Beer Advisor - A beer ontology

Lucas Standaert[1], Anna Yaroslaski[2], Marcelo de Castro[3]

Rensselaer Polytechnic Institute[1,2,3]
standl@rpi.edu[1], yarosa2@rpi.edu[2], decasm3@rpi.edu[3]

Abstract

Due to the wide variety of beers with different flavor profiles and styles, choosing a beer is not a simple task. The goal of the application introduced in this paper is to provide users with beer recommendations, to simplify the selection process. By matching their preferences with attributes of commercially available beer, we can provide users a collection of beers they will likely enjoy. There are a number of existing ontologies and accompanying applications that serve a similar function, however few have solely targeted beer. Therefore, the current work aims to show how a beer ontology focused on providing recommendations can be implemented and used in an application. This current report also presents the technical challenges related to the implementation of such ontology. The ontology is then evaluated using a set of competency questions and results are going to be presented.

Introduction/Motivation[1]

Beer is said to be one of the oldest alcoholic drinks created by humans. In fact, we have been creating different types of beers for millennia. Due to the wide variety of beers with different flavor profiles and styles, choosing a beer is not always a simple matter. In addition, a number of craft breweries have emerged that have contributed a wide assortment of quality beers. This mass quantity of beers has led to conflicting style guidelines that label beers incorrectly. A consumer's preferences towards certain styles and brands of beer can also have a major impact on what beer they buy. Providing them with choices made by users with similar preferences can help to expand their list of potential beers. Overall knowledge of beer is another factor that can influence a consumer. While amateur beer drinkers may simply judge based upon style, beer aficionados have a much higher standard and may wish to narrow their search further. Our goal is to create an ontology that will help to resolve these discrepancies between databases, and an application that can assist both experienced and inexperienced beer drinkers.

Use Case

The ontology-enabled application developed in this work aims at providing users with beer recommendations that match their preferences with attributes of commercially available beers, including local craft beers. The application combines information from different databases in order to find listings of beers that can then be organized by the beer ontology and information can be retrieved using SPARQL queries. The users might specify different sets of characteristics that they want to find in a beer. The first set is related to beer intrinsic characteristics, which include alcohol content, measured by Alcohol By Volume (ABV), bitterness. determined by the International Bitterness Unit (IBU), sweetness, measured using Original Gravity (OG) and color, using the Standard Reference Method (SRM). Ingredients are also features that belong to this set of intrinsic characteristics. The second set is related to extrinsic characteristics of beers such as its name and the brewery that produced it.

Hence, it is clear that this application has mainly two types of stakeholders that will have roles as actors. The first type of stakeholders are the primary actors that effectively use the ontology-enabled application looking for a beer. Examples of these actors include a beer drinker customer, a beer store, a bar and a beer distributor. The second main type of stakeholders will play a secondary role in the application by providing the beer data that will be organized by the ontology. Beer databases such as OpenBeerDB and beer.db are examples of this set of actors. In addition to that, breweries are important stakeholders that might not perform any action in the application.

In order to illustrate how the application can be used, three different usage scenarios of the beer recommender application are presented next.

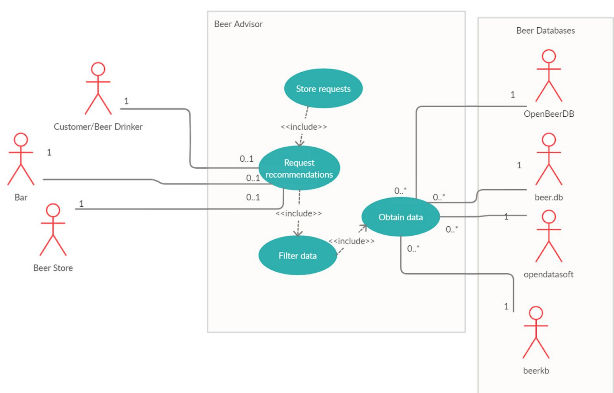
Scenario 1: A person has just moved to a new town and they are looking for local beers as they want to support microbreweries that are located in that town. The person then selects a type of beer, say India Pale Ales (IPAs), and selects the town. The application should return the specified type of beer produced by breweries located in that town.

