Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Master Thesis

Sigurd Sippel

Domain-specific recommendation based on deep understanding of text

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Sigurd Sippel

Domain-specific recommendation based on deep understanding of text

Master Thesis eingereicht im Rahmen der Masterprüfung

im Studiengang Master of Science Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Kai von Luck
Zweitgutachter: Prof. Dr. Klaus-Peter Schoeneberg

Eingereicht am: 21. April 2016

Sigurd Sippel

Thema der Arbeit

Domain-specific recommendation based on deep understanding of text

Stichworte

Empfehlungssysteme, KDD, Data Mining, Tiefenverständnis, Featureextraktion, Ontologie, Basiskategorien, Validierung, Domänenexperten

Kurzzusammenfassung

Diese Masterarbeit betrachtet den Prozess zur Entwicklung domain-spezifischer Empfehlungssysteme am Beispiel der Domäne Cocktailrezepte. Auf Basis einer Ontologie wird ein Tiefenverständnis der Texte — der Rezepte — erzeugt. Die Ontologie ist modelliert mit Basiskategorien, die dazu dienen Markmale wie Zutaten aus dem Rezept zu extrahieren. Für eine Vergleichbarkeit der Rezepte werden Zutaten auf Basis von Aromen modelliert. Für den Prozess der Datenaufbereitung bis hin zur Empfehlung wird KDD als Leitlinie verwendet. Die Empfehlung anhand einer domänen-spezifischen Distanzfunktion wird berechnet. Zur Klassifikation einer Empfehlung zu einem gegebenen Favoriten wird ein k-nearest neighbor verwendet. Die Validierung erfolgt durch eine Befragung von Domännenexperten hinsichtlich der Akzeptabilität der Empfehlungen.

Sigurd Sippel

Title of the paper

Domain-specific recommendation based on deep understanding of text

Keywords

Recommender Systems, KDD, Data Mining, Deep Understanding, Feature extraction, Ontology, Basic Categories, Validation, Domain experts

Abstract

This thesis considers the process of development for a domain-specific recommender system that uses the domain of cocktail recipes for experiments. Based on ontology a deep understanding of text is created — recipes are considered. The ontology is designed by basic categories to extract features such as ingredients. Ingredients are modeled by flavors for comparability. The process of data processing along with the recommendation uses the KDD process as its guidelines. The key of the recommendation is based on domain-specific distance functions. A nearest-neighbor approach is used to classify recommendations for a given favorite. Validation is considered based on the acceptability of domain experts.

Contents

1. Introduction

In order to understand the recommendation process, a specific domain is used for experiments that are focused on deep understanding of text. Deep understanding [ASB08] leads to a rich semantic representation of data, which is necessary for content recommendation. As an example of a specific domain, the domain of cocktails is chosen because it is definite and documented by bartending manuals and books of cocktail recipes written by domain experts. The quality of recommendation is not directly measurable, but domain experts can also be interviewed to get feedback on the quality.

The cocktail recommended by a bartender in a bar has to be appropriate to the ambience and to the preferences of individual guests. The success of a bar depends on how appropriate the recommendation is. If the guest asks the bartender for a recommendation, the bartender then acts as a gatekeeper [PGK11, p. 105] for the cocktails for him, because the bartender is limited by the given circumstances. However, though he probably knows about a great variety of drinks, only a few are on his mind at a particular point in time.

Cocktail recipes are considered as special kinds of cooking recipes because they contain basically a small list of ingredients and suggestions about preparation and glassware. When cooking, people tend to use a limited collection of recipes, which they try to remember. A huge number of cooking recipes is available in the form of books or on the internet, and it is possible to select a new meal every day by using these sources [STIM09, p. 9]. Owing to such a diverse and dizzying range of choices, a system of automatic recommendation deals with such diversity [XYL10, p. 254].

A bartender's recommendation serves as a metaphor for the content-based recommendation approach. The cognitive processes of human beings and machines are different, so only the performances and not the processes have to be validated.

Recommender systems [RV97] make it possible for a user to classify an appropriate recommendation if the necessary knowledge about the items — such as quantified ingredients — are given. This thesis considers a recommender system of cocktail recipes that are appropriate to domain experts. It includes the algorithm of and modeling techniques for recipes. Since there is little research on cocktails, theoretical approaches to cooking recipes will be applied to cocktail recipes. The differences is that cooking recipes considers a longer process of preparation including more ingredients, the consideration in cocktail recipes goes more in depth such as more definite ingredients instead of breadth.

Section 2 describes the objectives of the thesis. In Section 3, the literature review focuses on the deep understanding of text. Content-based recommendation systems, the KDD process, and the modeling of an ontology with basic categories are the basis for a path-based distance measurement. Domain-related work such as modeling of ingredients and sensations such as flavors are considered. Section 4 describes the experiments started by the understanding of domain knowledge. This is followed by a feature extraction of raw text and ends with a recommendation approach including validation. The last section, 5, provides the conclusion and prospects for future work.

2. Objective

For analysis and understanding of a domain-specific recommendation system the following application is envisaged: When a guest seeks a recommendation for a drink in a bar, a selection of cocktails may be offered according to a menu card or in a more personal way by the bartender. A part of the recommendation process is based on psychological factors such as the atmosphere or guest's character but this thesis considers objective factors based on cocktail recipes.

Cocktails are written down as cocktail recipes that contain a name, ingredients including quantity, and partially short information about preferred glassware and preparation. Detailed availability of these specific parts is presented in the experiments.

> Manhattan Cocktail [1]
> 1 dash of gum syrup, very carefully;
> 1 dash of bitters (orange bitters);
> 1 dash of curacao, if required;
> 1/2 wine glass of whiskey;
> 1/2 wine glass of sweet vermouth;
> stir up well; strain into a fancy cocktail glass;

These recipes are available in cocktail books[2], blogs[3], or cocktail databases[4]. The sources present a huge volume of data, which is already available and increases with time.

There are different types of cocktails: Besides classic cocktails such as a Manhattan, which are cold and contain only liquid ingredients, there are hot cocktails and molecular recipes containing drops or foams. This thesis focuses on the classic recipes with two or more recipes that contain partially a cherry, a zest, or mint but are basically liquid. If it is liquid, the result is a mixture containing all ingredients of this recipe. This approach assumes that a recipe results from a single mixture and each necessary ingredient is already prepared.

[1] 1882 Harry Johnson, Bartenders Manual p. 162
[2] euvs-vintage-cocktail-books.cld.bz
[3] www.winebags.com/50-Top-Cocktail-Blogs-of-2015/2910.htm
[4] www.kindredcocktails.com

Cocktail recipes contain relevant information to prepare a specific cocktail. Partially a longer descriptive text is available, but the main information is written down in a short, compressed style of language.

This thesis considers a recommendation is based on content-specific features such as ingredients and their characteristics, contextualization and individualization are not considered. These features are extracted from cocktail recipes. Implicit personalization is modeled with the help of an exemplary favorite. Instead of an ingredient, a favorite recipes tells something about the characteristic. It contains quantities, which put different ingredients in relation. This information is used to recommend cocktails.

A recommendation has to be appropriate for the guest; therefore, it has to capture the interest of the guest. It has to combine what he likes — and implicitly knows — as well as something new. Something he likes or is new could be a ingredient, a combination of ingredients or a specific flavor. A practical reference is a menu card (Figure 2.1). The menu card contains a number of recipes that are displayed as a numeric id on a coordinate system. Each of the four directions contains semantic information: Refreshing, spirituous, comforting, and adventurous. The guest can get a recommendation by prioritizing this semantic information. For example, a Manhattan recipe contains a sensation of strength and smoothness, which is located in the southeast of the map.

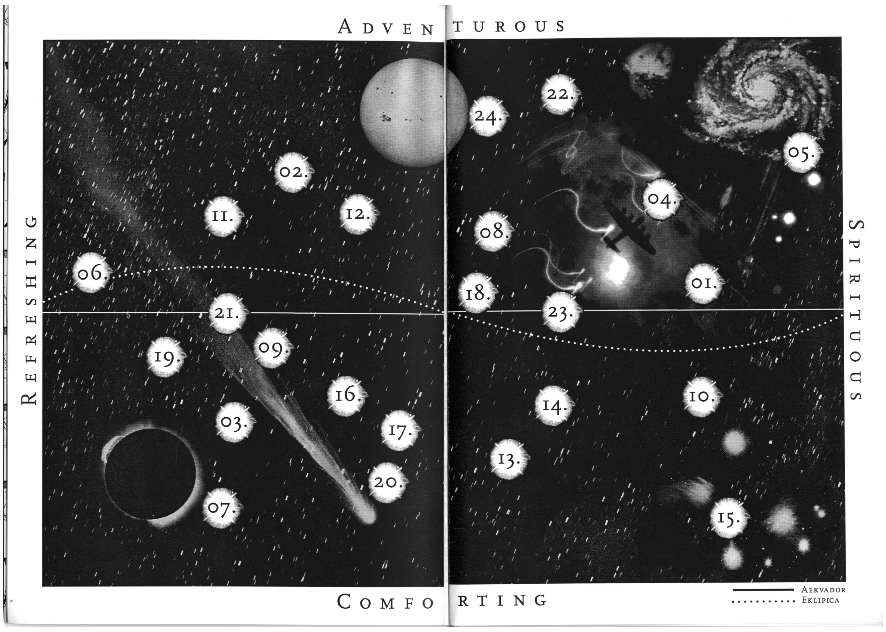


Figure 2.1.: Semantic cocktail map of the cocktail club Golem, Hamburg

Using this graphical approach of recommendation, this master's thesis considers finding a semantic modeling for appropriate cocktail recommendation. The focus is to understand a huge volume of data by using of background knowledge. In this discipline, it aims to make the machine better than the bartender. The main question is as follows: Does a knowledge-based distance function have sufficient precision for a cocktail recommendation? Recommendations for a specific domain — in this case, cocktails — can only be validated by domain experts such as bartenders. Expert knowledge is experienced-based therefore their own recommendation or their evaluation of a given recommendation is subjective. A threshold for sufficient precision is that the recommendation is acceptable to domain experts but it is not necessarily their own recommendation.

3. Literature review

The structure of the literature review is as follows: The first part of the literature review provides the main aspects of a recommender systems in section 3.1. In section 3.2, the KDD process is described, which enables comparisons between datasets and detects correlations between them. Similarity measurement for understanding of relationship between found categories is considered in 3.3. Section 3.4 introduce the idea of basic categories. Categories are considered as opportunity to model knowledge in an ontology, which is described in Section 3.5 including graph-based approaches of recommendation systems for cooking recipes. The understanding of sensations is described in section 3.6. This includes the consideration of textures, flavors, nutrition, and a cocktail recommendation approach with an emotion tracking based on colors. Section 3.7 is gives a summary of validation approaches for recommender systems. The last part of the literature review describes the extracted challenges where parts of the literature review are dealt with in the following experiments.

