Portuguese Gastronomy

The Art that feeds Life.

Analyzing Big Data – 1st Semester 19/20

Enterprise Data Science & Analytics Post-Graduation

GROUP COMPOSITION

Francisco Costa, 20181393

João Gouveia, 20181399

Nuno Rocha, 20181407

Pedro Rivera, 20181411

December 6th, 2019

Table of Contents

Keywords: Cuisine, Portugal, Portuguese Gastronomy, Spark, Big Data, NLP, Graph

Introduction & Contextualization

Portuguese gastronomy is characterized by a unique combination of three elements: geography, climate and history. A broad geographical spectrum combined with a mild climate allowed the development of a wide range of food, resulting on a diet using various vegetables, fruits and a healthy balance between meat and fish. The Age of Discoveries was a historical landmark for cultural exchange. Establishing a global trade network made possible to use new ingredients and acquire foreign culinary knowledge. Even nowadays, it is possible to witness this development and its influence on other cultures.

This rich cuisine enabled the use of recipe variants for a single ingredient, taking the example of the cod fish. However, a strong heritage does not necessarily guarantee a bright future. In this 21st century, as we witness the transformation of the modern society into a digital world, tools are needed to leverage the use of data analytics and gather new insights to build an enriched experience to ones who enjoy the pleasure of this cuisine.

Under that scenario the group aimed to analyze Portuguese gastronomy using Big Data technologies.

Methodology

The group started this study by analyzing several traditional recipes and the required ingredients for each regional dish. The relationship between dishes and ingredients is core to our analysis, as this will provide the groundwork to build other relationships that will enable our model to develop a deeper understanding of this gastronomy.

In order to properly identify the culinary information from the recipes, it was also needed to develop a natural language processing algorithm that will be further explained in this report. This algorithm is built on a distributed paradigm, allowing huge performance improvement and scalable design. To leverage the highly related data from each component, the group proposed a solution built on a graph model, which provides greater flexibility to scale for additional components or data from foreign gastronomies.

Finally, the group will also describe some use cases in which this solution can be able to deliver insightful information both for consumer and business perspectives.

Data Sources

The regional recipes data was extracted by performing a web scraping to Roteiro Gastronómico de Portugal (*www.gastronomias.com*), a website with information about traditional recipes as well as their origin based on national regions.

Additionally, and since Wines play an important role in the local gastronomy, some information was collected through scrapping from VINHA (*www.vinha.pt*), a wine shop which incorporates wine food pairings and ratings from Vivino, which is a social wine network that collects opinions from experts and general public.

Why Natural Language Processing?

The recipe data source is a collection of regional recipes from different authors. This translates into poorly defined structure and textual variants from recipe to recipe. To be able to properly analyze such information, it was necessary to apply natural language processing (NLP) to structure such data into the following components: **Quantity**, **Unit**, **Ingredient**.

For an easier understanding, consider the output for the following input data examples:

Input data
500 g de açúcar
2 colheres de sopa de azeite
2 cebolas
salsa
1,5 dl de azeite
raspa da casca de 1/2 limão

Quantity	Unit	Ingredient
500.0	g	Açúcar
2.0	colheres de sopa	Azeite
2.0		Cebolas
1.0		Salsa
1.5	dl	Azeite
0.5		Limão

As for the NLP algorithm, it can be divided into the following steps:

Quantity Extraction

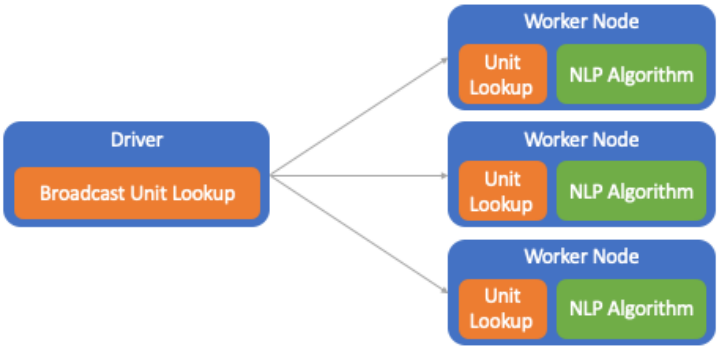
As one may expect, numerical representations can be expressed in multiple formats. The algorithm is capable of extracting numeric quantity data from those multiple formats. Consider the following examples:

Format	Example
Integers	2 500
Decimal	3.5 1,5
Fractions	1/2 1/4

Output
2.0 500.0
3.5 1.5
0.5 0.25

Unit Extraction

The unit extraction algorithm compares every word (or combination of multiple words) that could be matched to a known unit reference. In the cuisine domain this can be SI reference units, culinary units or reference to a part of an ingredient. The algorithm uses a unit lookup list that is broadcasted to every worker node.