Scenario 2: A bar in Pittsburgh is looking for expanding its selection of beers by adding a local beer to it. However, they want winter beers, which are usually dark-colored, that have a maximum of 7% ABV. The owner can then ask for beers that are made from breweries located in Pittsburgh which have their alcohol content in the previously established ABV limit.

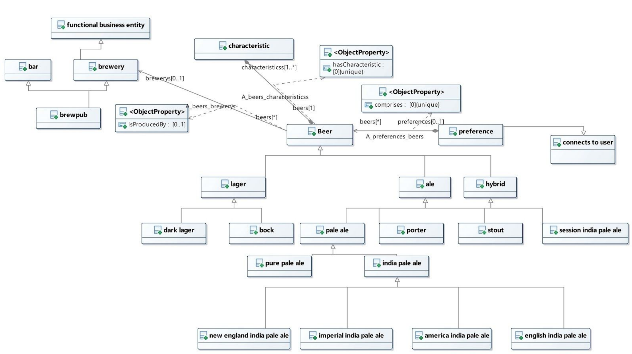
Scenario 3: A Beer Advisor user has already found a nice list of beers using the application. However, they feel that they have limited their list too much due to characteristics that might be too specific. Therefore, they expand their beer list by checking out what beers appear in other user's search history.

More details on usage scenarios and general information about our use case can be found through the link below. https://beer-advisor--rpi-ontology-engineering.netlify.app/oe2020/beer-advisor/usecase.

Technical Approach

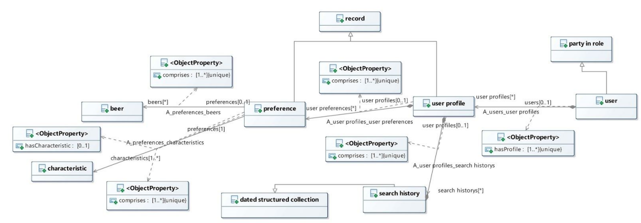


We first start with a look at the overall architecture of our application. The application will first draw from a list of established beer databases, and then create a set of individuals from this. From here, these individuals will be sorted utilizing inference rules in our ontology. Users will then be able to make requests to the application which will utilize our ontology to filter the data and return the results to our users. These searches will be stored for later use. Users can also be recommended beers based on their preferences and the preferences of other users.



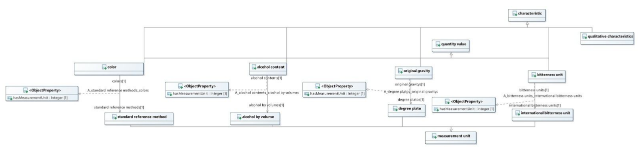
Our beer class inherits the fibo ontologies product class through the use of the beverage and libation class. These come into play later in discussion of our brewery class. Each beer has a specific alcohol content, bitterness unit, color, original gravity, and some ingredients. Beers must also be produced by a brewery. From here, every type of beer is instantiated as a subclass of the beer. The ontology then sorts beers into their specific styles utilizing the different ranges and ingredients that are defined later in the ontology. To help distinguish these beers from their sibling classes, they are made disjoint.

The brewery class is instantiated as a subclass of the FunctionalBusinessEntity class that is defined within the fibo ontology. Each instance of the brewery class has a name and an address and produces some form of ber. This allows us to query for certain breweries based on location, specific beers made by a brewery, and a combination of the two.

