Conversational Wine Recommender System
Summary

Thursday 18th June, 2020 - 15:12

Flavio De Jesus Matias
University of Luxembourg
Email: flavio.dejesus.001@student.uni.lu

Sviatlana Höhn
University of Luxembourg
Email: sviatlana.hoehn@uni.lu

***Abstract*—This document presents the bachelor semester project of Flavio De Jesus Matias under the tutoring of Sviatlana Höhn. The project consists of a conversational wine recommender system, which will summarize data about wine reviews using text summarization techniques and provide the user with a chatbot to which he may ask questions about an arbitrary wine.**

1. Introduction & Project description

Chatbots are a very important tool for most online businesses today since they rely on their virtual assistants to handle basic user requests or questions. In general, chatbots only present a small amount of textual data to the user to avoid overloading them and therefore the importance of quality summarization has increased over time.

Thus, the necessity for an effective and automatic summarization system has also increased. Writing summaries by hand is an extremely time-consuming and exhaustive process and therefore it is important to have well-performing algorithms to analyse and process lots of textual data into small readable summaries.

The objective of the project is to build an algorithm to summarize wine reviews. Then a conversational wine recommender system will retrieve the summaries on user demand and display them.

2. Background

Both parts of the project required the programming language Python. However, to build the summarization algorithm, knowledge in summarization techniques was needed. As learning about multiple summarization techniques is part of the learning goal of the project, this was rather learned during the project than right at the beginning.

Apart from having knowledge of summarization techniques, it was more important to make a choice and choose the right approach for the project. The summarization technique chosen was the "extractive summarization", which is the simplest and most used approach to generate summaries.

3. Requirements

3.1. Generate a summary out of a set of reviews

In order to function correctly, the algorithm needs to be able to process an arbitrarily long input of reviews and transform it into a summary. Each summary must be generated according to the standards needed for later usage of the chatbot.

3.2. Natural language understanding & retrieval

The NLU model has to recognize the different wine sorts and be able to extract custom entity values from the user's request. The chatbot has to identify the intent in the user's request correctly, even if the desired wine sort is not contained in the training examples. From the detected entity values, the chatbot has to retrieve and deliver the corresponding summary to the user. If the wine sort is not present in the database, then the default answer from the chatbot is returned or similar wine sorts are suggested to the user.

4. Design & Production

4.1. Scientific Deliverable

The scientific part of the project consists of developing an algorithm to generate a dataset of summaries from wine reviews. The algorithm was separated and organized into 4 different files (see figure 1).

4.1.1. data.csv. The file data.csv contains the initial dataset of the algorithm. With approximately 130 thousand rows of different wine reviews, this gives the project a solid base to work with.

4.1.2. SummarizerClass.py. This file does all the summarization work. All the reviews passed to this class are split into sentences and now that we have a list containing all the sentences we just need to score them. There are **two**

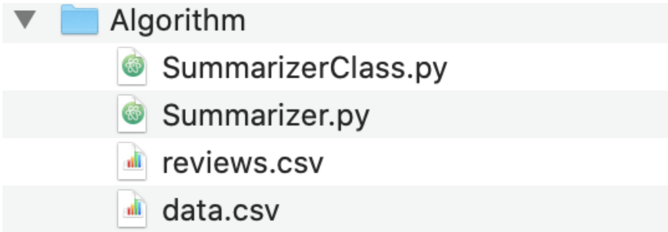

Figure 1: File structure of the algorithm to generate summaries from wine reviews

crucial aspects when it comes to scoring the sentences: the sentence length and the Tf-idf score, where the Tf-idf score counts **twice** as much as the sentence length. How exactly the score for both of these parts is calculated is explained in detail in the final report. Now that we have a list containing all the sentences and the sum of its sentence length and Tf-idf score, we can create the final summary that consists of the n best-scored sentences in this list.

4.1.3. Summarizer.py. This file acts as the bridge between the old and the new dataset. The first thing the file does is to load the old dataset, i.e. the start dataset. Then it removes all the duplicates and all the wine sorts that have less than 20 reviews. The dataset we are working with is shown in figure 2.

After cleaning the dataset, we iterate over all the remaining wine sorts and initialize the summarizer class with a list of all the reviews for this specific wine sort. The last step in this file is to write the generated summary in the new dataset, which is the reviews.csv file.

4.1.4. reviews.csv. The file reviews.csv contains the final dataset created by the summarizer algorithm. The structure of the file is kept very simple, it only has **3 different columns**: wine name, number of reviews and the summary itself.

4.2. Technical Deliverable

The technical deliverable of the project consists of building a conversational wine recommender system, i.e. a chatbot.