3.1. Recommender systems

Assuming that deep structured data is given, a recommender system[1], called RS, uses such data to assist a user in finding something he is interested in [RRSK10, p. 2]. For example, in a web shop a user gets a recommendation of some products, which are similar to the last added item in his shopping basket. From the shop-owner's point of view, an RS increases the number of items sold. Another way of motivating a user is to give him the opportunity to select an item from a huge catalog, but it is necessary to know what the user likes.

An RS is strongly connected to a user [RRSK10, p. 6]. If a recommendation is accurate, it will satisfy him more. If a user gets what he wants, he will come back. The recommender system needs to learn how to improve itself with each user visit. The service owner can optimize his service (like stocking management) if an RS tells him what users are interested in. Because of user satisfaction and loyalty, an RS is a part of human-computer interaction. Search engines

[1] Previous version is located in section 2 of https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2014-aw2/sippel/bericht.pdf

are recommender systems and are used in two ways: A user can find an interesting website with a search engine and can check how important a website is based on his interest.

At the center of data are users and items [RRSK10, p. 10]. The relationships between them enable predicting how useful an item is for a user. With estimation, a limited set of items with the best estimation can be assigned as a prediction to a user. It is a prediction, because the RS does not know whether the user will be satisfied, but it concludes that the predicted item will be the most appropriate one by a given distance measurement. In order to predict an item for a user, the item has to be comparable to other items. The concept is as follows: If a user likes item x, he will probably like an item y, which is similar to x.

Recommendation systems need user ratings such as numeric or ordinal value to predict items [DK11, p. 108]. Ratings are received in different ways, which are discussed in the evaluation of recommendation systems. In the end, the ratings describe the similarity between a user and an item. The classification whether an item is appropriate to a user or not is described by function $f(u, i)$ (Equation 3.1), which maps a user u of a set of users U and an item i of a set of items I to a classification $S = \{appropriate, unappropriate\}$.

$$f : U \times I \rightarrow S \tag{3.1}$$

For implementing this function, it is necessary to define the similarity between users and items. In the neighborhood-based recommendation [DK11], there are two different approaches: First, the content-based approach uses user profiles that are described by rated items. For example, if the user chooses his current favorite only, this favorite is the description of the user profile and a recommendation is a similar item to the favorite. Personal information is explicitly missing. The second approach is the collaborative filtering that uses similar profiles of item ratings.

Collaborative filtering is divided into neighborhood-based and model-based approaches. The neighborhood approach recommends items that are highly rated by similar user profiles. It is either a user-based approach or an item-based approach. User-based neighborhood approach uses the user's rating of an item to find other user profiles containing similar rates. These similarly classified user profiles are used to predict the estimation $f(u, i)$. Item-based neighborhood approach recommends items that are similar to the given item i and contains similar (good) ratings. In this case, an item-to-item similarity is additionally necessary.

Model-based approaches learn a predictive model by a huge training set of user data and a chosen method such as Support Vector Machines. Instead the neighborhood-based and content-based approaches are usable by small sets of data because training is not necessary. Another advantage is the simplicity, because only a similarity function is needed. The decision whether a result is appropriate or not is easy, because the responsible data can be analyzed directly.

Summary

The decision of using an content-based or a collaborative filtering approach depends on data trustworthiness. Less trustworthy ratings result in higher precision such as many ratings of low quality. For using a collaborative filtering approach, a huge volume of user data is necessary to get a recommendation, which is called the cold-start problem. If there is no user data, a content-based approach is necessary. A content-based approach needs no user data, but a precise similarity measurement depends on the quality of knowledge.

3.2. Knowledge discovery in database and data mining

Knowledge has to be extracted from the huge volumes of data before a deep similarity measurement is possible. Manual analysis of huge volumes of data is a slow, expensive, and subjective process [FPSS96, p. 28]. Knowledge discovery in database and data mining process[2] — called KDD process — aims at automatically extracting useful knowledge from huge volumes of data [FPSS96, p. 27]. For knowledge extraction out of simple text, a heavy preprocessing is needed to remove noise, which includes removing of stop words, stemming, and lemmatization [CMR07]. After this preprocessing is done, the KDD process is applicable: A company's periodic report, based on facts in relational databases, is an example of an application of KDD. Computed knowledge in this case is as follows: Sales of a product in a quarter are lower than the last five-year period. Raw data in a database contains much information [MHC06, p. 3]; the end product of the process of information discovery is the extracted knowledge [FPSSU96, p. 39].

Preconditions for the KDD process (Figure 3.1) are as follows: Understanding a domain, locating available background knowledge, and identifying a goal. In the domain of cocktails, possible background knowledge is an ontology of ingredients. For example, a goal is to identify similar recipes. The KDD process is divided into five steps [FPSS96, p. 30]: The first step involves selecting a target dataset that will be used to extract knowledge. The set of facts in the database comprises the data [FPSSU96, p. 41]. In a recipe database, these could be titles, authors' names, and ingredients of recipes. For identifying similar ingredients, information such as titles and authors' names are not relevant. Only relevant information should be in the target dataset. The next step is the preprocessing for reducing noise and outliers such

[2] Previous version is located in section 3 of https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2014-aw2/sippel/bericht.pdf

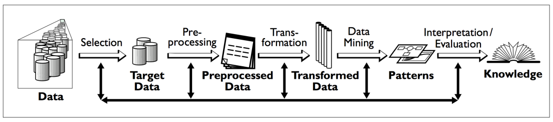


Figure 3.1.: KDD process [FPSS96, p. 29 Figure 1]

as duplicated ingredients. If a plausibility check of the remaining data is undertaken, then useless data can be weeded out. For example, a recipe without any ingredient is useless. In the third step, data is transformed into a comparable form — the target feature vector. Data can be reduced or converted. For example, the quantities can be converted into the same measurement units.

The next step is data mining. In line with the model, the composition of the target data, and the goal, there are four possible model functions: Regression, clustering, classification, and summarization.

A regression analysis is used to find a function to describe the feature vectors. For example, in a time sequence, regressions are able to predict the future trends because of the known feature vectors [FPSSU96, p. 44]. Clustering computes distance with each feature vector combination and creates a group of feature vectors — a cluster — with a smaller distance than a specific threshold [JMF99, p. 274]. Classification maps feature vectors to predefined classes [FPSSU96, p. 44]. A summarization process detects the most important parts of a document and computes the correlations to other documents [CWML13, p. 527]. For example, it reduces a document to only the most frequently used words. After a function is chosen a concrete algorithm has to be selected and run for data mining. The result is the correlation of the input feature vectors — the patterns.

The last step is the evaluation and interpretation of the mining results. Feature vectors that do not have any correlation or domain-specific semantic sense have to be removed. The technical feature vector cannot be easily read by a user. Visualization, such as in a connected graph, makes it readable. If the accuracy of the result is good enough, potentially useless features can be removed to optimize the performance. A feature is useless if accuracy without this feature is at least as good as with the feature.

KDD as a development process is not a downfall model. The steps can be repeated at every moment if it is necessary to make the knowledge extraction process more accurate.

Data mining is a statistical method of analyzing data. In statistics, a random sample is considered significant if it is collectively valid [FKPT07, p. 33]. A problem is that if someone searches long enough in a set of statistical data, he might find a significant pattern that may not have any link with reality [FPSSU96, p. 40]. Therefore a validation process in needed.

Summary

KDD is a timeless and adaptive approach for extracting knowledge from data because no algorithm is predefined and it is domain independent. It is a process with loosely coupled steps. Not every step is important, but these steps are necessary. The disadvantage is that every developer who wishes to use KDD will have to find his own focus, like distance measurement or feature extraction. Every step is a potential money sink. Nevertheless for recommendation systems, KDD is a guideline for the technical process.

3.3. Distances in context of clustering and classification

Distance measurement[3] is necessary to understand relations between data. A Levenshtein distance [PROA12, p. 706] gives the number of characters that have to be changed to transform a string into another one. The transformation entails addition, removal, and switching of characters. The distance between two strings Rye and $Whiskey$ is higher than between Rye and Gin (Equation 3.2), though in a semantic way, a rye is a special kind of whiskey, and rye and gin are different. Therefore, when characters are considered, no semantic comparability is possible.

$$\begin{aligned} levenshtein(rye, whiskey) &= 8 \\ levenshtein(rye, gin) &= 3 \end{aligned} \tag{3.2}$$

A process of clustering detects groups — called clusters — in a set of feature vectors [JMF99, p. 265]. Clustering is unsupervised learning, since it is not necessary to have a specific learning set. Depending on the quality of feature extraction, groups contain feature vectors that are more similar to each other than to outside feature vectors. Similarity is defined as a distance function such as a Euclidean distance (Figure 3.2). The extracted features are components of the feature vector. A Euclidean distance computes the difference between each component, squares it, and takes the second root of the sum.

$$d_2(\mathbf{x}_i, \mathbf{x}_j) = (\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2)^{1/2}$$

Figure 3.2.: Euclidean distance [JMF99, p. 271]

[3] Previous version is located in section 3.2 of https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2014-aw2/sippel/bericht.pdf

The validity of the result — the located clusters — has to be meaningful and comprehensible, but this is subjective [JMF99, p. 268]. An indication of this is that clusters might be too big or too small, which are seen as an annoying artifact. Another way of understanding this is through following description of a cluster: A cluster, as a set of feature vectors, has a centroid. For example, a centroid is a feature vector closest to the center of a triangle that describes the center (Figure 3.3). A centroid is also a compact representation of a cluster, which not necessarily tangible or useful.

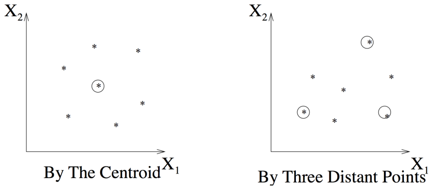


Figure 3.3.: Centroid of a cluster [JMF99, p. 282]

The initial state of a clustering algorithm is either agglomerative or divisive [JMF99, p. 274]. An agglomerative algorithm creates for each feature vector a new cluster. The clusters will be merged together. A divisive algorithm creates one cluster for all feature vectors that will be split. Both these methods need a stopping criterion, a threshold, in order to decide whether to merge or split.

For computing clusters, there are hierarchical and partitional algorithms. The hierarchical and agglomerative approach seeks the nearest pairs and uses these pairs to find the nearest pairs of pairs. The result is a nested cluster — a tree (Figure 3.4). One of the clusters is useless, but this cluster can be cut at every depth to get the end result. It has a high complexity in time and space [JMF99, p. 277].