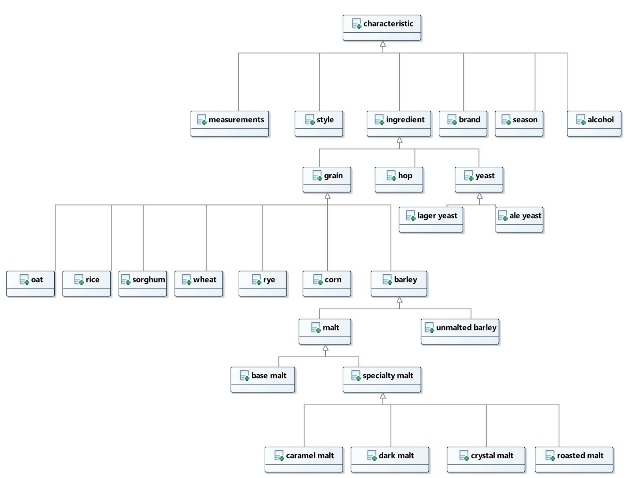


In our attempts to differentiate our ontology from other existing ontologies, we came across the idea of creating users in our ontology. This makes it one of the most important parts of our ontology. Our user class leverages the party in role class, defined in the fibo ontology. Each user must be a person and must have a user profile. The user profile is how our ontology stores the data from each user. Inside of a user profile is stored their search history and their user preferences. The search history is a dated collection, which allows us to search for more recent queries from users. In addition to this, each user profile has a user preference, which also inherits from the record subclass. Each user preference is a combination of beers and characteristics, which allows us to query and compare users based on both their search history and preferences.

In order to assist us in storing a user’s preferences, preferences are defined within our individuals ontology. Each preference is defined based upon a specific characteristic and is given a range. For example, a preference towards a sweet is beers that have an original gravity above 1.100. This allows us to also easily compare users as this set of preferences is shared amongst them. This will also assist us in a rating system we have planned, where users will be able to provide beers with a specific rating.



The final major talking point is our characteristics. The main characteristic class is a subclass of the fibo aspect class. Inside of this class, each characteristic that describes our beer is instantiated as a subclass. The four main characteristics are the original gravity, alcohol content, bitterness unit, and color characteristics. Each of these classes is a subclass of the quantity value subclass, allowing us to give each one of these measurements a quantitative value. Each measurement is then also given a specific unit, alcohol by volume, degree plato, international bitterness unit, and standard reference method. From here, each beer can have a specific range defined by through these values.



In addition to this, beers also have a wide assortment of other characteristics, including ingredients. These are more qualitative characteristics, so we decided not to focus too heavily upon them, as the above characteristics were readily available.

For more details about the structure and our conceptual models, they can be found using the link below.
https://beer-advisor--rpi-ontology-engineering.netlify.app/oe2020/beer-advisor/ontology

Related Work

The development of semantic web bolstered the number of applications involving all different sorts of ontologies. Therefore, it is not surprising to find many ontologies related to food and beverage. These ontologies are often used as a tool that enables an application rather than the end use of a project [Ontology101]. This is because there is plenty of information on the web stored in different databases that needs to be structured. Ontologies, then, come in hand and can be used to organize that knowledge about a certain food or drink.

In fact, one of the most famous ontologies is the wine and food ontology presented in [Wine]. It provides ways of classifying wines and of suggesting proper meals that accompany well a particular wine. Hence, an application that leverages such ontology could be used in different scenarios; a restaurant that wishes to list the wines that will properly accompany a new dish in the menu could use such application. Or, in addition, a wine aficionado could look for the perfect wine that will match his favorite steak dish. The ontology is presented as having three overarching classes that are disjoint: red, white and rose wine. These overarching classes are then parent to many other children classes that will serve as more specific classification for the wine. For instance, Wine has subclass Red wine that has a child class Red Bordeaux that can be, for example, a Graves [Wine]. This type of classification approach was used as a basis for classifying beers in the ontology presented in this current report, where some overarching classes were used to comprise even more specific subclasses.

However, the wine and food ontology, as one would expect, covers only wines as beverages. On the other hand, Bevon, a beverage ontology, can be used to classify many other drinks, including beers [Bevon]. The ontology contains classes for different types of beers such as Irish Red and Pilsner and it tries to organize its beers by using two overarching classes: Lagers and Ales. These two types of beers are not considered disjoint in Bevon. In addition to that, our current ontology has also a third overarching class: Hybrid beers. Therefore, beers that do not fit exactly into Ales or Lagers can be classified as Hybrids, such as Lambics or even Session beers. Bevon, on the other hand, would need to be extended in order to address this type of issue. Moreover, Bevon has two characteristics that are attributed to beer: color (via SRM) and bitterness (via IBU). This was also adopted in our ontology but other characteristics such as ABV and sweetness were also added. Bevon also presents a property that can represent a brewery as a characteristic of a beer. This simple solution to link a beer to its brewery, however, does not allow the representation of more information about the brewery, such as its location. Although Bevon has many interesting attributes to beers, the ontology has had its development stalled since 2015 and, therefore, it presents many limitations such as the lack of ingredients from which beer is brewed.

