Wine Reviews - Information Retrieval

Rodrigo Abrantes, 201506561, MEIC, FEUP, U. Porto
Nuno Marques, 201708997, MEIC, FEUP, U. Porto
Nuno Santos, 201405774, MEIC, FEUP, U. Porto

This article provides information about our Information Processing and retrieval class with theme of wine reviews.

1 Introduction

This project aims to have an entire pipeline of processing and retrieval, so that the end user may search for wines based on many different attributes such as: taste, meals that go well with it and specific attributes such as acidity. This article is structured following our steps in building the project, which are: dataset definition and preparation, pipeline and information retrieval.

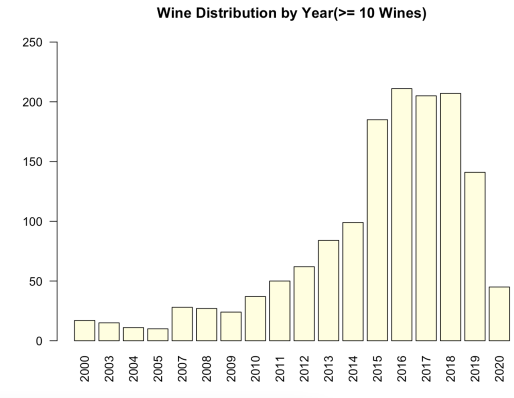

Figure 1: wine distribution by years with 10+ wines

2 Dataset definition and preparation

2.1 Final Dataset

Our group chose this theme, not only for the motive and logic of it's existence, but also for the available data we could find and how easily and efficient we could build a final dataset based on it, if the types of information would be interesting and complete for a later search. With that said we found several portfolios of wine available to compose the final data but the problem was every repository had different attributes so combining all sets would not be possible because on a search the result would have different types of information based on what set was the wine described. To overcome that, we decided to use scrapping on Vivino, a website with many wine entries and many information about each wine including several reviews. From ruffly 10 thousand entries we interpreted, we eliminated duplicates, selected the ones that had several English written reviews, eliminated those with missing or wrongly parsed attributes and got to a final Dataset consisting just short of 2000 wines and 15 reviews per wine. See figure 8.

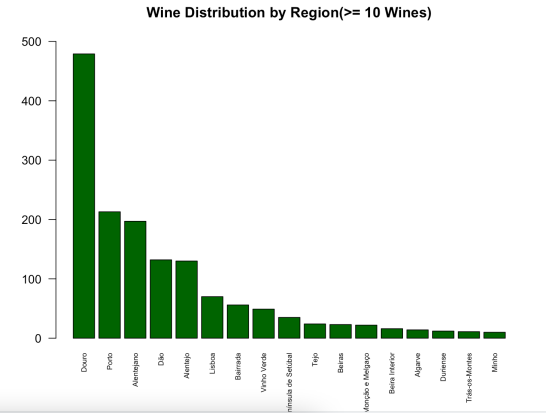

Figure 2: wine distribution by regions with 10+ wines

2.2 Data analysis

After the definition of the final dataset we needed to analyse the distribution on the data we had collected, to make sure we had wines present from several regions aside from several types, different ratings,etc.