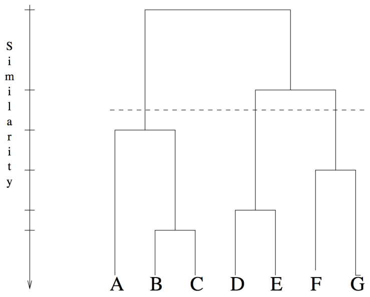


Figure 3.4.: Hierarchical clustering [JMF99, p. 276]

The partitional approach considers the feature vectors as one partition [JMF99, p. 278]. The k-means algorithm is an example. For initialization, it chooses a feature vector randomly for k clusters and considers the feature vector as the centroid, because it is the only one. Then a loop starts: The clusters are split or merged. The centroid of each cluster is recomputed. The loop will stop if the clusters are not, or only minimally, changed. The k-means algorithm has a low complexity of $\mathbf{O}(n)$, but it needs isotropic features for delivering a good result.

Up to this point, a feature vector is in only one cluster. The clusters are disjunct sets. This is called hard clustering; the alternative is fuzzy clustering [JMF99, p. 281]. The main difference is that the assignment of a feature vector to a cluster is not finished if the closest cluster is found. It needs a set of clusters that are close enough (Figure 3.5).

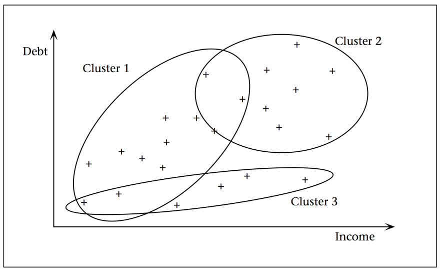


Figure 3.5.: Fuzzy clustering [FPSSU96, p. 45 Figure 5]

In order to use a clustering algorithm, the feature vectors have to be constructed. For computing the distance between two feature vectors, domain-specific knowledge is needed for selecting the features and distance function [JMF99, p. 289].

Classification is different from clustering. It maps feature vectors to labels. A feature vector is either classified as label or not. There is no state between. As it uses defined classes, classification is a kind of supervised learning [AJOP11, p. 48].

$$trainingset = \{(x_1, l_1)...(x_n, l_n)\} \tag{3.3}$$

$$neighbours \subseteq trainingset \tag{3.4}$$

$$neighbours = \{(y_1, l_1)...(y_k, l_k)| \tag{3.5}$$

$$min(\sum_{i=1}^{k} d(q, y_i), trainingset)\} \tag{3.6}$$

The k-nearest neighbor classifier[4] — called k-NN — classify a feature vector q to its nearest neighbors [AJOP11, p. 48]. The k defines how many neighbors are being considered. Assuming that the training set has a size n and contains tuples of feature vectors x and mapped labels l, the k neighbors are a subset of $trainingset$ with the following condition: The sum of all distances d between query item q and the $neighbors$ is minimal.

[4] Previous version is located in section 7.1 of https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2014-aw2/sippel/bericht.pdf

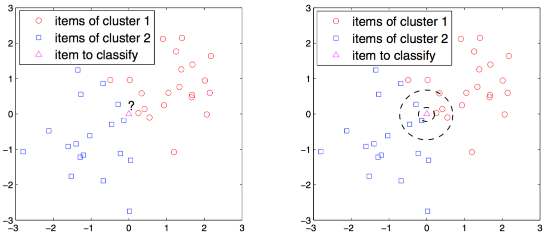


Figure 3.6.: Example of an query on k-nearest neighbor classifier [AJOP11, p. 49]

If k is too small, the classifier may yield a bad result because there might be too much noisy data in the neighborhood. If k is too high, too many different neighbors come into the neighborhood. With $k = 1$ in the example (Figure 3.6), the feature vector is classified as a square label; with $k = 5$ the result is a circle label. This algorithm is a lazy learner because it does not run a training phase before a random feature vector is classified; it just uses the neighborhood of q.

Summary

Clustering produces clusters that are not predicted by humans, but clusters have probably no semantic sense. In case of recommendations, it is useful to get an initial idea as to how distances work and which kind of data is in the dataset. For recommendation itself, a classification such as k-NN is a lightweight decision-maker that deals easily with each kind of feature vector and allows easy debugging because of the simple calculation method.

3.4. Basic categories

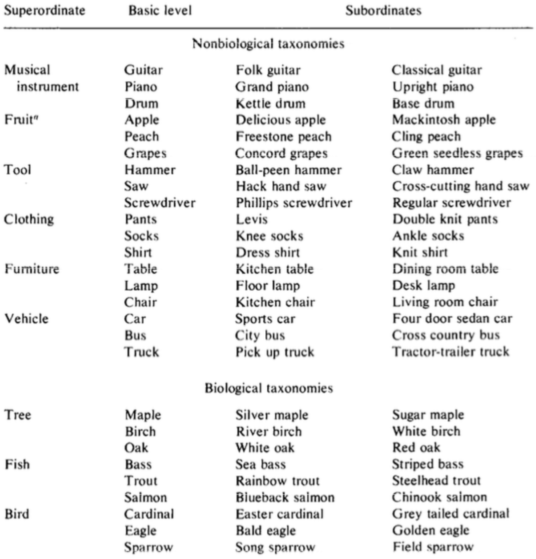
For a deep understanding of a text, a definite relationship between data is necessary[5]. One example of modeling relations is the following basic categories: From a semantic point of view, two ingredients that are not equal can share some properties. These properties have to be known to say something about the distance between two ingredients. For instance, rye and $bourbon$ are special types of $american\ whiskey$, so they share properties like origin, manufacturing, barrel-aging, and color.

All organisms classify their environments [RMG$^+$76, p. 382]. In the world, huge quantities of information reach a person. These are called stimuli, which everyone has to process. A category assigns a name to a group of things that share important properties. A category is not arbitrary, because all things in the world depend on one another, though the strength of

[5] Previous version is located in section 3 of https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2014-sem/sippel/bericht.pdf

dependency can vary significantly. A *gin* and *whiskey* are both different kinds of *spirits*, but a *gin* is not very similar to a *whiskey*. Both are unrelated to a *cocktail glass*. Of course, these can be associated with each other, but ingredients and drinking glasses belong to entirely different classes.

A taxonomy contains categories of the same class, which are organized as a root tree. The size of a category subtree depends on the level of abstraction. The category *spirits* contains *gin* and *whiskey*, but *gin* does not contain *whiskey*. The term *spirits* is an abstraction for the other ones. The level of abstraction that carries the most information is called the basic level [RMG+76, p. 383].



Superordinate	Basic level	Subordinates	
		Nonbiological taxonomies	
Musical instrument	Guitar	Folk guitar	Classical guitar
	Piano	Grand piano	Upright piano
	Drum	Kettle drum	Base drum
Fruit[a]	Apple	Delicious apple	Mackintosh apple
	Peach	Freestone peach	Cling peach
	Grapes	Concord grapes	Green seedless grapes
Tool	Hammer	Ball-peen hammer	Claw hammer
	Saw	Hack hand saw	Cross-cutting hand saw
	Screwdriver	Phillips screwdriver	Regular screwdriver
Clothing	Pants	Levis	Double knit pants
	Socks	Knee socks	Ankle socks
	Shirt	Dress shirt	Knit shirt
Furniture	Table	Kitchen table	Dining room table
	Lamp	Floor lamp	Desk lamp
	Chair	Kitchen chair	Living room chair
Vehicle	Car	Sports car	Four door sedan car
	Bus	City bus	Cross country bus
	Truck	Pick up truck	Tractor-trailer truck
		Biological taxonomies	
Tree	Maple	Silver maple	Sugar maple
	Birch	River birch	White birch
	Oak	White oak	Red oak
Fish	Bass	Sea bass	Striped bass
	Trout	Rainbow trout	Steelhead trout
	Salmon	Blueback salmon	Chinook salmon
Bird	Cardinal	Easter cardinal	Grey tailed cardinal
	Eagle	Bald eagle	Golden eagle
	Sparrow	Song sparrow	Field sparrow

Figure 3.7.: Classification with basic level object [RMG+76, p. 388 Table 1]

A category, which has many implicit properties, helps to say something about it. A large number of categories with small discriminations present a detailed perspective of the class. A basic category, such as a *car* (Figure 3.7), combines two categories; it is not too abstract, such as a *vehicle*, and not too detailed, such as a *sports car*. A basic category is a category at the basic abstraction level. It can be imagined as a picture and is probably tangible [RMG+76, p. 406]. The more abstract category is called a superordinate and the more concrete category is the subordinate [RMG+76, p. 385]. The tree depth is not limited, and so, for every subordinate, a refinement is possible [RMG+76, p. 432].

In a new domain, the categories have to be detected. People do not find correlations where there is nothing; they can only find less than what there is [RMG+76, p. 430]. One method of detecting this is to ask people what they see in a picture — and ask them which pictures they would put together under one category [RMG+76, p. 416]. The first results are the basic object and the last are the superordinates. In the domain of cocktails, a randomly chosen person could categorize the picture of a bottle of gin as a bottle of some kind of alcohol. Experts can change the results because their knowledge has many special properties [RMG+76, p. 430]. But experts are most focused [RMG+76, p. 432], such as on one basic category of ingredient that affects the richness of the details in the result. In this case, the categories are not balanced; the relationships differ in accordance with their relevance.

In a study of airplane classification (Figure 3.8) involving people with and without expert knowledge, the recognition in accordance with superordinates was very similar, but on a basic level, experts' recognition was greater in line with superordinates. Therefore, experts are needed, but the balance of the results has to be kept in mind.

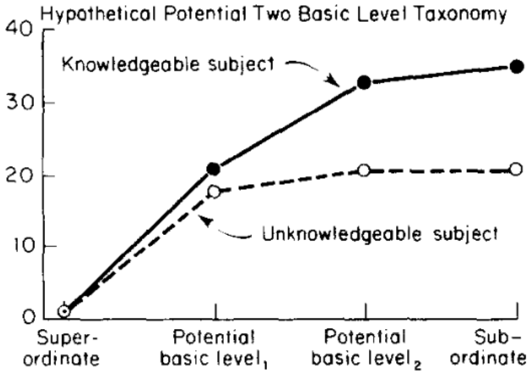


Figure 3.8.: Airplane classification [RMG+76, p. 431 Fig 4]

Now, *whiskey* is highly classified by country of origin, such as *Ireland* or the *US*, and then it is classified into ingredient-based categories, such as *bourbon* and *rye* for the *US*. *Gin* does not have such a detailed official classification for subordinates. The relations are not completely equal to any relations with another parent category. Experts are aware of that, but they cannot change this situation. Coloring the basic categories and weighting their subordinates are methods to balance knowledge, but the reasons are not objective.

Independent of experts, some basic objects are obviously a kind of superordinate, while some basic objects are not. From the perspective of gin experts, *London dry gin* is the most widely propagated kind of gin. Most gin products are subordinates of this. There are also special gins, such as the ones that are barrel-aged in peaty whiskey casks. Both are basic objects, but *London dry gin* has the most common properties in the category. It is a prototype of this category [RMG+76, p. 433]. It represents the center of the category.

