Healthier tasty recipes recommender system

Martin Chatton
martin.chatton@epfl.ch

Lucas Massemin
lucas.massemin@epfl.ch

Lionel Pellier
lionel.pellier@epfl.ch

Abstract

In this paper, we gather more than 1.5M cooking recipes extracted from several existing datasets and exploit the USDA database to compute the nutritional values of recipes and ingredients. Using these data, we build an embedding for the ingredients to gain insight into how they are combined in recipes. Moreover, we use a standard food label to rate the healthiness of recipes and group the healthiness scores into three levels, green, amber and red. Finally, using the previous results, we design a recommender system able to change a recipe to make it healthier, while aiming to preserve its tastiness. The recommender takes as input any recipe whose ingredients appear in the datasets' recipes, along with their quantities. The output is a new recipe, healthier equivalent of the potentially unhealthy input recipe.

1 Introduction

As a daily experience, food plays an important role in human life, more than just a primary need. Indeed, we do not only eat to survive, but meals also have a social dimension and are a source of satisfaction. With this in mind, humans have adopted a relatively complex model for meals composition, to achieve maximal tastiness. The latter's constituents, courses, are created by combining ingredients in particular ways stated in the so-called "cooking recipes". Parallel to the growth of the number of recipes, scientists studied the actual human beings needs in terms of food, and came up with recommended nutrients intakes for people to stay in good health. Unfortunately, people tend to follow their taste, and those recommendations are often ignored, leading to a diabetes epidemic and other potential food-related diseases. In this paper, we first get insight into what makes a recipe tasty by looking at ingredients associations and then design a recommender that suggests a healthy recipe equivalent to an unhealthy and tasty recipe given as input.

2 Datasets

The recommender uses known ingredients associations, quantities and ingredients compositions to find a recommendation. Those information are retrieved from four different datasets, namely "what's cooking"[1] (WIC), "From cookies to cooks"(FCTC, 2013), "1M Cooking recipes"(1M, 2017) and the "USDA food composition database"[2] (USDA).
WIC and FCTC contain 96763 recipes, for which they simply provide lists of ingredients.
1M is more complete, as it contains more than 1 million recipes. For each recipe and for each ingredient, a string encodes the ingredient name along with its quantity, a boolean indicates whether the ingredient can be found in USDA and a string gives the extracted ingredient name.
Finally, USDA contains no recipes, but provides a mapping between ingredient ID, description and nutrients composition.

3 Preprocessing

We preprocessed the recipes to reduce the number of ingredients by merging similar ingredients together ("canola oil" and "corn oil" for instance). Also, for each merged ingredient, we retrieve his nutritional profile. The overall ingredient reduction is presented in Figure 1.

[1] https://www.kaggle.com/c/whats-cooking
[2] https://ndb.nal.usda.gov/ndb/search/list?home=true

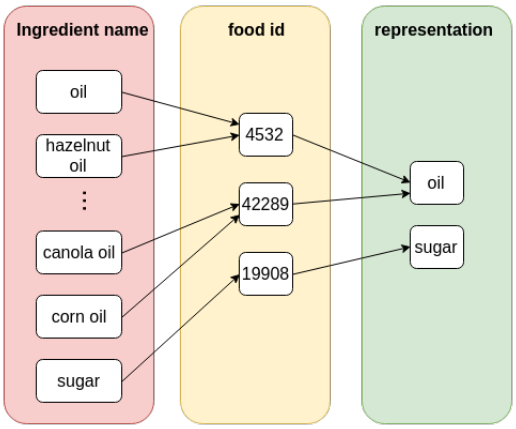

Figure 1: ingredients reduction process

3.1 Ingredients Cleaning

For consistency purposes, ingredients entries have been cleaned the same way in WIC, FCTC and 1M. Among other processes, ingredients have been singularized and special words have been removed (mainly size and useless adjectives).

3.2 USDA Mapping

In order to gather nutritional information for each ingredient, we simply selected the USDA food id whose description contained all the words included in the recipe's ingredient name.

3.3 Ingredients Merging

The previous mapping already merged some ingredients together, more specifically the ones that were mapped to the same food ids.
For each food id, we create a string representation based on the names of the ingredients that were mapped on this id. The representation is composed of the words that occur in more than 50% of the names of the ingredients mapped on the id. Consequently, some ids end up with the same representation, this concludes the merging process.