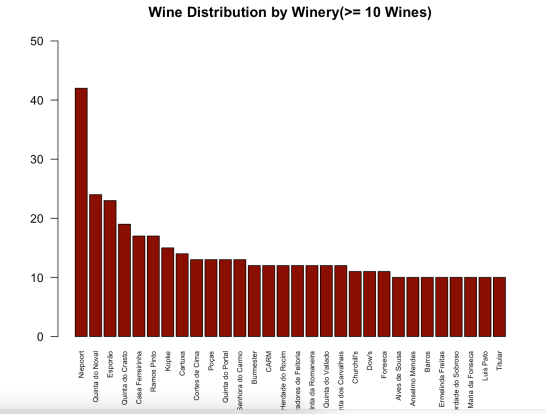

Figure 3: wine distribution by wineries with 10+ wines

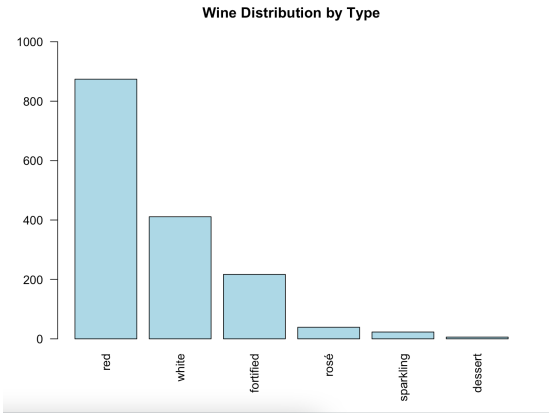

Figure 4: wine distribution by type

2.3 Present attributes in dataset

As for the actual type of data available in our dataset, it is divided in two files. The first, represented in Figure 1, we have knowledge of its id(the one it received when scrapped from vivino), the winery that produced it, the year it was produced, its name, the type(red, white, rosé, etc), the rating, price, country of origin and then more specific, the region and a link to an image of said wine. The second file(Fig 2) contains all the reviews for all the wines, with their rating, note and the user that made it.

vivino_id	winery	year	name	type	rating	price	country	region	image_url
2484848	Casa de Cello	2016	Casa de Cello Quinta de San Joanne Branco 2016	white	3.8	13.000000	Portugal	Vinho Verde	//images.vivir
2658364	Herdade da Ajuda	2011	Herdade da Ajuda Reserva Tinto 2011	red	3.9	11.500000	Portugal	Alentejano	//images.vivir
1722302	São João	2012	São João Lote Especial Tinto 2012	red	3.7	6.650000	Portugal	Bairrada	//images.vivir
2069631	Quinta do Romeu	2016	Quinta do Romeu Tinto 2016	red	3.7	12.000000	Portugal	Douro	//images.vivir
1744679	Costa Boal Family Estates	2011	Costa Boal Family Estates Flor do Tua Reserva Tinto 2...	red	3.8	75.000000	Portugal	Trás-os-Montes	//images.vivir
1894092	Xutos & Pontapés	2017	Xutos & Pontapés Tinto 2017	red	3.8	9.000000	Portugal	Alentejano	//images.vivir
1197707	Sociedade Agrícola de Pias	2011	Sociedade Agrícola de Pias Encostas do Enxoé Tinto 2...	red	3.6	25.000000	Portugal	Alentejano	//images.vivir
1504338	Quinta do Ortigao	2011	Quinta do Ortigao 4 Dezasseis 2011	red	4.4	30.000000	Portugal	Bairrada	//images.vivir
8348940	Fiuza	2018	Fiuza Reserva Premium Sauvignon Blanc 2018	white	3.7	7.990000	Portugal	Tejo	//images.vivir
6349654	Quinta de São Sebastião	2018	Quinta de São Sebastião Syrah 2018	red	3.4	7.330000	Portugal	Lisboa	//images.vivir
5264080	Quinta Maria Izabel	2018	Quinta Maria Izabel Vinhas da Princesa Branco 2018	white	4.4	47.440000	Portugal	Douro	//images.vivir
1621251	Mingorra	2017	Mingorra Terras d'Uva Rosé 2017	rosé	3.8	3.220000	Portugal	Alentejano	//images.vivir
80415	Dow's	1980	Dow's Vintage Port 1980	fortified	4.5	134.150000	Portugal	Porto	//images.vivir

