An exploration of recipes

Furtuna Andrei-Calin, Gafton Paul-Adrian, Mircea Sorin-Sebastian, Mocanu Alexandru

***Abstract*— As the standard of living greatly increased in the past century, people have more access to food than ever. Exotic ingredients, which used to be prohibitive and scarce, become more and more accessible to the general public. This, along with the increase in communication, created a globalization of the recipes. We analyze a collection of recipes, trying to gain more insights into how various dishes are related and also into the recipes themselves.**

I. INTRODUCTION

In this project we are aiming at analyzing relationships between recipes and their ingredients based on the Recipes 1M data set [1].This data set contains approximately 1 million recipes retrieved from different websites, and was first used in order to build a model which would give cooking instructions from an image representing the dish. A smaller subset exists, with additional nutritional information concerning fat, sugars, saturates, etc. Thus, we will try to build different graphs and carry out different analyses based on this dataset.

II. DATA ACQUISITION AND PROCESSING

We use a subset of the Recipes1M dataset for our analysis. The subset contains 51235 recipes. The attributes of the recipes that we use are the recipe name, its ingredients, the quantities used from each ingredient, the nutritional values per 100g and cooking instructions. The ingredients of a recipe are represented as a list of keywords. The tuples of keywords may be too explicit too use in order to define a list of possible ingredients. Therefore, we generate our list of ingredient names such that those names don't occur neither too often nor too rarely.

Using the dataset, we build two types of graphs: an ingredient graph and a recipe graph.

The ingredient graph $G_I = (V_I, E_I)$ consists of vertices V_I associated to the ingredients and edges E_I such that the weight w_{ij} between ingredients i and j is equal to the number of recipes in which i and j occur simultaneously. The number of nodes is 251 and the number of edges is 15996, so an average degree of about 64, which is very high.

The recipe graph $G_R = (V_R, E_R)$ consists of vertices V_R associated to the recipes and edge E_R such that the weight w_{ij} between recipes i and j is equal to a sum of tf-idf-like terms for the ingredients that i and j have in common. Explicitly, for recipes i, with n_i ingredients, and j, with n_j ingredients, if they share ingredient k, the term associated to it is $t_{ij,k} = \frac{2}{n_i+n_j} \log(\frac{N}{N_k})$, where N is the total number of recipes and N_k is the number of recipes containing ingredient k. We mention that some of the ingredients were extremely common (such as water or salt), occurring in more than 40% of the recipes and therefore they were excluded from the list of ingredients while constructing the graph, as they would have increased the number of edges to the point that they would not fit into memory. Those ingredients were however reintroduced afterwards as features of the nodes. In addition to that, the graph representation that we obtained after a first stage of graph construction was a set of adjacency lists. A more suitable format, namely a sparse weight matrix, would not fit into memory, so we proceeded with a step of downsampling, storing only the heaviest 10% of the edges. The number of nodes is 51235 and the number of edges is 12785429, so an average degree of about 250.

Along with the graphs, we store two dataframes, one containing the names of the ingredients and another one containing the names of the recipes and for each recipe tuples of ingredients contained in the recipe along with the percentage of the

quantity used.

Additionally, since we did not have any information about the cooking time, we decided to retrieve this information from instructions provided with each recipe. The instructions consist of raw text stored as sentences. In order to compute the cooking time, we looked for words that quantify time (such as seconds, minutes, hours and equivalent words) and identified the corresponding values situated in front of those words. After some normalization and dealing with exceptions, we had, for each recipe, different time quantities (seconds, minutes, hours) and their corresponding values. Multiplying the two and summing all for each recipes gave us an approximate cooking time. We also decided to add, in minutes, the length of each recipe (in sentences), in order to avoid having 0-minute recipes.

III. GRAPH EXPLORATION

Using the available data, we built multiple graphs to gain more insight into its structure. This means that starting from the original graphs described above, we also performed downsampling. This was done either to concentrate on some parts of the graphs or, in case of the recipe graph, to make some types of analysis manageable.

A. Ingredients Graph

The graph consists of a single connected component. By analyzing the degree distribution, presented in figure 1 for both the weighted and unweighted versions, we can see that the weighted graph has a rather scale-free figure. However, when judging the nature of a graph, we are interested in the unweighted version, in which case we see fairly uniform distribution of degrees. This means that our graph is quite dense and it does not fall into the scale-free category.

We also generated an Erdos-Renyi graph with a similar number of edges (due to the denseness of the network we could not generate a Barabasi-Albert graph) and concluded that neither the distribution nor the average clustering coefficient recommend the ingredients graph as falling in this category of random graph.

If we downsample the graph to only keep the ”heaviest” 2% of the edges, we see that only a quarter of the nodes are included in the giant component of the subsampled graph.