3.4 Quantity Extraction

Quantities are essential to compute the recipes healthiness. Most ingredients entries follow the pattern "quantity unit ingredient", for instance "3 teaspoons honey". Based on that observation, we build a parser using a natural language toolkit library[3] which is used to read the input recipes. The parser is able to find all the ingredients quantities for 206887 recipes in the datasets. The parser is limited by the fact that some ingredients are given without a specific unit ("4 eggs" or "6 apples" for instance). Those cases were identified, and a weight in grams was manually computed for the most common ingredients.

[3]http://www.nltk.org/

4 Recommender Design

In this part we describe the recommender architecture and the recommendation policy we adopted. We first introduce the way we exploit the ingredients associations and then present our healthiness metrics before introducing the swapping algorithm.

4.1 Food Embeddings

We got inspriration from the famous John Firth's quote, namely *"you shall know a word by the company it keeps"*. This intuition seems valid when applied to ingredients, thus we decided to map the ingredients associations problem on the words associations one. We used a famous word embeddings library based on Word2Vec (2013) to build our ingredients embeddings, considering each ingredient representation as a word, and each recipe as a sentence.

4.2 Healthiness Metrics

The healthiness of a recipe is computed using 5 components, salt, sugar, fats, saturated fats and energy. We first present the quantity ranges defined as healthy per nutrient per 100 grams and then introduce two scores, one per nutrient and one per recipe.

Healthy quantity ranges

We used the FSA traffic lights system[4] to determine whether the nutrients quantities were high, medium or low. For the energy, a similar intuition was derived from the Recommended Daily Intakes[5]

Nutrient	Unit	Recommended Amount
Salt	grams	0.3 to 1.5
Sugar	grams	5.0 to 22.5
Fats	grams	3.0 to 17.5
Saturated fats	grams	1.5 to 5
Energy	Kcal	200 to 400

Table 1: Nutrients recommendation per 100g.

[4]https://www.resourcesorg.co.uk/assets/pdfs/foodtrafficlight1107.pdf

[5]https://ods.od.nih.gov/Health_Information/Dietary_Reference_Intakes.aspx

Per-nutrient Score

For each nutrient, we compute a score (penalty) based on its quantity per 100g. Because we do not want to prioritize a specific nutrient, we define the nutrients scores to be the same at their recommended range extremities. We choose 3 to be the score at the lower extremity, and 25 to be the score at the upper one. The score function is thus a composition of linear functions, note that the slope stays the same for quantities greater than the leftmost interval value. Also note that the values 3 and 25 have been chosen to be 'far' from each other to penalize unhealthy nutrients quantities more aggressively. The per nutrient score function is represented in figure 2.

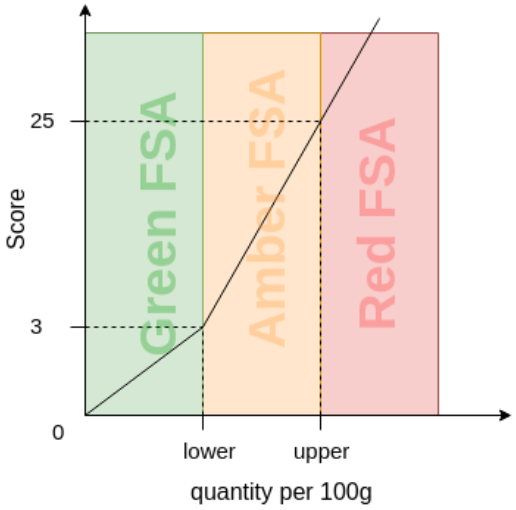

Figure 2: per-nutrient score function

Multi-nutrient Score

The global score for a recipe is simply the quantity-weighted sum of the nutrients scores of its ingredients. We designed the per-nutrient score such that a red traffic lights color cannot be compensated with only one green color and should be avoided as much as possible.

4.3 Swapping Strategy

The recommender should be able to give an accurate swapping recommendation when a recipe is given as input. The swapping strategy is the following.

First, for each ingredient, we identify the n = 10 most similar ingredient representations (in terms of proximity in the embeddings space) and keep the ones whose similarities are above a certain threshold t = 0.45. From those candidates, we discard the ones whose name contains the name of the ingredient we want to swap to get more interesting swaps. Several ids are mapped to each of those representations, we retain the one with the best multi-nutrient score. Because we want to keep only one food id for the ingredient to be swapped, we again keep the one with the best multi-nutrient score. In that way, for each ingredient, we get the best swapping possible. In the final phase, we simply swap the ingredient whose swapping candidate has the best multi-nutrient score.

5 Results

5.1 Embeddings

We will show the embedding results through two examples.