Figure 5: first file with wine attributes

vivino_id	rating	note	user
2484848	4.0	A very nice Alvarinho 😊 An interesting fruit (and ber...	wine & mind
2484848	5.0	Had this wine in Lagos at the hotel we stayed in ... rea...	Caroline Hulsman
2484848	4.0	Great Alvarinho, loved it. Intense yellowish colour, no...	Martim Amaral Neto
2484848	4.0	Very very calm. Excellent balance, light and calm. Frui...	Aleksey Lisenkov
2484848	3.0	Candied lemon, hay, beeswax, chalky. Doesn't fulfil it...	Peter Arijs
2484848	3.5	Delicate aromas, very floral, light white melon, crisp a...	Rosanna Bucknill
2484848	3.5	✅ Still delivering after 5 years, this Alvarinho made b...	Marco Carmini
2484848	4.0	Citrusy and slightly sweet, smooth. A wonderful wine	Lauren Adie
2484848	3.5	Interesting nose of orange peel.	Mielies
2484848	4.5	A wonderful surprise. Aged and supreme.	Sergio Raposo Frade
2484848	4.0	Like a LDH white. Didn't know you could age a Alvahri...	Winston Chen
2484848	3.5	Excellent wine. Very good value for money.	Luis Ricardo Silva Viegas
2658364	4.0	This wine from alentejo has a soft elegant nose with a...	EdTheWineAdvocate
2658364	3.5	VERONA BELO HORIZONTE MG 3.3 Good QPR. Ruby r...	MARCELO BRANDÃO

Figure 6: second file with with the wine reviews

3 Pipeline

We chose Python for data acquisition and processing, and R to data cleaning and statistics/graphics creation, understanding and study. Our pipeline then consists of using Python to amass information through scrapping from Vivino, processing and storage in the appropriate files for later use. Several packages were needed to make this work, such as "Pandas". With the access to those packages we were able to clean the resulting dataset, removing duplicates, missing values,outliers,etc. Although the part of R is not included in the pipeline, we would then use R to build the statistics and graphs to better understand and comprehend the data we had for the rest of the project.

```
all: setup collect-data display-analysis

setup:
        pip install matplotlib
        pip install requests
        pip install pandas
        pip install progress

collect-data:
        python vivino_info_scraper.py
        python vivino_reviews_scraper.py

display-analysis:
        python vivino_analysis.py
```

Figure 7: Project pipeline script

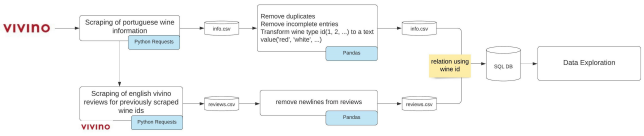

Figure 8: Project pipeline diagram

3.1 Objectives, search possibilities and quality of information retrieval

Although we are not working in this section as of yet, we theorized some possibilities for the later search on our data. We thought it would better shape the way we map our information and our sources if we knew what would be done with it later. The final data has attributes that must be present as a base search such as origin,rating , Winery, year,etc. But we felt that those were very basic searches so we needed to focus more on the reviews part because there the complexity of the search possibilities would be far higher. As seen in the Bigrams and Trigrams graphs(Fig 10 and 11) we can formulate the search engine based on the sequence of words present on the reviews, so sequences like "fruity taste", "hint of...",etc, that are present in the query will have good results.