Ontologies such as the ones presented in [Beer-Ont-1] and [Beer-Ont-2] address this problem by adding properties that link beers to specific ingredients such as hops, malts and yeasts which can be classified into different types. The ontology in [Beer-Ont-1] is organized in a suitable way for classifying beers into many different styles and its ingredients. It also presents a characteristic that allows awards to be linked to beer individuals. However, it does not present any beer attributes that can be used to add characteristics to beers. In addition to that, the classes of beers are not very coherent since Lager and Pilsner are considered different classes without any relation between them but, in reality, a Pilsner should be considered a child class of a Lager. The ontology in [Beer-Ont-2], on the other hand, presents a set of attributes that allows alcoholic content and sweetness to be represented. In addition, [Beer-Ont-2], as the ontology presented in [Bevon], also allows beers to be linked to breweries. In [Beer-Ont-2], however, breweries belong to a dedicated class and, therefore, more information can be attributed to it.

It is easy to see that many ontologies related to beverages were developed in the last years and that a few are especially concerned with beers. However, none of them fully address a complete classification of beer characteristics, for example. Furthermore, none of them allow the representation of beer classes that do not particularly fit the overarching classes Lagers and Ales. We can cite Lambics or even Session beers as examples of such beers that are hard to classify. None of those ontologies approached the idea of a user that could be interpreted as a beer drinker and that might have a set of beers that are preferred.

In this context, our beer ontology is developed to address all these issues, setting more characteristics to beers that are classified in a more coherent manner. For example, a beer can be connected to a list of ingredients and characteristics such as bitterness, sweetness, color and alcohol content can be attributed to beer individuals. Moreover, the ontology presented here also allows the representation of breweries which can be linked to its produced beers. The ontology presented here also allows the representation of a user class. This class is focused on the possibility that the ontology can be used in an application. The user can have preferences linked to its profile and, therefore, this information is also available to be queried.

Evaluation

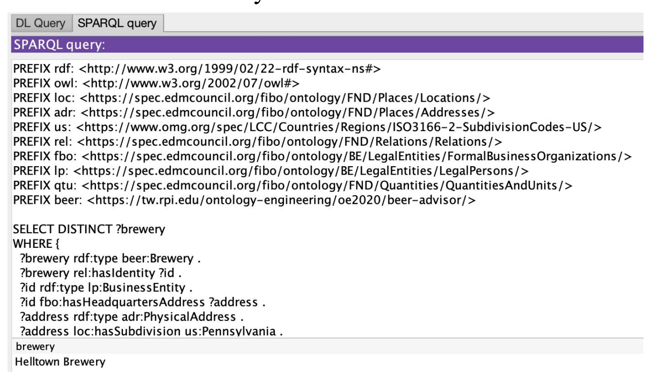
During the course of working on this project we have identified five questions that we can use to evaluate the functionality and completeness of our application. The questions and their implementation stage are as follows:

1. What are types of winter beers that are under 8% alcohol content? (Supplementary)

This is a basic lookup problem inside of our ontology. This question is currently supplementary as we were focused upon the 2nd, 3d, and 4th competency questions as our main focus.

