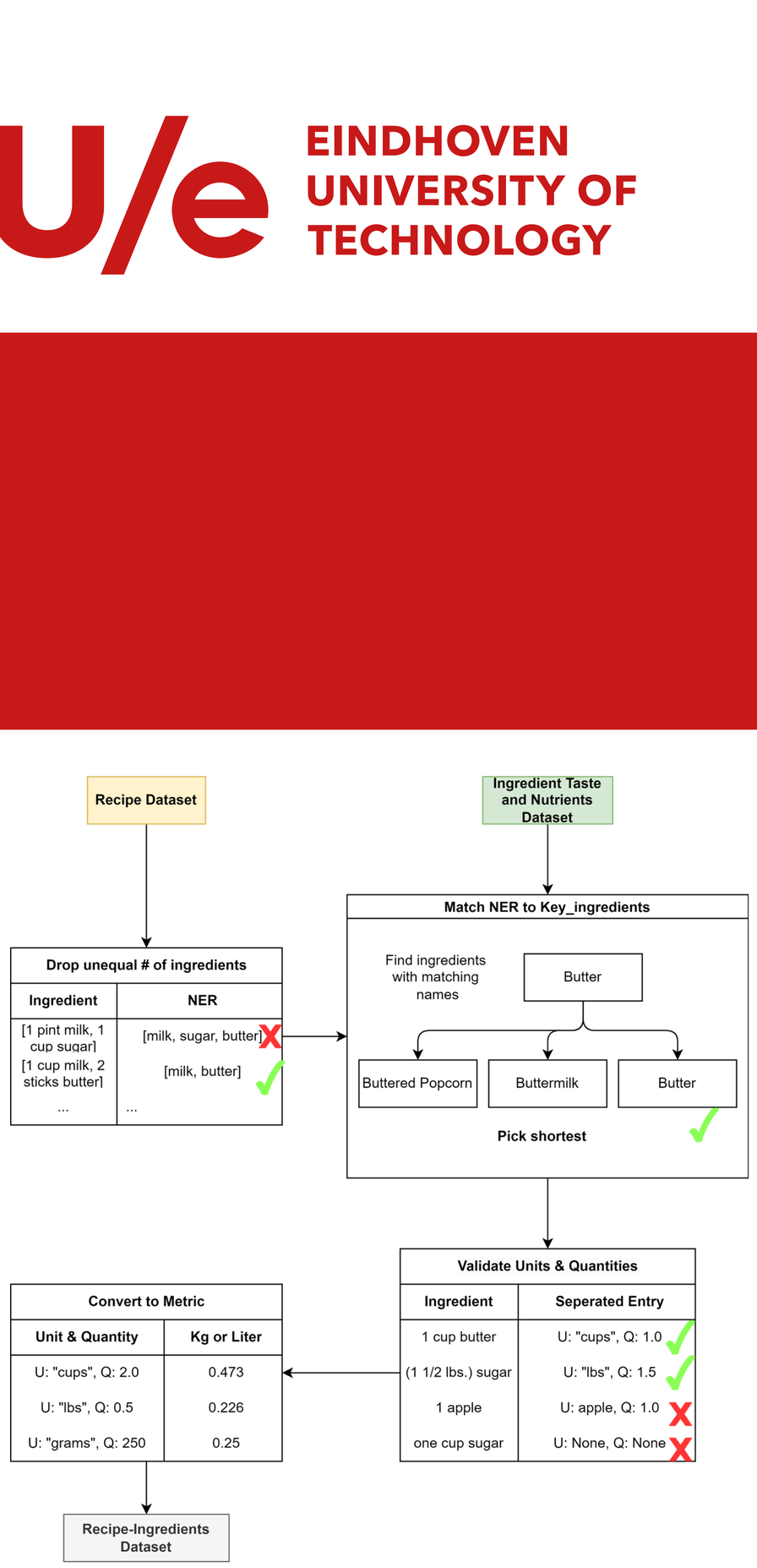

Figure 1: Data cleaning and pre-processing pipeline.

SPARQL

```
SELECT DISTINCT ?recipeName
      (SUM(?carbs)/(SUM(?carbs)+SUM(?fat)+SUM(?protein)) AS ?carb_percentage)
      (SUM(?fat)/(SUM(?carbs)+SUM(?fat)+SUM(?protein)) AS ?fat_percentage)
      (SUM(?protein)/(SUM(?carbs)+SUM(?fat)+SUM(?protein)) AS ?protein_percentage)
WHERE {
   ?recipe n:hasIngredient ?ingredient.
   ?recipe n:hasRecipeName ?recipeName.
   ?ingredient n:hasIngredientName ?ingredientName.
   ?ingredient n:hasCarb ?carbs.
   ?ingredient n:hasFat ?fat.
   ?ingredient n:hasProtein ?protein.
}
GROUP BY ?recipe
HAVING (SUM(?carbs)/(SUM(?carbs)+SUM(?fat)+SUM(?protein)) < 0.1)
ORDER BY DESC(?fat_percentage)
```

Results

We found 13 recipes with 1 ingredient and 1 recipe each with 2 or 3 ingredients that fit the ketogenic diet. However inspecting the recipes shows the results are not very reliable. All recipes are mostly based on chicken in the matched ingredients, but most are clearly not real chicken dishes. For example 'Oven Paprika Potatoes'.

We think this is because of issues matching the ingredient taste and nutrition profile. For example we did not match 'salted butter' to any entry in the Food Facts database. More time may allow us to improve the matching algorithm here or improve the Key Ingredient attribute so it works better.