The chatbot was created using Rasa, which is an open-source conversational AI framework. After setting up the default chatbot, every unnecessary part of the code was removed. Then, the first thing to do was to set the domain of the chatbot by adding a new intent (wine name) and new actions (to request information about a wine name). Secondly, stories which represent real conversation data between a user and the chatbot were added. Thirdly, the NLU model needed to be trained. For every wine sort, 7 different sentence/question templates were added. Then, every example was added twice: with capitalized initial letters and with lowercase initial letters. This covers the whole database including capital and lowercase examples and since we have **227** different wine sorts, this equals to $7 * 2 * 227 = 3178$ training examples in the NLU model. Lastly, a new custom action was added to retrieve the summaries out of the database and display them to the user.

5. Assessment & Conclusion

The multiple requirements set at the beginning were successfully achieved since the algorithm can successfully generate summaries out of the wine reviews. The chatbot can also identify custom entity values that are not present in the training examples as well as retrieve available summaries out of the database when available.

In the future, the project could be improved in several ways such as adding new information about every wine sort, allowing the user to get more details about their requested wine sort. Furthermore, the chatbot could be linked to a social media account of a wine selling company. By connecting the chatbot with the Facebook page of the company, it could answer the user's questions and help them without having to wait for a real employee.

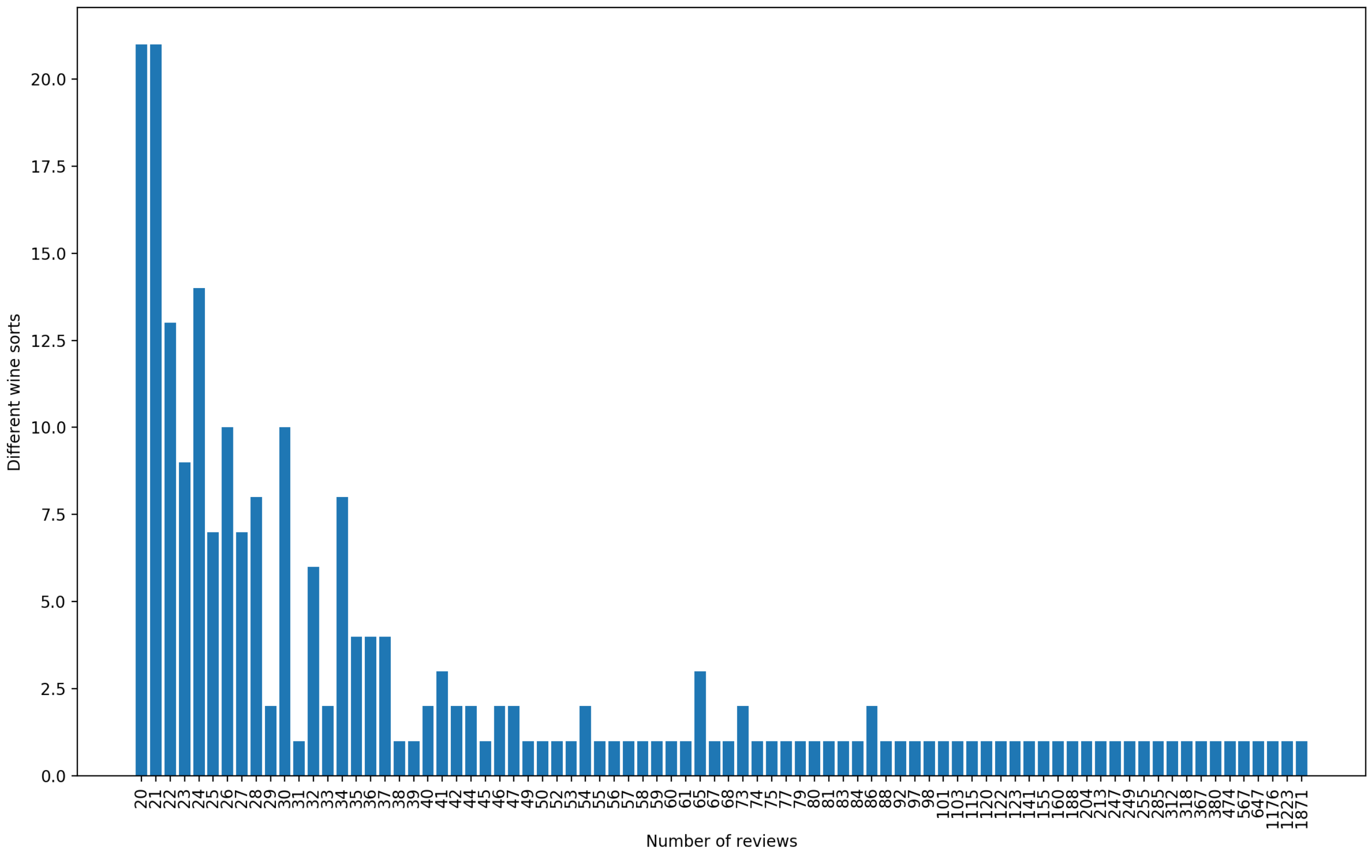

Figure 2: Number of reviews per different wine sorts