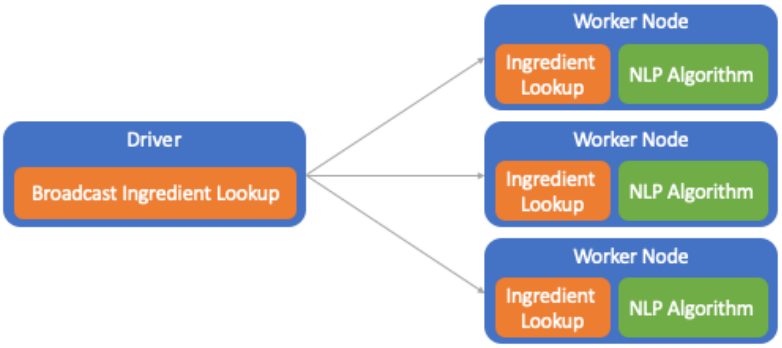


Since the unit lookup is an extensive list, consider the following examples of different types of units:

Type	Unit Lookup
SI reference units	gramas, g, kg, dl
Culinary Units	colheres de sopa (tablespoons)
Part of an ingredient	folhas (leaves)

Ingredient Extraction

The ingredient extraction is the most important step, in which the algorithm behaves similarly to the unit extraction. However, it is capable of detecting plural forms and convert them directly to singular in order to allow suitable comparison between inputs that refers to the same ingredient. It also uses an ingredient lookup list that is broadcasted to every worker node.



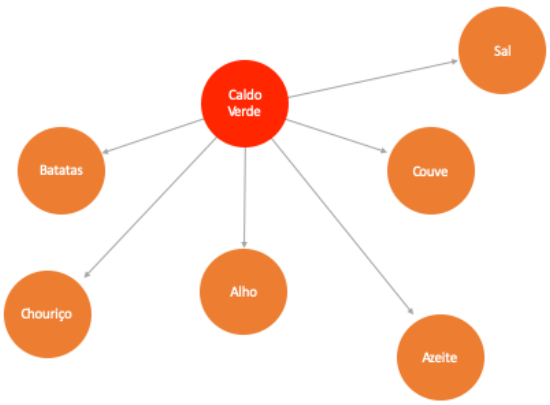
Since the ingredient lookup is an extensive list, consider the following examples of different types of ingredients:

	Ingredient Lookup		Output
Singular and plural forms	limão, limões pães, pão	→	Limão Pão

Why Graph technology?

The group faced a key decision in how to analyze the relational components of Portuguese gastronomy. Although tabular structures are extremely common, their use is not the most adequate when it comes to analyze highly relational data. Since each gastronomic component is highly relatable to each other, the choice relied on a graph model.

Building a solution around graph technology enabled to take advantage of highly connected data to drive various insights. The group treated each data component as vertices and defined its relationships as edges. This simplifies the data model and allows great flexibility regarding scalability, since any future addition of new information can be added by creating new vertices and defining additional relationships with new edges. Consider the following example of the graph relationship between a dish and its ingredients:



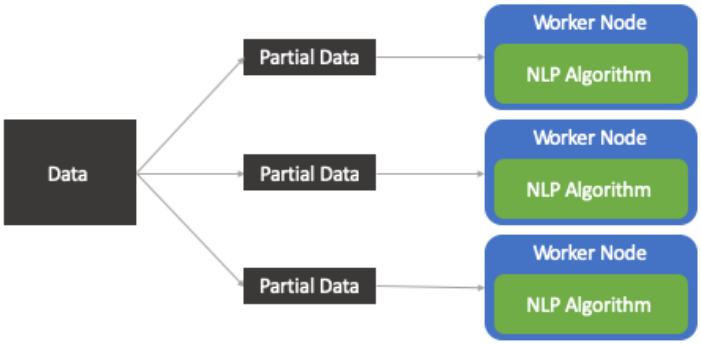
Why Spark?

The Apache Spark framework is the right choice to implement such analytic solution by offering a distributed environment with high performance due to a distributed memory-based architecture.

It enables a unified platform for: (1) distributed storage and processing of tabular structures (DataFrames), (2) the use of parallel processing of our natural language processing algorithm, and (3) the integration of graph structures and algorithms.

Parallelize the NLP algorithm

The NLP algorithm that extracts quantity, unit and ingredient information can take advantage of the distributed design. This allow multiple worker nodes to perform the processing tasks on each set of partial data and reduce significantly the compute time to perform this algorithm.



Designing the Graph model

The Spark framework comes with a graph component named GraphX, built around RDDs. However, new developments have been made that led to the creation of GraphFrames library.

Unlike GraphX, GraphFrames is based on DataFrames, rather than RDDs. Another advantage is the fact of being much faster than the former, due to the Catalyst and Tungsten performance layers built into Spark SQL. Catalyst, the query optimizer, and Tungsten, the direct memory manager that bypasses the JVM, can be considered turbocharger add-ons.[1]

Although this library it is still under development, its potential is already recognized by the Apache Software Foundation, to the point of considering its inclusion for future Apache Spark revisions.

---

[1] Malak, Michael S., and Robin East. 2016. Spark GraphX in Action. Manning Publications.

Final Output

Once all the previous steps were executed, it was time to build the graph model. An overview of all the relational components in our model can be seen in the following figure:

- Regional origin of dishes;
- Recipe information: Dishes and used Ingredients;
- Wine pairing to Food Categories;
- Restaurants, establishments that prepare dishes and serve them to consumers;
- Producers (individuals or organizations) that produce an ingredient and supply it to restaurants.