2. What is a brewery in Pennsylvania that makes IPAs under 8%? (Active)

Due to confusion surrounding our first competency question, this question targets the lookup ability of our ontology. Utilizing our query below, the ontology should return Helltown brewery.



```
DL Query | SPARQL query
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX loc: <https://spec.edmcouncil.org/fibo/ontology/FND/Places/Locations/>
PREFIX adr: <https://spec.edmcouncil.org/fibo/ontology/FND/Places/Addresses/>
PREFIX us: <https://www.omg.org/spec/LCC/Countries/Regions/ISO3166-2-SubdivisionCodes-US/>
PREFIX rel: <https://spec.edmcouncil.org/fibo/ontology/FND/Relations/Relations/>
PREFIX fbo: <https://spec.edmcouncil.org/fibo/ontology/BE/LegalEntities/FormalBusinessOrganizations/>
PREFIX lp: <https://spec.edmcouncil.org/fibo/ontology/BE/LegalEntities/LegalPersons/>
PREFIX qtu: <https://spec.edmcouncil.org/fibo/ontology/FND/Quantities/QuantitiesAndUnits/>
PREFIX beer: <https://tw.rpi.edu/ontology-engineering/oe2020/beer-advisor/>

SELECT DISTINCT ?brewery
WHERE {
  ?brewery rdf:type beer:Brewery .
  ?brewery rel:hasIdentity ?id .
  ?id rdf:type lp:BusinessEntity .
  ?id fbo:hasHeadquartersAddress ?address .
  ?address rdf:type adr:PhysicalAddress .
  ?address loc:hasSubdivision us:Pennsylvania .
brewery
Helltown Brewery
```

If we examine the results of this query, we find that we do obtain our expected results.

brewery
<https://tw.rpi.edu/ontology-engineering/oe2020/beer-advisor-individuals/HelltownBrewery>

3. What is an IPA that is 5% ABV or below? (Active)

The goal of this question is to test our similarTo relationship. The application will begin by calling the following initial query:

SPARQL query:

```
SELECT DISTINCT ?beer

 WHERE {
  ?beertypes rdfs:subClassOf* beer:IndiaPaleAle .
  {
   ?beer rdf:type ?beertypes .
  }
  UNION
  {
   ?beer rdf:type beer:IndiaPaleAle .
  }

  ?beer beer:hasAlcoholByVolume ?alcohol .
  ?alcohol rdf:type beer:AlcoholContent .
  ?alcohol qtu:hasNumericValue ?abv .
  FILTER (?abv <= 5)
 }
```

beer

However, there are no IPAs that have an ABV of 5% or less, so this query would return no results. Rather than simply returning no results, we would like to point the user toward beers that are similar to the one that they requested. As such, whenever the query returns no results, the application should call a secondary query which is as follows:

```
SELECT ?beertypes ?beer

WHERE {
 beer:IndiaPaleAle beer:similarTo ?beertypes .
 ?beer rdf:type ?beertypes .
 ?beer beer:hasAlcoholByVolume ?alcohol .
 ?alcohol rdf:type beer:AlcoholContent .
 ?alcohol qtu:hasNumericValue ?abv .
 FILTER (?abv <= 5)
}
```

beertypes	beer
session india pale ale	Voodoo Ranger American Haze

4. I really like New Belgium's IPA's, what other beers have people searched for from New Belgium? (Active)

This question is our baseline for testing similarities between users. In this scenario, the ontology will compare different users' search histories and return beers from the New Belgium brewery that are found in others search histories. Utilizing our current ontologies, the results should be the Glutiny Pale Ale, the Fat Tire Belgian White, the Fat Tire Amber Ale, and the Voodoo Ranger American Haze. If we utilize the query below and compare it with the results beneath it, we find that it does match up.

DL Query | SPARQL query

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX loc: <https://spec.edmcouncil.org/fibo/ontology/FND/Places/Locations/>
PREFIX adr: <https://spec.edmcouncil.org/fibo/ontology/FND/Places/Addresses/>
PREFIX us: <https://www.omg.org/spec/LCC/Countries/Regions/ISO3166-2-SubdivisionCodes-US/>
PREFIX rel: <https://spec.edmcouncil.org/fibo/ontology/FND/Relations/Relations/>
PREFIX fbo: <https://spec.edmcouncil.org/fibo/ontology/BE/LegalEntities/FormalBusinessOrganizations/>
PREFIX lp: <https://spec.edmcouncil.org/fibo/ontology/BE/LegalEntities/LegalPersons/>
PREFIX qtu: <https://spec.edmcouncil.org/fibo/ontology/FND/Quantities/QuantitiesAndUnits/>
PREFIX beer: <https://tw.rpi.edu/ontology-engineering/oe2020/beer-advisor/>