Summary

The concept of basic categories is useful to design ontologies and fill them with knowledge. The granularity has to be adjusted to get meaningful categories, because these are necessary to get a meaningful similarity between categories (section 3.5) that are reasonable by domain expert.

3.5. Extraction and distances with ontologies

Hierarchies of categories are simple ontologies[6]. A popular ontology is Resource Description Framework (RDF) [w3.15], which is based on XML or Turtle [CLS01, S. 7]. It is a domain-independent description language that connects content. It also separates content into several classes.

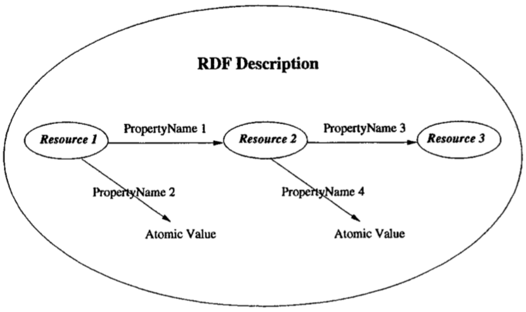


Figure 3.9.: RDF model

The RDF model contains a set of triples $(resource, property,\ atomic\ values)$. Instead of atomic values such as labels or titles, there could also be other triples. This nested definition is used to model trees [CLS01, S. 10]. Every property can have a URI for ensuring a unique address. The property describes the edge that connects the left with the right one. There are predefined properties. Every resource has a type that is referred to as a class such as $ingredient$. Self-defined properties are supported.

In order to calculate the contextual distance between two recipes, components of the recipe — such as ingredients — have to be classified under categories. This is the preprocessing in KDD which converts from an unknown entity to a known one. In the absence of this classification, it is only possible to state whether the name of an ingredient is the same as that of another. Categories identify similar properties.

[6] Previous version is located in section 4 of https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2014-sem/sippel/bericht.pdf

Graph-based modeling such as with RDF is used for item similarity [DK11, p. 136]. For example, a set of users is given (Figure 3.10). Each user u has a directed edge to each item i. Each edge contains the rating of an item for this user.

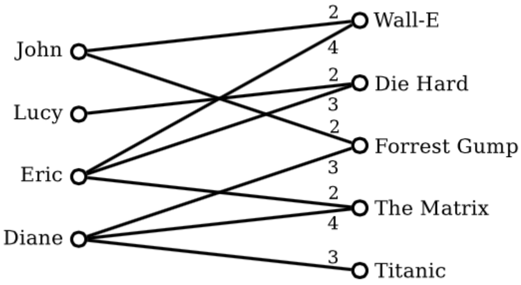


Figure 3.10.: Example of a bipartite graph based on user rating for items [DK11, p. 135 Fig. 4.4]

This is used to get not only direct ratings between u and i, a transitive similarity is possible. There are two approaches: The first one is to propagate information along edges with the highest weight, called propagation. The second one is to reduce the influence of nodes that are further away; it is called attenuation.

Path-based similarity [DK11, p. 136] is the similarity between nodes in graph, which is described by a function that maps two nodes to one number according to the needed steps.

For user similarity, there are two different approaches: The first is to find users, with a similar set of rated items, called horting. The second is to find a user which rates similarly; this is called predictability. The set of common ratings I_{uv} (horting) of users u and v is used to calculate the predictability of I_{uv} (Equation 3.7). The differences between rating r of both users are summed up if a threshold γ is not exceeded. A linear transformation l is used to scale the ratings of user v to the ratings of user u. The ratio of the sum to the length of horting set is the predictability of user v for user u.

$$\frac{1}{|I_{uv}|}\sum_{i\in I_{uv}} |r_{ui} - l(r_{vi})| \leq \gamma \tag{3.7}$$

For an unrated item i of user u, the path P (Equation 3.8) is defined as the shortest path measured by predictability. This path contains only users who have rated i yet. The predicted rating r_{ui} is the composition of all linear transformed ratings of the path. The composition is defined as average of all ratings.

$$P = \{u, v_1, v_2, ...v_m\} \tag{3.8}$$

The path-based similarity can be applied to an ontology. The ontology has to know the ingredients and synonyms to find semantic similarities, but the world is always greater than a ontology. This is an important risk for the precision of the distance. There are many ontologies containing categories which are not always useful for a specific domain. Wineglass could be a old measurement unit or a drinking glass, context is necessary for precise decision. WordNet (Figure 3.11) is one that also includes cocktail ingredients. WordNet contains words with types of words, such as nouns (n), synonyms, and hyponyms (subcategories) [Mil95, p. 40]. But there are missing categories, too, such as the unit wine glass. It contains only the drinking glass, which is probably the wrong taxonomy.

Specialized databases, such as e-commerce databases of ingredient shops, contain more products and these are probably categorized. However, these databases have to be available as well as integrated into the ontology. Manual optimization is necessary to extract these features precisely.

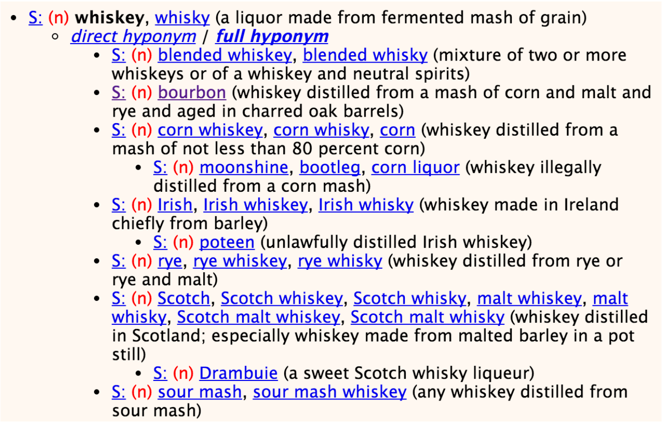

- S: (n) **whiskey**, whisky (a liquor made from fermented mash of grain)
  - *direct hyponym* / ***full hyponym***
    - S: (n) blended whiskey, blended whisky (mixture of two or more whiskeys or of a whiskey and neutral spirits)
    - S: (n) bourbon (whiskey distilled from a mash of corn and malt and rye and aged in charred oak barrels)
    - S: (n) corn whiskey, corn whisky, corn (whiskey distilled from a mash of not less than 80 percent corn)
      - S: (n) moonshine, bootleg, corn liquor (whiskey illegally distilled from a corn mash)
    - S: (n) Irish, Irish whiskey, Irish whisky (whiskey made in Ireland chiefly from barley)
      - S: (n) poteen (unlawfully distilled Irish whiskey)
    - S: (n) rye, rye whiskey, rye whisky (whiskey distilled from rye or rye and malt)
    - S: (n) Scotch, Scotch whiskey, Scotch whisky, malt whiskey, malt whisky, Scotch malt whiskey, Scotch malt whisky (whiskey distilled in Scotland; especially whiskey made from malted barley in a pot still)
      - S: (n) Drambuie (a sweet Scotch whisky liqueur)
    - S: (n) sour mash, sour mash whiskey (any whiskey distilled from sour mash)

Figure 3.11.: Example of a WordNet ontology [pri15]

WordNet is one huge ontology; others include DBPedia[7] and OpenCyc[8], but a lot of actual information is missing [KRSW09]. Wikipedia contains a category system[9] and infoboxes[10]. These are used by YAGO [KRSW09] as sources for an actual knowledge base that enriches this information by WordNet.

[7] dbpedia.org

[8] opencyc.org

[9] en.wikipedia.org/wiki/Category:Systems

[10] en.wikipedia.org/wiki/Help:Infobox

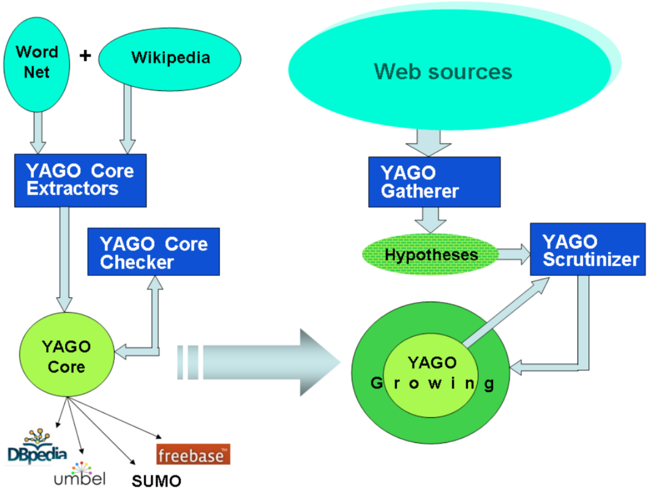


Figure 3.12.: YAGO architecture [KRSW09, p. 43 Fig. 2]

The architecture of YAGO (Figure 3.12) is oriented to extract Wikipedia knowledge with WordNet by the $Core\ Extractor$. The info boxes represent a semi-structured key value store such as birthday of a politician, which are used to extend a model such a specific politician with the new relation $birthday$ to the dependent value. The Wikipedia categories are used to extract the isA relation of the politician such as Nicolas Sarkozy (Equation 3.9).

$$Nicolas\ Sarkozy\ isa\ PresidentOfFrance \tag{3.9}$$

For each extraction, a temporal validation is needed, because this information is time dependent. A president acts for a specific period. Therefore, each relation contains a period. Missing information is filled with positive and negative infinity.

The extracted data has to be passed by the consistency checker because the data of Wikipedia is redundant and inconsistent. For this knowledge (Equation 3.9) is not enough to conclude that Nicolas Sarkozy is a president or a French person. For conclusion of this transitive relation, the ontology of WordNet is used. The aim is to map all entities to WordNet classes. WordNet classes contain, according to the basic categories, super- and subclasses, called hyper- and hyponyms. If this mapping is not possible, the extracted entities are discarded. Additionally, there is a type checker. A relation such as $isCEO$ has to be connected to a company. If the connected entity is a country (Equation 3.10), the type is wrong and the entity is discarded, too.

$$Nicolas\ Sarkozy\ isCEO\ France \tag{3.10}$$

The output is an RDF model maintained by the $core$, which is able to export data to ontologies such as DPPedia. This knowledge is used to parse unstructured text found on crawled websites by the $Gatherer$. This text is analyzed by the dependency parser $LEILA$, which is able to connect words of sequenced sentences. The $Scrutinizer$ uses the extracted content, called $Hypothesis$, to make a consistency check on the knowledge base. Only if the extracted content is not in contradiction to prior knowledge, it is added to the $Growing$ component. YAGO is a learning ontology-based approach, which comprises consistency checks and conclusion techniques.