As explained in the Data Sources segment, the Wines dataset was incorporated in the model, by being connected to the recipes through the Food Category. Regarding the restaurants and producers' components, some examples were integrated into the graph, just to exemplify some of the further described business cases.

Querying the Graph model

It was finally time to start digging for some insights. Nevertheless, to do that, it seemed important to understand the information's flow in the model. By other words, it is essential to realize the vertices can be either a source or a destination, depending on the edges that are under analysis. For instance, note that **Dish** is a source for the edge that connects it to **Ingredient**, but is a destination in the edge that connects it to **Food Category**.

So, for example, one aims to find *which dishes pair with Pêra-Manca wine from 2014?*

```
find_dishes = g.find("(a)-[]->(b); (b)-[]->(c)") \
               .filter("a.id = 'Pêra-Manca (2014)' and c.Type = 'Dish'") \
               .select("c.id")
```

From the previous query, the ***find*** method provides the path and direction, while the ***filter*** method establishes relevant characteristics (in this example: starting point and destination). Finally, the ***select*** method allows to specify what is the information returned by the query.

Use Cases

As mentioned before, in order to show the potential of such project, the group decided to develop some uses cases that are relevant for Business-To-Business and Business-To-Consumer perspectives.

1. Find red wine pairings for Lamprey Rice (Arroz de Lampreia) dish:

```
find_red_wines = g.find("(a)-[]->(b); (b)-[]->(c); (c)-[]->(d)") \
                  .filter("a.id = 'Vinho Tinto' and d.id = 'Arroz de Lampreia'") \
                  .select("b.id")

wines_by_rating = find_red_wines.join(df_wines, find_red_wines.id == df_wines.Name) \
                                .orderBy('Rating', ascending = False) \
                                .select(['Name', 'Price', 'Rating'])
```

Wine	Price	Rating
Qta. Monte Xisto (2014)	56	4.5
Herdade dos Grous (Reserva 2015)	35	4.3
Dom Bella (Pinot Noir 2013)	48	4.3

2. Find producers for a highly requested dish: Bacalhau à Gomes de Sá:

```
find_producers = g.find("(a)-[]->(b); (c)-[]->(b)") \
                  .filter("a.Type = 'Producer' and  c.id = 'Bacalhau à Gomes de Sá'") \
                  .select(['a.id', 'b.id'])
```

Producer	Ingredient
Mar Lusitano	Bacalhau
Riberalves	Bacalhau
Horta do Adão	Batata
Quinta do Arneiro	Batata

Additionally, the solution can compute quantities for 75 dishes

```
dishes = 75

find_ingredient_quantities = df_unpacked.filter("Dish = 'Bacalhau à Gomes de Sá'").rdd \
                                        .map(lambda x: (x['Ingredient'], float(x['Quantity'])*dishes, x['Unit'])) \
                                        .toDF(['Ingredient', 'Quantity', 'Unit']) \
                                        .orderBy('Quantity', ascending = False)
```

Ingredient	Quantity	Unit
Bacalhau	37.5	kg
Batata	37.5	kg

3. Find nearest restaurants serving a goatling (Cabrito) dish from a traditional Ribatejo region:

```
find_restaurants = g.find("(a)-[]->(b); (b)-[]->(c); (c)-[]->(d); (e)-[]->(c)") \
                    .filter("a.id = 'Ribatejo' and d.id = 'Cabrito' and e.Type = 'Restaurant'") \
                    .select("e.id")

my_location = (38.732603, -9.160548)

find_nearest_restaurants = find_restaurants.join(df_restaurant, find_restaurants.id == df_restaurant.Name).rdd \
                                        .map(lambda x: (x['Name'], get_distance(my_location[0], my_location[1], x['Lat'], x['Lon']))) \
                                        .toDF(['Restaurant', 'Distance']) \
                                        .orderBy('Distance')
```

Restaurant	Distance
Apeadeiro	1.85
Forno Velho	1.88
As Salgadeiras	2.79

Limitations & Conclusions

As in any other project, there are some limitations that the group could identify that would otherwise increase its quality. To begin with, the identification of units and ingredients is based on a manual lookup list, ergo, it is required some governance in order to identify new entries. However, this limitation can be mitigated by providing exhaustive lists of units and ingredients, since these components can be considerably finite. The identification algorithm will improve its accuracy as long as the lists of units and ingredients keep being frequently updated.

Another potential improvement could be the use of a word similarity algorithm, as this will allow the identification of units or ingredients not present in their lookup lists. This upgrade will have to be considered in terms of computing, since there will always be a balance between an exhaustive lookup list and an effective similarity algorithm that identifies special cases or typos.

To conclude, this project aimed to not only analyze the Portuguese Gastronomy but doing it so by exploring the several capabilities of Apache Spark. Regarding the later, one may recall the natural language processing algorithm was built on a distributed paradigm allowing to reduce the computing time. As for the analysis, the graph serves as a base model to infer the most varied insights about the rich Portuguese gastronomy, that is with no doubt one of the most important drivers for the way this country is seen outside and within borders.