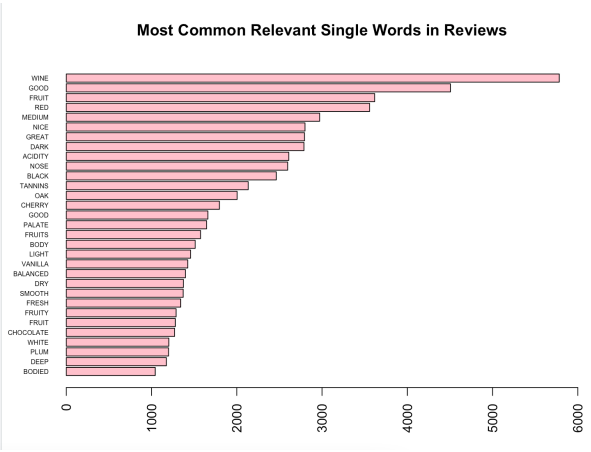

Figure 9: Most frequent words in reviews

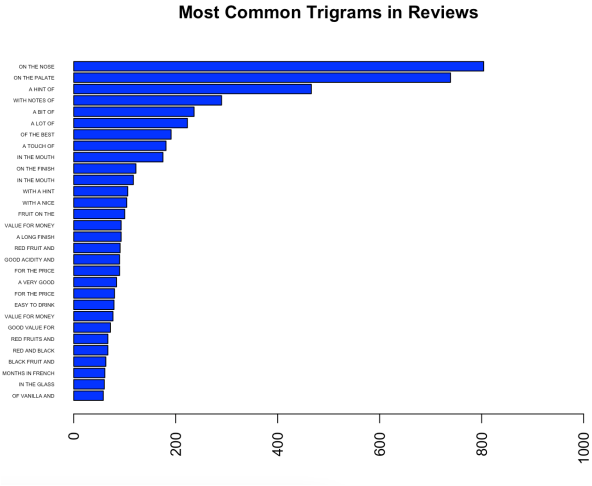

Figure 10: Reviews statistics

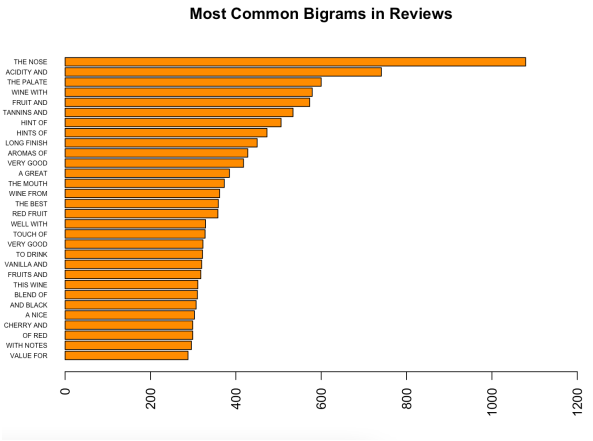

Figure 11: Reviews statistics

4 Information Retrieval

4.1 Information Indexing Process

To index our datasets we used *solr* a distributed indexing and load-balanced querying tool. Before adding our data collection to solr we aggregated our individual reviews for each wine in a single field. This way adding our data to solr was a much easier process. A schema file to specify how our data was to be indexed was also needed. We created a schema with various tokenizers. For all textnfields we used solr's *Standard Tokenizer, ASCII Folding Filter and the Lower Case Filter* in order to normalize them. For the note field that contains our reviews we used solr's *Porter Stem Filter* that more aggressively normalizes our text. For the text "the foxes jumping quickly" this filter outputs [the, fox, jump, quickly].

Afterwards we created an automated script to create a core, add a schema and finally add our data collection to our solr docker container.

```
reset_core.sh:
#!/bin/bash

docker exec pri_solr bin/solr delete -c wines
docker exec pri_solr bin/solr create_core -c wines

curl -X POST -H 'Content-type:application/json' \
--data-binary @schema.json \
http://localhost:8983/solr/wines/schema

curl
    'http://localhost:8983/solr/wines/update?commit=true'
    --data-binary
    @../datasets/vivino_reviews_final.csv -H
    'Content-type:application/csv'
```

With this our solr core is populated, indexed and ready to be queried.

4.2 Information Retrieval Process

For the retrieval process with used solr's query parser. An endpoint is provided by solr for each query so we can later analyse our query results with a python script. The query parser allows many types of searches, for our use we found *Term Boosting and Proximity Searches* to be the most effective. To test this we tried finding reviews that mention the wine being "value for money". Using *note:"value" AND note:"money"* as a query wielded 303 results, many reviews had money and value in it but not necessarily together. Next we tried the proximity search, *note:"value money"~5* searches for both words in proximity of 5 words, this reduced our results to 234 but it still had many irrelevant reviews such as "value for money may be questionable". So we tried adding good to the proximity search *note:"good value money"~5* and found relevant 143 results.

To further text our retrieval process we now wanted a good value for money wine that goes well with chicken, using this query: *note:"good value money"~5 AND note:"chicken"^5* now we also want chicken mentioned and boosted the term chicken so wines that have many mentions of chicken get ranked higher. This wielded just 13 relevant results. The highest ranked wine contained

the following in the note field: "... Appropriately, I had it with grilled chicken kebabs... Good value for money in Portugal".

We were now ready to move on to making queries based on our information needs and analyse our search results.

4.3 Information Needs

As for contextual examples where our search engine would be useful we must consider the situations where it would be needed, so we studied our information retrieval based one some scenarios:

"Wine with a taste of green apples": Users will often want to find wines that have a certain specific taste.