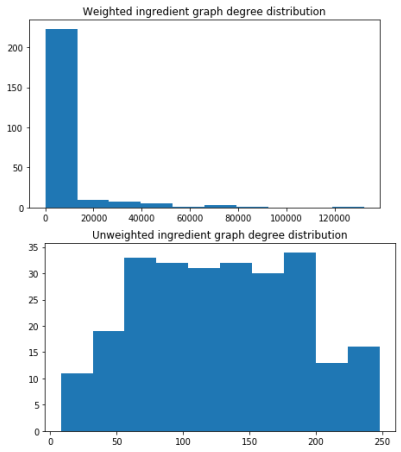

Fig. 1: Degree distribution for ingredients graph

We also look at several centrality measures (degree centrality, closeness centrality and betweenness centrality) and conclude that water, salt, sugar and butter always are the top ingredients. This is expectable due to the widespread use of these ingredients.

B. Recipes Graph

The important number of nodes do not let us provide a proper visualization of the graph. Hence, we decided to plot only a small subgraph in order to have an idea of how it is represented (Figure 2). For this visualization purpose, we used Gephi 0.9.2[2] and colored the different nodes according to their degree.

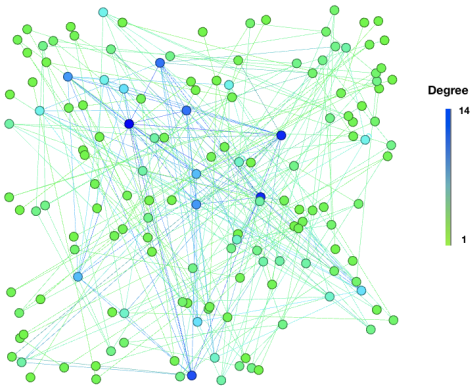

Fig. 2: Representation of a subgraph of the recipes graph.

We carry out a similar analysis on the recipe graph. Again, the degree distribution does not show any hint of scale-free behaviour, as it can be seen in figure 3.

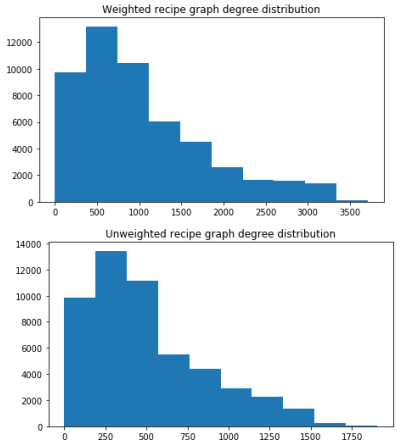

Fig. 3: Degree distribution for recipes graph

Interestingly enough, the distributions for the weighted and unweighted graphs now look similar, with the difference that the weighted one has degree values roughly twice as big as the unweighted one. This is due to the fact that most of the edges have values around 2.

We also note that due to the sparsification performed when constructing the graph, we obtained a large number of connected components (roughly 3.5k). However, most of them are isolated nodes and the largest one has 47642 nodes. To carry out any further analysis, we need to downsample the number of nodes even further. So we take the giant component, we take the 10% most "heavy" edges and take the giant component of the resulting graph. We get a graph with 28378 of nodes and 604219 edges, so an average degree of about 21. Figure 4 illustrates the degree distribution of the resulting graph.

It looks roughly as a power-law, thus indicating a somewhat scale-free structure.

We compared this graph to Erdos-Renyi and Barabasi-Albert graphs with similar numbers of edges, but our graph does not fall in any of these categories. The average clustering coefficients, especially, were much lower for the random network models than for the recipe graph.

Last but not the least, the only reasonably computable centrality was the degree centrality.

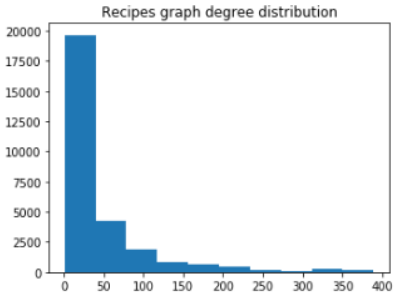

Fig. 4: Degree distribution for downsampled recipes graph

All nodes have a low degree centrality which is explainable through the small number of connections that the nodes have.

Finally, let us have a look at two attributes we are mostly interested in : the cooking time and the fat per 100g. We decided to restrain the visualisation at recipes having a cooking time less than 2 hours.