SELECT DISTINCT ?beer
WHERE {
 ?searchhistories rdf:type beer:SearchHistory .
 ?searchhistories rel:comprises ?beer .
 ?beer rel:isProducedBy ?brewery .
 ?brewery rdfs:label ?brewlabel .
 FILTER (?brewlabel = 'New Belgium Brewing'@en)
    }
```

beer
Fat Tire Amber Ale
Glutiny Pale Ale
Voodoo Ranger American Haze
Fat Tire Belgian White

beer
<https://tw.rpi.edu/ontology-engineering/oe2020/beer-advisor-individuals/FatTireAmberAle>
<https://tw.rpi.edu/ontology-engineering/oe2020/beer-advisor-individuals/FatTireBelgianWhite>
<https://tw.rpi.edu/ontology-engineering/oe2020/beer-advisor-individuals/GlutinyPaleAle>
<https://tw.rpi.edu/ontology-engineering/oe2020/beer-advisor-individuals/VoodooRangerAmericanHaze>

5. Is there a stout made by a local brewery in Idyllwild, California? (Future)

Due to time constraints, we decided not to focus upon this query as it would bring up a variety of different issues through location. However, in the future, this is something we plan to implement.

These five competency questions can be found in their entirety through the following link: https://beer-advisor--rpi-ontology-engineering.netlify.app/oe2020/beer-advisor/usecase.

To further ensure the quality and effectiveness of our ontology, we need to add additional measures. In order to do this, we've come up with a set metrics to help us gauge this. First we'll look at the ontologies effectiveness in inference beer styles based upon the criteria it is given. Using smaller sample sizes, we utilize this preset data and test to see if our ontology can accurately access what style of beer each individual is. By comparing the number of correct answers to the number of incorrect answers, we can tell where the ontology may have difficulty ascertaining the style and fine tune the ontology to fix this.

The second metric we will be looking at will be in relation to our similarTo property. Currently this property is in its very early stages due to time constraints and we are simply tagging related beer styles. In the future we would like to expand this metric to include brewery, location, original gravity, and other important characteristics. We hope to assign scores to these beers and return a ranked list back to the user. To measure this, we will manually compute scores of beers before they enter the ontology. After this, we will compare these scores to those in the ontology to confirm they are correct, keeping track of which scores are not.

The final metric will be, answer this with the group.

Discussion

- Using semantics we are able to identify what style a beer is even if it is not explicitly stated. It also allows us to universalize the data from different beer databases.
- Additionally, the semantics enable our program to any user with any level of knowledge on beer, instead of just those with a lot of knowledge or those with very little. While an application geared

toward either population would be useful in its own right, an application that can function equally well for both is far more convenient.

See our project page for more information: https://beer-advisor--rpi-ontology-engineering.netlify.app/oe2020/beer-advisor/.

Future Work

Looking towards the future, we still have a few things we will need to work on. The first is general expansion within our individuals rdf and our beer styles inside of our main rdf. We want to expand our main styles of beer to include a multitude of different beers, as this would allow us to further specify different styles of beer. We would next like to incorporate the many databases we have inside of our individual ontology. We also want to incorporate a hierarchy for ranking. Currently our system looks for relatively basic similarities, such as styles and breweries. We want to create a much better representation for the similarities between users so the main goal here is to create a ranking system that allows us to do this. Finally, we want to launch and maintain our app to the public. This will be done after a significant period of testing to make sure that our ontology is returning the proper results.

Conclusion

Use of the section heading style is required.

References

[Ontology101] Noy, N. F., & McGuinness, D. L. 2001. Ontology development 101: A guide to creating your first ontology.

[Wine] "Wine Ontology: an Example OWL Ontology" [Online]. Available: http://www.daml.org/ontologies/76.html [Accessed on 28-Nov-2020]

[Bevon] "BEVON: Beverage Ontology" [Online]. Available: http://rdfs.co/bevon/latest/html [Accessed on 28-Nov-2020].

[Beer-Ont-1] "Beer Ontology 1.0" [Online]. Available: https://www.cs.umd.edu/projects/plus/SHOE/onts/beer1.0.html [Accessed on 28-Nov-2020].

[Beer-Ont-2] "Beer Ontology, OWL Lite" [Online]. Available: https://dbs.uni-leipzig.de/files/coma/sources/fd/beer.owl [Accessed on 28-Nov-2020].