"Wine that goes well with a chicken meal": Users will often want to find wines that are a great match to the food they are having.

"I'm quite full and I'm looking for a light wine for a summer afternoon": Users will often want to find a wine with a specific weight, suited for a drink at a specific time of day.

4.4 Evaluation

To measure the quality of our search results we tested queries with and without enhancement and calculated their **precision** and **recall** in python this is the results we got:

4.4.1 Taste Query

We started evaluating our system by using a simple search of *"green apples"* for a user looking for a wine with such a taste. Without enhancing our query in solr we found that results had both little precision and recall. This was expected as reviews may mention *green* in relation to it's color and *apple* to describe it's smell for example, including the very first result which was a false positive.

Base Results:

Metric	Value
Average Precision	0.189771
Precision at 10 (P@10)	0.2

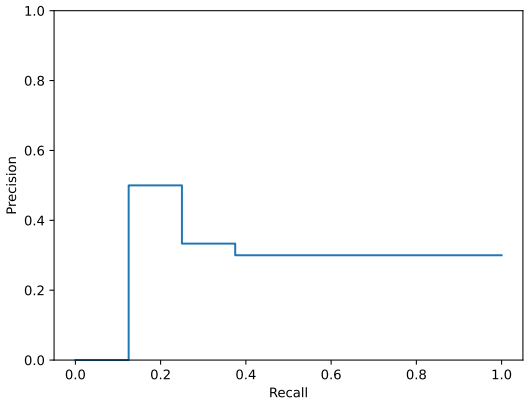

Figure 12: Reviews statistics

In order to enhance our system we added a proximity search to find reviews that mention *green apple* specifically about the wine's taste, by searching for the words *"taste", "green", "apples"* in close proximity and term boosting *"green apples"* the precision and recall of the query was much improved:

Metric	Value
Average Precision	0.884921
Precision at 10 (P@10)	0.6

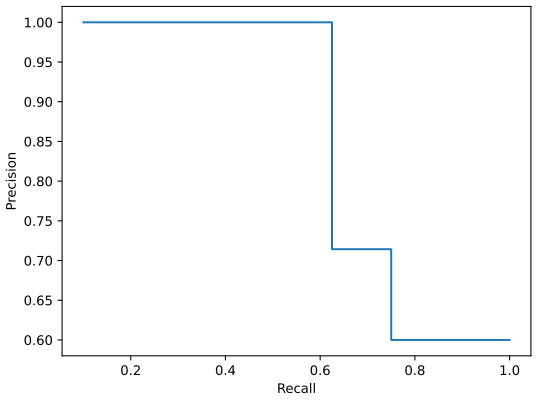

Figure 13: Reviews statistics

Query using other taste terms such as: *vanilla, peach, citrus, cherry, etc* wielded similar results.

4.4.2 Dish Query

Our next evaluation will fall on an information need related to wines that go well with certain dishes. Lets imagine a user wants to have a dinner, where the protein is *chicken*, and wants to find a wine that he can serve that will go well with the dish prepared. We start by performing a simple search for *chicken*. Without any enhancing of the query, the results we get all refer to chicken in some way, but there is no way to know if the comment made about *chicken* is actually a positive, neutral or negative review. So, as expected, the results won´t be as helpful as they could in finding what we consider the best choices to accompany *chicken* dishes.

Base Results:

Metric	Value
Average Precision	0.060626
Precision at 10 (P@10)	0.1

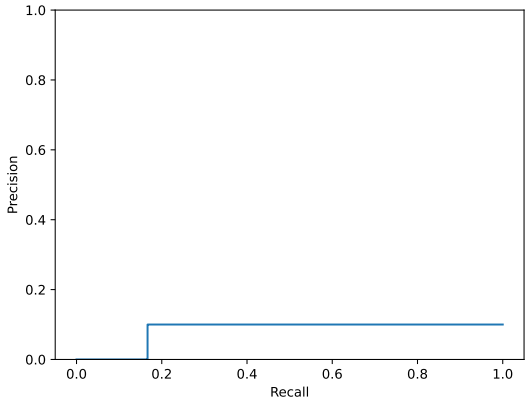

Figure 14: Reviews statistics

In order to enhance our search results, we added to the query a proximity search for the words *good*, *great*, *perfect* and *excellent* next to *chicken*. We also added term boosting so that *good chicken* will be the less relevant choice, *great chicken* the intermediate choice, and both *perfect chicken* and *excellent chicken* the most relevant one . By doing that, the precision of the search results greatly improves, as show below.