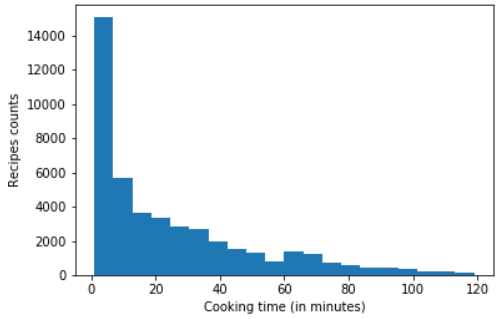

Fig. 5: Cooking time distribution

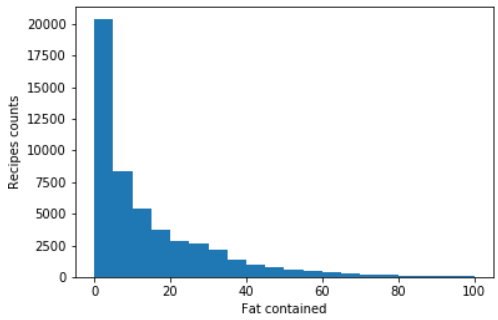

Fig. 6: Fat per 100g distribution

The two distributions could be interpreted as power laws : a lot of small values (less than 20 minutes to cook, less than 20 'fat' contained) and some large values.

IV. RESULTS

A. *Clustering of recipes*

We are interested here in how we can regroup some recipes based on their cooking time and the fat they contain. In order to do this, we used the KMeans algorithm with 5 clusters on a matrix composed of the cooking time and the fat contained for each recipe. A representation of the resulting clustering can be found in Figure 7

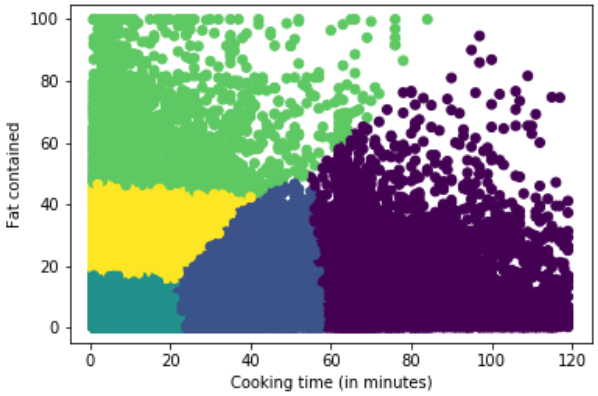

Fig. 7: Clustering of the recipes based on the fat and cooking time using KMeans.

We can observe several important groups : a first group composed of fast recipes and low in fats, fast recipes but with a high amount of fat and finally medium-long recipes with low and medium fats. The first group is particularly interesting to analyze. Indeed, if we consider that we eat about 2,000 calories a day, then amount of fat should not be higher than 80 grams (the reference is between 20 % and 35% of total calories per day). Hence, for a meal, considering we look at the values per 100g of the dish, the quantity of fat should be less than 20. Furthermore, today people look for easy and quick meals to prepare, especially specific groups such as students or people not having enough time to cook. All of this corresponds to the first cluster : less than 20 in fat and around 20 minutes of cooking time.

If we take a look at this first group, the main recipes are : smoothies, dips/sauces/seasoning, salads, soups and deserts but also drinks. A lot of those recipes include fruits or vegetables, hence explaining the small quantity of fat. These should be the recommended quick dishes for a healthy way of eating. On the other hand, if we look at the fast dishes but high in fat, we find mostly sauces/dips, desserts too, snacks and dishes with ingredients such as cheese, butter, nuts (high in fat). Finding sauces and dips in this category is not surprising : it is quick to prepare a sauce, but the value of fat is per 100g - one rarely eats 100 grams of sauce ! This is the category of dishes that should mostly be avoided.

V. LIMITATIONS AND CONCLUSION

Throughout the project, we tried to gain insights into the different relationships between ingredients and recipes : ingredients are connected based on the recipes in which they are used (dense graph), while recipes have much smaller degrees. We also highlighted important features of the recipes, such as the cooking time and nutritional values.

Nevertheless, we were confronted with some difficulties such as the size of the recipes graph (use of sparse matrices, long computation time, downsampling of the edges) and some tasks such as visualisation have been performed only on subgraphs (some tasks couldn't be performed such as different centrality computations). Furthemore, our dataset does not only include meals/dishes but also drinks, sauces/dips, thus making our analysis less relevant: a filtering of the recipes could have been performed before (exclude recipes containing the words sauce, dip, juice, drink or alcohol names for instance). Finally, the computation of the cooking time is not precise, even if it constitutes a good first approximation, given the inconsistency of the raw text (different source websites, different authors).

REFERENCES

[1] Javier Marin et al. "Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images". In: *IEEE Trans. Pattern Anal. Mach. Intell.* (2019).

[2] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. "Gephi: An Open Source Software for Exploring and Manipulating Networks". In: (2009). URL: http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154.