Summary

An ontology and a distance measurement based on the path between nodes are the primary aspects to design a content-based recommendation system. WordNet and YAGO are large and tend to be generic. This includes lack of data in detail; not each relation makes sense in domain-specific content. Nevertheless, their validation approaches to prevent finding senseless relation are important to extract knowledge in domain specific area because these ontologies needs also adding of new data.

3.5.1. Graph-based recommendation of cooking recipes

Identifying a recipe by name is difficult because the name is either based on anecdotes, appearance, or the main ingredient [WLL$^+$08, p. 979]. Translation complicates this situation so that a recipe name is no indicator for the content of recipe. A deeper understanding of the structure of a recipe is needed to get a similarity measurement for searching and recommendation.

This is a graph-based recommendation approach to cooking recipes, which is focused on a cooking graph of ingredients and actions [WLL$^+$08, p. 979]. The cooking graph (Figure 3.13) contains colored nodes — ingredients and actions. The edges are also colored: The edges of action flow describe the ordering of the actions and the edges of ingredient flow describe the ordering of ingredient; they together show how the recipe works. Actions additionally have constraints that have to be satisfied to do this action. A constraint could be the temperature of an ingredient. Time is another constraint of an action, which indicates the start time of an action. The ingoing edges of the directed graph are the predecessors, which have to be done, while the outgoing edges are the successors, which come next.

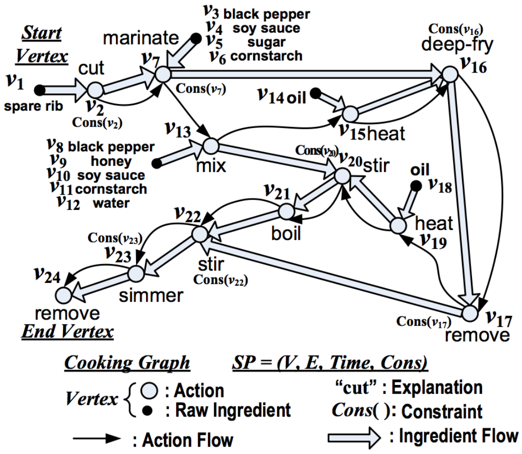
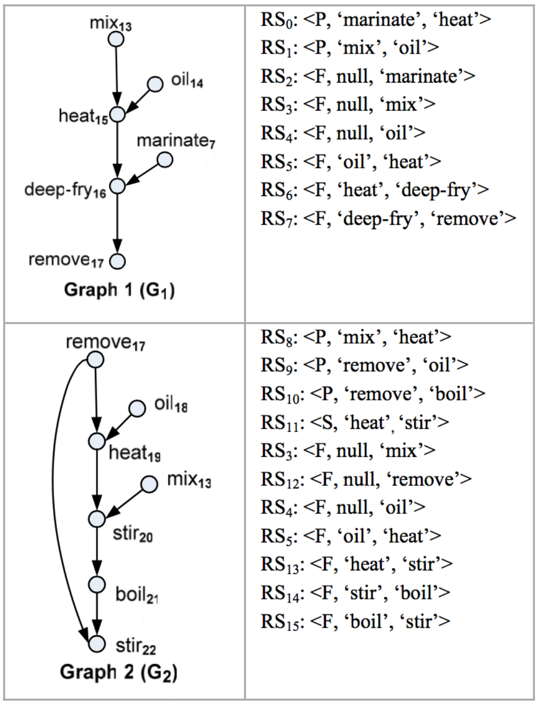


Figure 3.13.: Example of a cooking flow graph [WLL+08, p. 981 Figure 1]

Additionally a cooking recipe is described by cooking style, region, and images of the dish. Owing to the expensive process, the recipes are prefiltered to remove the most dissimilar recipes to make acceptable responsive times of the recommendation system possible.

The cooking graph is transformed into a simplified structure. The new structure always contains a pair of two nodes v_1 (source) and v_2 (target) and three types, namely a predecessor adjacent node pair, a successor adjacent node pair, and a forward edge pair, which represents the relation from v_1 to the node with the nearest timestamp v_2. If there is no action before, v_1 is not bounded. For example (Figure 3.14) $marinate$ and $heat$ are predecessors (RS_0) of $deep - fry$ and the forward of $heat$ is $deep - fry$ (RS_6).



RS_0: <P, 'marinate', 'heat'>
RS_1: <P, 'mix', 'oil'>
RS_2: <F, null, 'marinate'>
RS_3: <F, null, 'mix'>
RS_4: <F, null, 'oil'>
RS_5: <F, 'oil', 'heat'>
RS_6: <F, 'heat', 'deep-fry'>
RS_7: <F, 'deep-fry', 'remove'>

RS_8: <P, 'mix', 'heat'>
RS_9: <P, 'remove', 'oil'>
RS_{10}: <P, 'remove', 'boil'>
RS_{11}: <S, 'heat', 'stir'>
RS_3: <F, null, 'mix'>
RS_{12}: <F, null, 'remove'>
RS_4: <F, null, 'oil'>
RS_5: <F, 'oil', 'heat'>
RS_{13}: <F, 'heat', 'stir'>
RS_{14}: <F, 'stir', 'boil'>
RS_{15}: <F, 'boil', 'stir'>

Figure 3.14.: Example of recipe graphs

A recipe graph G is described by a set of the data structure ReciSet (RS), which is called $RSS(G)$. If two recipe graphs G_1 and G_2 are given, the number of occurrences of RS_i in both RSS are used to calculate the distance.

Summary

Models of this approach cover a broad semantic space of cooking recipes; this approach is focused on the structure of recipes. The cooking graph is too sophisticated for a cocktail model, because the actions are limited in cocktail recipes, but it shows how large complexity of deep understanding in a big recipe is split up into a small comparable data structure.

3.5.2. Graph-based menu planning

Based on recipe recommendations, the following step is to recommend a complete meal [KLSL12]. This is called meal planning, which is usable for daily dinners or holiday events to obtain a meal including salads, appetizers, main dishes, and desserts. In this approach, a set of ingredient favorites are given by the user. The items are a set of recipes and meals, which are defined as sets of recipes. These are extracted from huge databases such as food.com. The recipes are described by a set of tags such as main ingredients, region, preparation, diet, and nutrition. The recipe similarity is described in [WLL$^+$08].

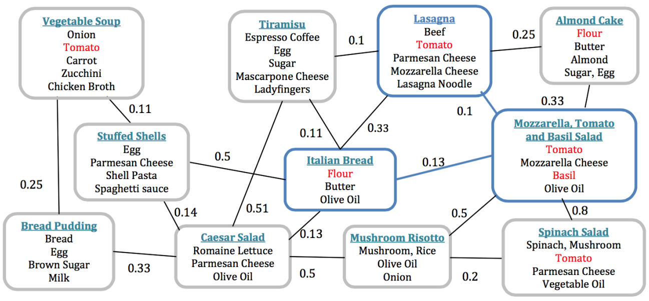


Figure 3.15.: Example of a graph-based meal plan including similarities [KLSL12, p. 3 Fig. 2]

In this approach, the recipes are modeled in a *recipe graph* as nodes (Figure 3.15). The weight of the edges represents the similarity between two recipes. The distance between two recipes is defined by the occurrence of these two recipes in the set of available menus. The lower weight indicates a higher similarity. The similarity between menus is defined as the sum of weights of the connected recipes. Each recipe which contains one of the favorite ingredients is considered for recommendation.

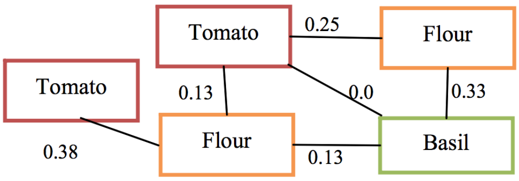


Figure 3.16.: Example of a group graph [KLSL12, p. 4 Fig. 4]

The recommendation is calculated in three steps: At first, a group graph is generated by using the favorites (Figure 3.16). Each favorite ingredient, which is contained in a recipe and the recipe is connected to a recipe containing a different favorite, is mapped to a node in the group graph; the edges and weightings are adopted. In the group graph, ingredient nodes can be duplicated. These recipes that contain both one are connected to the common ingredient node with weighting of 0. In the second, the query relevance graph is generated by replacing all ingredient nodes of group graph with the original recipe nodes (Figure 3.17). The new edges are visualized by dotted lines. The third step calculates the minimum spanning tree by lower costs of weightings. It starts at the ingredient node of the highest degree. In this case, it starts from the node of tomato. The results are a set of recipes that present the recommended menu.

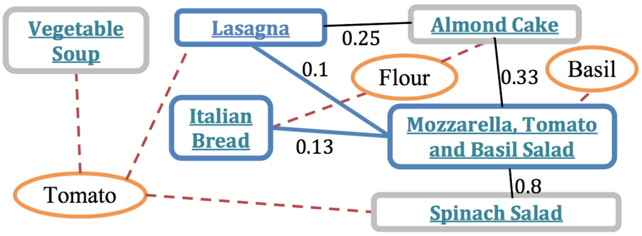


Figure 3.17.: Query relevance graph [KLSL12, p. 4 Fig. 5]

In order to get a first evaluation of this approach, the menu set is divided into training and a testing set. At first, an entropy metric is designed (Figure 3.18). For the tags in menu T_p, a probability p of a tag is used which shows how often the tag occurs in recipe set. A higher probability results in a lower entropy of one tag. The entropy function TE summed up the entropy of each menu. Low entropy shows a high coherence of the menu.

$$TE(T_p) = -\sum_{i=1}^{n} Prob(t_i) \log_r Prob(t_i)$$

Figure 3.18.: Tag entropy of a menu [KLSL12, p. 5]

The second evaluation metric is the tag co-occurrence density. At first, the co-occurrence of two different tags is calculated (Figure 3.19). Each menu of the training set P_{train}, is paired with each other recipe of this set, the co-occurrences of tags in a recipe pair are summed up and normalized by the size of recipe pairs N_p. The sum of co-occurrences per menu is normalized by the size of menus.

$$C(t_i, t_j) = \frac{1}{|P_{train}|} \times \sum_{p \in P_{train}} \left(\left(\sum_{\substack{s_1, s_2 \in p \\ s_1 \neq s_2}} \begin{cases} 1, \text{if } t_i \in s_1, t_j \in s_2 \\ 0, \text{otherwise} \end{cases} \right) \Big/ N_p \right)$$

Figure 3.19.: Tag co-occurence of two different tags [KLSL12, p. 5]

The tag pairs of a menu are computed and their co-occurrence are summed up and normalized by the size of tag pairs N_t, which is called the co-occurrence density of one menu. The results are that menus with higher recipe size get higher entropy and slightly lower co-occurrence. If the number of favorite ingredients gets higher, the entropy gets lower and the co-occurrence gets slightly lower too.

$$TCD(p) = \frac{1}{N_t} \times \sum_{\substack{s_1, s_2 \in p, s_1 \neq s_2 \\ t_i \in s_1, t_j \in s_2}} C(t_i, t_j)$$

Figure 3.20.: Co-occurrence density of a menu [KLSL12, p. 5]

Summary

This approach uses ingredient favorites to calculate a recommendation, which is an alternative approach to collaborative filtering. The distance function considers the structure of the recipe. Designing metrics such as entropy and co-occurrences for outputs of similarities is not an evaluation of a recommendation system, but is a method during the development process to get a feeling how the similarities work.