Enhanced Results:

Metric	Value
Average Precision	0.919312
Precision at 10 (P@10)	0.6

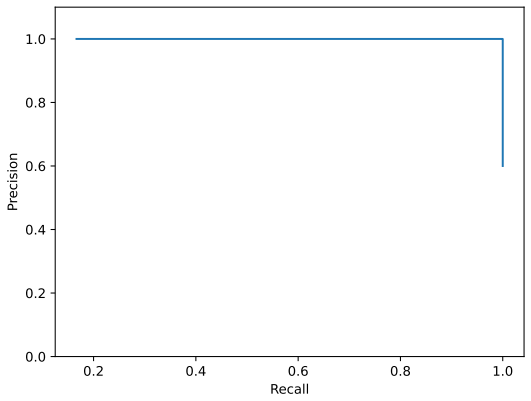

Figure 15: Reviews statistics

Query using other dish terms such as: *salmon, codfish, veal, lamb, etc* wielded similar results.

4.4.3 Wine body Query

Our third evaluation is based on a situational need, in that we're looking for wines that are light and we can drink on a summer afternoon. For this we divide the question in three: a "light wine", one that goes well on the "afternoon" and then to be more specific because that may change the results, a wine for a "summer afternoon". However, considering the results for this information need, we realized that the results we got from a simple search with "light wine" were the same for a search with "light","wine" and "afternoon". Even with an enhanced query, the reviews containing the most occurrences of "light wine" were the same that referenced the "afternoon". Using that same base query and comparing the results with an enhanced query using adjectives for the afternoon we had the following results:

Base Query:

Metric	Value
Average Precision	0.424956
Precision at 10 (P@10)	0.2

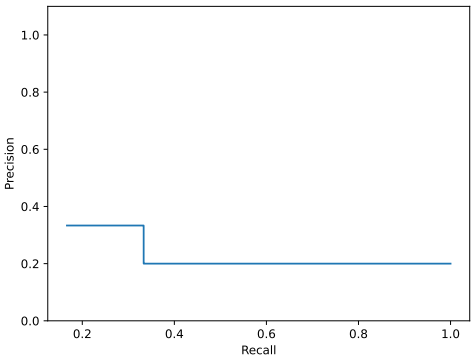

Figure 16: Using Solr - note:"light" AND note:"afternoon" AND note:"wine"

Enhanced Query:

Metric	Value
Average Precision	0.919312
Precision at 10 (P@10)	0.6

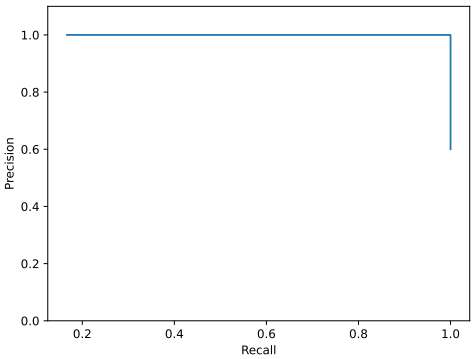

Figure 17: Using Solr - (note:"good afternoon"~5^5 AND note:"light"^5) OR (note:"great afternoon"~5^10 AND note:"light"^5) OR (note:"perfect afternoon"~5^20 AND note:"light"^5) OR (note:"excellent afternoon"~5^20 AND note:"light"^5)

5 Conclusion

Our current pipeline allows us to find appropriate search results using a pipeline with many steps from data collecting and processing, to indexing and retrieval with satisfactory results.

6 References

1 - Solr's website. URL: https://solr.apache.org/

2 - Elastic Porter Stem guide.URL: https://www.elastic.co/guide/