5.1.1 Butter

Ingredient	Similarity
margarine	0.656
heavy cream	0.556
cream	0.533
whipping cream	0.513
shortening	0.484

Table 2: Ingredients most similar to butter

In this example, we see that, even if the similarity scores are not very high, the suggestions are pertinent and could be used in practice as substitutes for butter. Also, we note that we face a precision trade-off with the representatives 'cream' and 'heavy cream'. Indeed, we do not want to lose information for 'heavy cream' and we cannot afford to blindly map 'cream' to a more precise representative. As a result, those two representative can be swapped even thought it makes little sense.

5.1.2 Lettuce

Ingredient	Similarity
iceberg lettuce	0.896
romaine lettuce	0.805
salad green	0.771
green lettuce	0.751
salad dressing	0.634

Table 3: Ingredients most similar to lettuce

In this example, we see that the similarity is much higher. The first recommendations are in-

deed substitute of lettuce, but are not a very interesting swap to make in a recipe. Therefore, for our recommender algorithm, we chose not to consider substitute that have the same representative, to obtain more interesting swaps. After that, we see that the algorithm suggest swapping salad for salad dressing. Here, the reason is that the algorithm is biased by the omnipresence of dressing in a lot of salad recipes, and the algorithm sometimes considers ingredients that are often associated with the ingredient to swap

5.2 Recommender

5.2.1 Baking

Baking recipes contain often a lot a fat (through ingredients like butter) and a lot of sugar. We test our recommender on a bakery recipe from our dataset too see if we can make it lighter.

Quantity	Ingredient
256g	Turbinado Sugar
2.1g	Cinnamon
2.1g	Clove
2.1g	Nutmeg
96g	~~Butter~~ Margarine
256g	Buttermilk
4.2g	Baking Soda
128g	Raisin
128g	Pistachio nut

Table 4: Recommender on a baking recipe

Our recommender suggests to replace the butter by margarine, resulting in an improvement of 23.04% regarding our metrics

5.2.2 Top 1 Marmiton

We also test our recommender on the meal that is ranked first on the Marmiton website. [6]

This results in an original recipe that we can perfectly imagine to be tasty. Indeed, it appears that other white meats are often used as replacement for the veal in a blanquette.

6 Further Work

First of all, notice that Word2Vec maps words next to each other in the reduced dimensional space according to their co-occurrences. Unfortunately, not only synonyms are likely to occur within the same context. Indeed, words often associated together ('cat' and 'purr' for instance) will share the same context as well and will have very similar vectors. In the ingredients associations case, this is equivalent to mapping "coconut" and "rum" next to each other, as they often appear together in cocktails. The recommender is only interested in finding swappable ingredients, which is equivalent to focusing on food 'synonyms'. This could be achieved with a lexical substitution model, such as (Word2Vecf, 2015) for instance.

[6] https://www.marmiton.org/recettes/recette_blanquette-de-veau-facile_19219.aspx

Quantity	Unit	Ingredient
1	kg	~~Veal~~ Rabbit
1	tsp	Chicken Broth
2	_	Carrots
1	_	Onion
1	cup	Mushroom
1	cup	Cream
1	_	Lemon
1	_	Egg yolk
1	cup	Flour
1/4	kg	White Wine
2.5	g	Kosher Salt
2.5	g	Pepper

Table 5: Recommender on a blanquette of veal

7 Conclusion

In this paper we shew that the word embeddings process was an effective way to tackle the ingredients associations problem. After cleaning several recipes datasets, we built an ingredients embedding from a large quantity of recipes. We illustrated the latter relevance by creating a recommender based on it, able to improve a recipe healthiness while preserving its tastiness.

References

Salvador, Amaia and Hynes, Nicholas and Aytar, Yusuf and Marn, Javier and Ofli, Ferda and Weber, Ingmar and Torralba, Antonio. 2017. Learning Cross-Modal Embeddings for Cooking Recipes and Food Images, pages 3068–3076. 10.1109/CVPR.2017.327

West, Robert and White, Ryen W. and Horvitz, Eric 2013. From Cookies to Cooks: Insights on Dietary Patterns via Analysis of Web Usage Logs, pages 1399–1410. 10.1109/CVPR.2017.327 New York, NY, USA

Mikolov, Tomas and Chen, Kai and Corrado, G.s and Dean, Jeffrey 2013. Efficient Estimation of Word Representations in Vector Space

Melamud, Oren and Levy, Omer and Dagan, Ido 2015. A Simple Word Embedding Model for Lexical Substitution, pages 1–7 10.3115/v1/W15-1501