3.6. Modeling of sensations

Human sensation are considered to understand how things such as smell or taste influence the consumer — the person who needs the recommendation. The acceptability factors of sensation of food include the following: Appearance, flavor, and texture [Bou02]. Appearance is described by color, shape, and size because it is the result of the optical senses. Flavors comprise the taste of tongue and the odor, which is recognized by the nose, is the chemical stimuli. In contrast to flavor, texture is described by physical stimuli. In contact with the body, the structure of food such as crispiness or softness is recognized. Appearance, flavor, and texture are directly recognized by human senses and affect the enjoyment of consuming. Therefore, these are called the sensory acceptability factors. Nutrition, cost, and packaging also influence the senses, but in a indirect way.

The ratio of liquid controls the importance of texture [Bou02, p. 2]. The texture of beef, such as tough of dry, decides how much a kilo costs. The importance of texture is dependent on food itself; the texture of beef is more important than the texture of tendential liquids such as a soup, coffee, and beer. In terms of health, the nutrition value is in focus [Bou02, p. 21] but for liquids besides the appearance the flavors are most important for sensation.

3.6.1. Dimensions of odors

In terms of flavors, liquids are very diverse. In the following study [ZS09], a modeling of a feature vector for odors is considered with respect to personal perception. For description of a substance, there are two approaches: The first is to describe semantically by a list of similar words and the second one is to map a numeric value of an odor to reference materials. The result represents a database of similarities, which is called odorant object space. The semantic description shows which substances have the same description, while the numeric description shows which substances have the same rating according to the reference material. Both are used frequently, but a standard of odor description is not available. An approved low-dimensional modeling can be used to identify consumer preferences. The challenge is to find independent descriptions of odors. In a previous study [ZS06], substances of the semantically described SAFC database[11] are classified into about 20 classes such as fruity, floral, nutty, or balsamic, which refer to a set of description sharing this classification. These classes have to be objective in order to be independent of psychological reactions such as exciting. Properties such as pleasantness or unpleasantness show correlations between descriptions. Two pleasant descriptions or two unpleasant descriptions are appear oftener together than a pleasant and a unpleasant one, but pleasantness is not dimension of odor.

For example, $Anisyl$ is described by nine descriptions (Equation 3.11) that are classified into classes (Equation 3.12). The classes are considered the underlying dimensions in the odor description space.

$$\begin{aligned} Anisyl\ acetate \rightarrow almond; cherry; coumarin; creamy; \\ lilac; fruity; plum; sweet; vanilla \end{aligned} \tag{3.11}$$

$$\begin{aligned} almond \rightarrow nutty \\ cherry, plum, fruity \rightarrow fruity \\ coumarin \rightarrow not\ classified \\ creamy \rightarrow butter \\ lilac \rightarrow floral \\ sweet, vanilla \rightarrow balsamic \end{aligned} \tag{3.12}$$

In the following study [ZS09], two databases of odor profiles are considered. The first is the Boelens-Haring database with 309 compounds that have numeric similarity for 30 reference materials. The second is the Thibouds database with 119 perfumes that are described by three or four main odor descriptors.

These two raw databases are considered to find relations to two popular models of fragrances: The first is Jelinek's odor effects diagram, which contains four main categories — acid, sweet,

[11] www.sigmaaldrich.com

bitter, and animalic. These categories are emotional descriptions such as stimulating for bitterness.

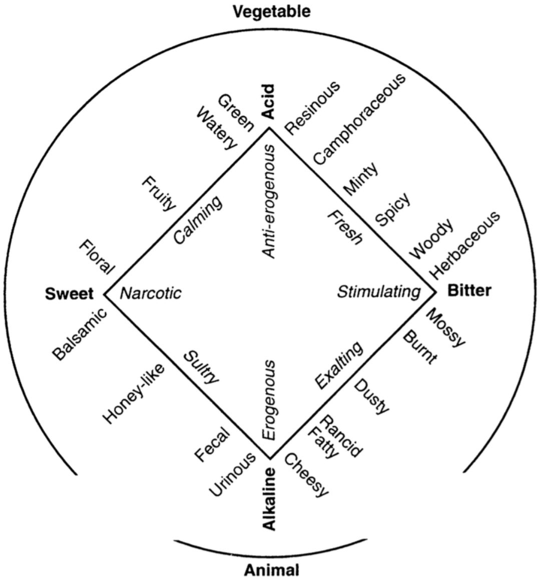


Figure 3.21.: Jelinek's odor effects diagram, originally created by Paul Jelink (1951) and updated by his son Stephan Jelinek [Jel97, p. 91 Fig 11.2]

The second is Edward's fragrance wheel (Figure 3.22). This model describes four main categories: floral, oriental, fresh, and woody. Aromatic is displayed in the center because it is intended to influence all categories. The categories are substantiated by subcategories. In the present study [ZS09], the version from 2008 is used. Since this is unavailable, the version of 1983[12] is presented here. The subcategories fruity and woody are added to the 2008 version.

[12] en.wikipedia.org/wiki/Fragrance_wheel

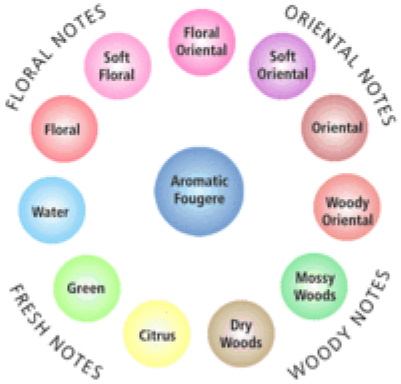


Figure 3.22.: Fragrance Wheel 1983

The first step of investigating the Boelens-Haring database is the calculation of correlations between descriptions. If the correlations are too high, the descriptions are identified as the same. The correlation coefficient (Equation 3.13)[13] is used to find descriptions pairs of n description sets that are often stored together in the databases.

$$r = \frac{\sum_{i=0}^{n}(x \cdot y)}{\sqrt{(\sum_{i=0}^{n} x^2) \cdot (\sum_{i=0}^{n} y^2)}} \tag{3.13}$$

The result is that the highest correlation is between sweet and aromatic. This is because the odors descriptions of the reference material of $vanilin$ identified by different authors sweet and aromatic; therefore, the association is heavily dependent on reference material.

The main step is to use a principal component analysis (PCA) to find a small, possible feature vector to describe the sensation of odors. Because of the dependent reference material, a threshold is defined for each descriptor. This is the average value of the descriptor in the database. Two principal components (PC) are found to describe the odor character (Figure 3.23). The values — odor descriptions — are displayed by the distance to the principal components which is called loading. The result is the loading plot of PC1 and PC2.

They use further a classification of odors [Glö91] into more feminine such as $floral$ or masculine such as $earthy$. A line of separation is found in the loading plot (displayed as dotted line), which is close to the axis of PC1. Therefore, the interpretation is that the positive values of PC2 more likely describe feminine and the negative values more likely describe masculine values. These results are according to the psychologically based model of Jelinek. For comparability, the black triangles in Figure 3.23 are values of B-H database and the open

[13] stattrek.com/statistics/correlation.aspx

circles are elements of Jelinek's model (this model is rotated clockwise). Descriptions have to be considered by their semantic value; the words can be different. For example, $burnt$ and $smokey$ are considered semantically similar. There are also exceptions such as balsamic, which is not very close to another. The result is a high accordance between Boelens-Haring database and Jelinek's model.

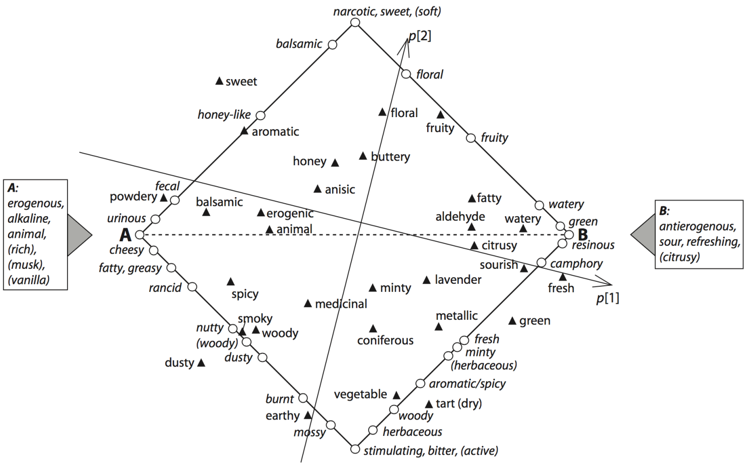


Figure 3.23.: Loading plot of PC1 and PC2 for Boelens-Haring database [ZS09, p. 236 Fig. 5]

The interpretation of PC1 is done in four ways — erogenous versus anti-erogenous (Jelinek), heavy versus light, warm versus cool (Thiboud), and powdery versus watery. Basically, there is a missing definition of what these vocabularies semantically mean. In this case, this study underlines the need of future work to understand the differences.

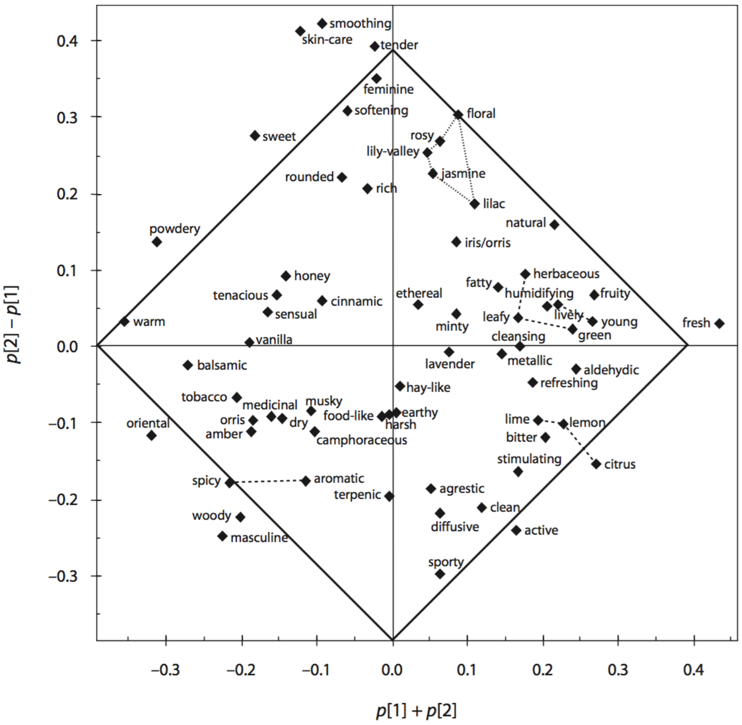


Figure 3.24.: Loading plot of PC1 and PC2 for Thiboud's database [ZS09, p. 241 Fig. 6]

The PCA was also used for the Thibouds database that results in two important principal components. These are presented in a loading plot (Figure 3.24), which is rotated clockwise for an easier comparison with the previous loading plot. The interpretations of the previous loading plot are matched in this loading plot. This is an important confirmation because it is an independent PCA that gives similar results.

Edwards' fragrance wheel has also similarities, such as floral notes are in opposite to woody notes and fresh notes are in opposite to oriental (warm) notes. The main difference is the centered $aromatic/fougere$ dimension. This dimension refers to the odorant Fougere Royal, which is described as a fresh, lavender, and mossy note. Therefore, [ZS09] suggests moving this to the location between citrus and dry woods. Edwards adopted this model based on this suggestion in an updated version of 2013 (Figure 3.25).

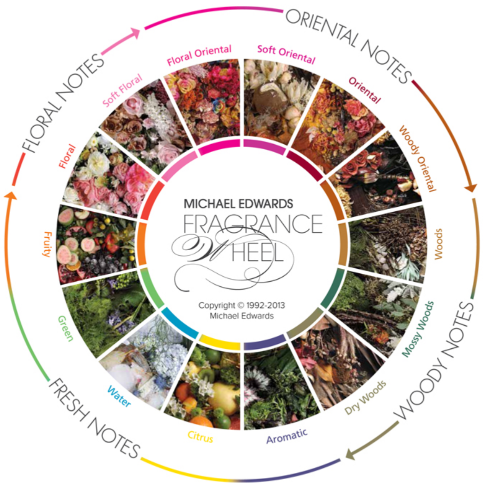


Figure 3.25.: Fragrance Wheel 2013 [Edw16]

Summary

The models are dependent on subjective point of view and the databases contain not uniform distribution of odors. Fruity descriptors are used oftener than metallic. The numeric analysis has to differ in some points from the odor models, but there is a proven correlation that is described by a vector of only two dimensions. The interpretation is less important than the composition because this is necessary to calculate each dimension.

Odors are not the same as liquids such as cocktails. A citrus odor could be acceptable for the user, but pure lemon juice probably not. Sugar or water is needed to make the sour lemon juice drinkable. The psychological perspective is an interesting part that matches several models. The transferability to cocktails is an opportunity for future work. The model for the taste of cocktails needs additional features, but this kind of numeric modeling is an appropriate basis of sensation modeling.

3.6.2. Nutrition-based modeling of cooking recipes

Another modeling approach is on the basis of nutritional[14] balance [KF10, p. 56:1]. The goal is to generate healthy meal plans. The user can get a completely auto-generated meal plan and can choose favorites, including self-monitoring of balance changes. It is based on the Japanese nutritional pyramid (Figure 3.26). The pyramid is divided into six food groups: Water, grains, vegetables, fish/meat, milk, and fruits.

[14] Previous version is located in section 5 of https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2014-aw2/sippel/bericht.pdf

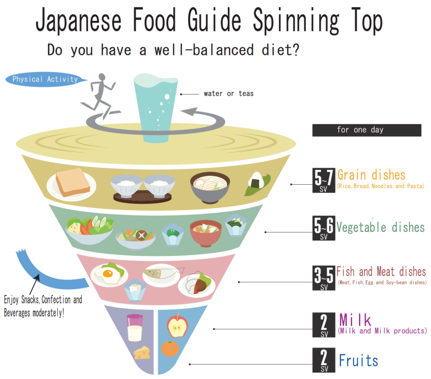
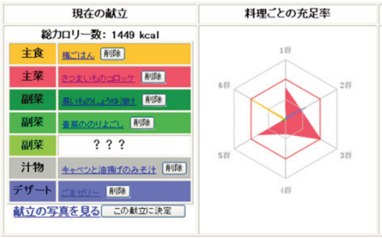

Japanese Food Guide Spinning Top
Do you have a well-balanced diet?
Physical Activity
water or teas
for one day
5~7 SV Grain dishes
(Rice, Bread, Noodles, and Pasta)
5~6 SV Vegetable dishes
3~5 SV Fish and Meat dishes
(Meat, Fish, Egg and Soy-bean dishes)
2 SV Milk
(Milk and Milk products)
2 SV Fruits
Enjoy Snacks, Confection and Beverages moderately!

Figure 3.26.: Japanese nutritional pyramid [mhl14]

With the help of a domestic science handbook, a dictionary is created that contains foods classified into food groups such as *meat* → *pork* [KF10, p. 56:4]. Since all the ingredients of a cooking recipe are classified and the quantities are available, it is possible to compute the ratio of every food group referred to in a recipe. All the ratios of ingredients of a particular food group are computed and summarized. All six food group ratios together form a recipe balance, which is visualized as a red rhombus (Figure 3.27).


現在の献立

総カロリー数: 1449 kcal

主食	梅ごはん
主菜	さつまいものコロッケ
副菜	長いものしょうゆ漬け
副菜	春菊ののりよごし
副菜	？？？
汁物	キャベツと油揚げのみそ汁
デザート	ごまゼリー

献立の写真を見る

料理ごとの充足率

Figure 3.27.: Example of meal balance [KF10, p. 56:4 Figure 4]

The meal planning uses the balance to find meals that together represent an optimal intake of food per day. The intake per day is specific to age, gender, and food class. It is a part of the Japanese food standard.

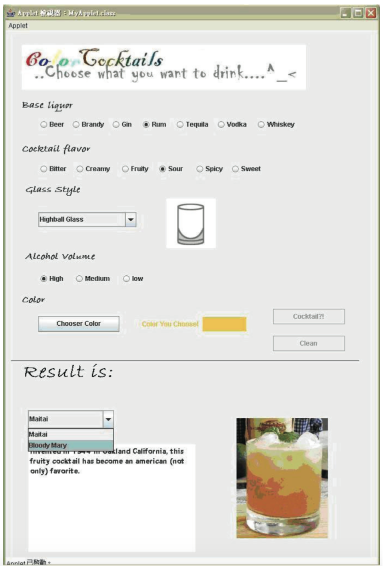
This is an approach toward optimizing diets from the point of view of healthfulness. The question that arises is this: How can the balance of such a recipe be characterized to make it comparable to another one? It is useless to know that an ingredient of a huge database does not exist in a specific recipe. However, the ingredients are classified into a small number of classes and only the classes are considered. It is important to remember that one group does not exist in a recipe. This can be computed, because all ratios are available, and the sum of all ratios is 1. It is important that this classes cover as many areas as possible.

Summary

A cocktail recipe is more restricted as there are no ingredients like fish and meat. Milk products are rare in classical cocktail recipes. Acid from fruits, water (like soda or melting ice), sweets (like liqueur, syrup), and alcohol (like spirits) are groups of ingredients that are frequently used. For a cocktail recipe, it is possible to compute the ratios in an ingredient group. All the ratios together form the balance. Focus of this approach is healthiness, nevertheless their technique of modeling and visualization is applicable for recommendation of cocktail recipe.

3.6.3. Flavor and emotion for cocktail recommendation

The approach $ColorCocktail$ [CHHH06] uses main ingredients, flavors, and colors for emotion tracking for a cocktail recommendation system. The cocktail is defined as an iced drink of wine or spirits, which is mixed with flavoring ingredients such as a liqueur. Additionally, it contains fruit, sauce, honey, milk or cream, and spice. Besides the flavors, the emotion of the person is included in the recommendation, which is expressed as a color. The shape of the glass is used because it presents the color of the cocktail. The alcohol volume is also added because it also affects the perception for the user. Based on this, a user interface is designed (Figure 3.28) that enables the user to specify what one wants to drink based on the definition of cocktail and emotion.



Applet 檢視器 : MyApplet.class

Color Cocktails
..Choose what you want to drink....^_<

Base liquor
○ Beer ○ Brandy ○ Gin ● Rum ○ Tequila ○ Vodka ○ Whiskey

Cocktail flavor
○ Bitter ○ Creamy ○ Fruity ● Sour ○ Spicy ○ Sweet

Glass Style
Highball Glass

Alcohol Volume
● High ○ Medium ○ low

Color
Color You Choose!

Result is:
Maitai
Maitai
Bloody Mary
Invented in 1944 in Oakland California, this fruity cocktail has become an american (not only) favorite.

Applet 已啓動。

Figure 3.28.: User interface of cocktail recommendation system [CHHH06, Fig. 4]

The user has to choose the feature vector directly in a qualitative way. The cocktail knowledge is stored on OWL ontology for computing the reasoning based on this feature vector. Therefore, understanding the raw text is not necessary. The ontology contains ingredients including their features of flavors and alcohol volume. The recipes are also stored in ontology.

Summary

This approach is in the early phase of development; therefore, the evaluation is not finished. But the most important part in this approach is to understand that a cocktail recommendation has to be passed to the sensation of the user. It includes a structural point of view because ingredients and glassware are used, as well as sensations such as flavors and colors.

3.7. Validation by domain experts

The quality of a recommendation is not directly measurable like temperature. In terms of clusters, the distribution of clusters and cluster sizes [SZ15, p. 1251] helps one obtain a feel of the diversity of collected data, but a recommender system has to be related to the user. Experiments without users cannot validate a recommender system[15]. Experiments require feedback mechanisms to find a ground truth, which can be used to obtain precise measurements.

Such feedback is very important for testing the validity. Clicking behavior [ANH13, p. 168] on the list of recommendation results will indicate which recommendation is being watched. User ratings [LWL14, p. 101] show how satisfied a user is with an item. In both solutions, a relationship with the initial favorite example is missing. Results serving to attract attention in an emotional way, or when the user is hungry for knowledge, have a low validity in such feedback. Recordings of the uses of the recommendation, such as video recommendations [BMCMB$^+$10], have greater validity. If the user watches a long, full-length video, then that is an important piece of information for the user profile. However, there is no relationship with a favorite example.

In a cooking recipe recommendation [HLE12] one way of getting to know what people like is to ask them[16]: Users rated recipes between one and five stars. A rating shows what users like. Following this, the users should explain their ratings in three categories. These categories are as follows: Health, preparation, and individual preferences, which have reasons in the form of a check box like $too\ many\ ingredients$ or $my\ favorites\ ingredients$. The reasons

[15] Previous version is located in section 6 of
https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2014-sem/sippel/bericht.pdf

[16] Previous version is located in section 6 of
https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2014-aw2/sippel/bericht.pdf

are divided into positives and negatives. The recipes are not randomly chosen — metadata is available for filtering recipes for vegetarian or lactose-intolerant users. It is assumed that the user consumes the food immediately; hence, there is a time-dependent filter. Only those dishes are chosen that are appropriate for the actual time of the day, such as breakfast in the morning. The learning data comprises the results, the ratings, and the reasons.

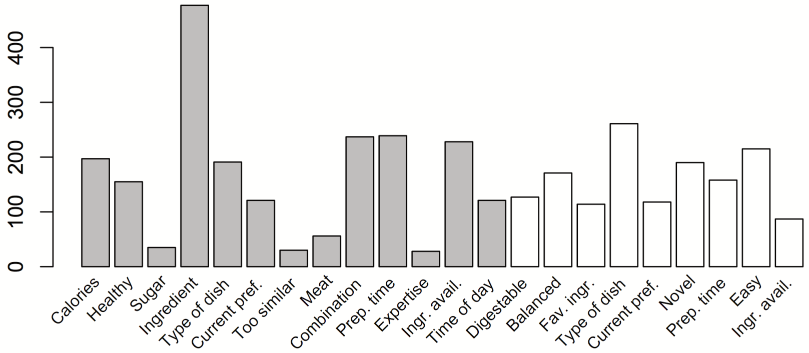


Figure 3.29.: Frequencies of reasons for recommendation ratings [HLE12, p. 21 Figure 2]

The diagram shows the frequencies of the reasons (Figure 3.29). The gray bars on the left side indicate negative reasons; the white bars on the right side are positive reasons. The *ingredients* that are disliked catch the eye in the diagram, while wrong *combinations* and *preparation time* that are too long are also often selected by the users. One the positive side, the *kind of dishes*, *preparation time* and *easy preparation* are used frequently.

The problem with this result is that the dependencies of the data are not clear. *preparation time* and *easy preparation* are dependent on the actual context; the reasons are not always valid. In a linear model analysis, *ingredients* and *combination* are the most significant factors [HLE12, p. 20].

Another example of data dependencies is the ratings of correlated calories (Figure 3.30). Users who select health reasons are often classified as being part of the health-conscious user group, while the rest comprise the unhealthy group. As a result, unhealthy users give high ratings to recipes with more calories, while the healthy users give low ratings to such recipes.

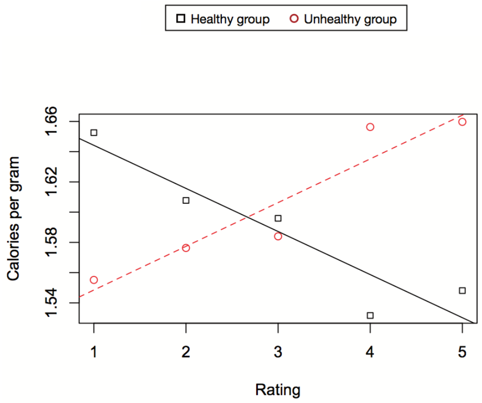


Figure 3.30.: Correlation between calories and ratings of recommendation [HLE12, p. 21 Figure 3]

Since the recommendations based on the ingredients are more precise, they are independent of the actual context [HLE12, p. 21].

Such recommendations are domain-dependent and have to be made in accordance with expert knowledge [SG11, p. 3]. In order to know how precise a recommendation is, it is necessary to ask a domain expert. Knowledge elicitation aims to extract knowledge from domain experts and present formal knowledge in ontologies for knowledge-dependent applications such as expert systems [Coo94, p. 801]. Expert systems uses knowledge to find recommendation for solving complex problems and decision-making tasks. Knowledge elicitation is a process of collection knowledge from domain expert. Alternatives to get knowledge are machine-learning techniques but knowledge elicitation is focused on extracting knowledge from humans. This also includes analysis of written manuals and letters to understand the domain.

The methods for initial conceptualization includes interviews and analysis of how domain experts make decisions during the interviews. It follow deeper structured interviews to understands concepts. A higher number of domain experts are interviewed to aggregate the input [Coo94, p. 821]. Data collection is used to persist knowledge such as rating the similarity of a pairs of concepts to get feedback of the understanding.

Knowledge elicitation needs a large number of available domain experts and interviews cost much time, thats the hard bottleneck of these technique, nevertheless it is applied simplified as possible for example in context of semantic web to create ontologies which are necessary to understand domain-specific web-accessible content [SS15, p. 3].

A recommender system and three experts who are isolated from each other can make one recommendation each for a single example; if all recommend the same, then it is a precise recommendation. Experts have different areas of interest, [McD83, p. 105], and so their focuses are different, which affects the recommendation. Such a validation has to fail. If the result of each recommendation would have to be acceptable only to each expert, then every expert could have a different opinion, but they could agree.

A validation needs a hypothesis, such as the recommender systems, which would be better than recommendations by domain expert. There are controlled variables, such as a static testing set and variables, which are focused on the test. The last is the generalization power, which shows how stable the conclusions are in different contexts.

There are three types of experiments [SG11, p. 10]: Offline experiments with a static testing set and feedback, such as by domain experts, can be used to test whether an accepted recommendation is the recommendation set (used in [FB10],[GEFT$^+$15]). In a user study, the expert would use the recommender system directly; it results in feedback about the use case, understandability, and the expected results for the testing set. A user study is only a qualitative measurement. It has no statistical significance. Online evaluations are used by real users like bartenders for real such tasks as for guests who are seeking recommendations.

Summary

Domain-specific recommendations need expert knowledge and a validation of acceptability by experts; a first validation is an offline study. Than a user study in small group is possible. An online study is the last validation step which has many dynamic variables; the users are not known or can be preselected. The users cannot be forced to use this recommendation system. This type of validation needs a high reliability of the recommendations and a lot of time to deal with this dynamic.

3.8. Challenges for experiments

There are four main challenges associated with the archiving of the defined objects. The first is to understand the domain-specific knowledge. This knowledge is not explicitly written down. Cocktail books contains recipes and additional information about best practice, which represent implicit knowledge. This have to be extracted and written down explicitly by mapping to an ontology. For the modeling basic categories are chosen. Therefore, the quality of the ontology needs to be the focus in every step of development. Missing or imprecise information entails major risks for this approach, since unrecognized data is lost data.

Cocktail books primarily contain recipes. Such recipes contain a title, the names of the ingredients, and a list of ingredients. The ingredients are mostly described in terms of quantities, which can either be concrete units of measurement or only proportions. Depending on whether the authors are American or British, the units can be Imperial ones, US customary measurement systems, or metric. There is also additional information about the ways of preparation (whether to shake or stir the cocktail) as well as which glassware would be useful. A target structure is needed, which presents a recipe in a structure based on knowledge in the ontology. For presentation a structure is necessary that is readable to humans such as domain experts. A list of ingredients with quantities and units is considered for finding out similar recipes, since they describe the structure of a recipe. A presentation that is readable to humans needs information such as a title, the original names of ingredients, and meta information. However, this is not part of a recommendation.

The second challenge is to deal with huge volumes of data. From the KDD perspective, the available data is the starting point. There are various sources for cocktail recipes — books, both historical and new ones, magazines, blogs, and online cocktail databases. From a technical point of view, the cocktail database is the easiest thing. But there are no open interfaces and often, metadata is missing, such as author names, time stamps, or descriptive text. There is an example[17] that presents recipes from some historic books, but the connection between recipes and books is missing. In a example[18] of a community-driven website, there are often strongly similar recipes that have an excessive rate of sweetness or cream. Quality management is unfortunately missing.

In the absence of an interface, the websites have to be parsed, so it is a better alternative to parse books or blogs directly. Meta information is available, content quality is controllable, and the authors are known. On the one hand, from the perspective of users, meta-information is needed to classify recipes; and from the technical perspective, it helps to identify relationships in time and space. On the other hand, there is a lot of unnecessary information, such as an introduction, page numbers, etc. The main aspects of a target structure are features that are useful for finding patterns in the recipe collection. For finding patterns, the extracted data have to be valid, therefore it is an advantage to process massive volume of data because outliers don't impair the result because these are not significant in the huge set of data anymore.

The third challenge is the recommendation based on a distance function. Due to the cold start problem, it is only possible to use collaborative filtering with available user data, but without user data, it is necessary to know something about the items to make a recommendation. The user has to know the example of a favorite to get a similar, though not obvious, item as a recommendation. The favorite is the positively rated item by the user. Assuming the user is

[17] kindredcocktails.com
[18] cocktaildatenbank.de

interested in two or more very different recipes, it cannot be concluded what he prefers at a specific time; therefore, only a rated item is considered for recommendation.

This is a content-based recommendation that uses only the similarity between items. For this a modeling of ingredients by flavors is in focus. This is needed for identifying the similarity of two cocktails. The extracted recipes are data mined by k-nearest neighbor classification based on path similarity approach to get a semantic similarity. For a first understanding how the distances work a metric related to entropy and co-occurrence metric.

The fourth challenge is validation. For first experimentations, a coherence and entropy measurement is applicable. But if a recommendation is not carried out in accordance with expert knowledge, then the recommendation would be useless. In terms of development, it is not sufficient to think that this is precise enough; the recommendation would have to compete in the real world. This challenge will depend on the motivation of the experts who will provide support with their knowledge. They will have to understand how this approach can affect.

For validation by domain experts, offline experiments are chosen: A specific group of domain experts, such as bartenders or bloggers, will be shown a set of recipe pairs. The first is the example and the second the potential recommendation. The domain experts are able to rate the validity of the recommendation. Domain experts have different kinds of backgrounds and experience. Some may be working as bartenders; others may be connoisseurs during their leisure time. Bartenders would be more focused on well-known and easily made drinks, while connoisseurs would focus more on experimental drinks. Again, experiences with respect to time or variety could be very different. Therefore, it is necessary to test them with handmade pairs. Both types of pairs contain appropriate, obvious, and unacceptable ones. Assuming that enough domain experts can be motivated and enough usable ratings are taken, the results show the precision of the chosen distance function, in accordance with the experts' knowledge.

The following experiments contain these four challenges for answering the primary question: Whether or not a knowledge-based distance function will have enough precision to domain experts.

4. Experiments

The experiments are separated into several small experiments which aims to achieve the four challenges (section 3.8). The result of each experiment is used for the following experiment. The first two experiments are designed to archive the first challenge: The first experiment in section 4.1 uses a domain-specific survey with domain experts to understand the field of cocktail recipes. The aim is to learn how data depends on recommendation. Section 4.2 contains an experiment of feature extraction of classic recipes to a defined target structure that presents the recipe. An ontology is designed to store the features such as ingredients or measurement units in a hierarchy. The extraction is used to redesign the target structure. For understanding huge volume of data (challenge two) this structure is used in section 4.3 to design and implement a parser for cocktail recipes that uses several phases of recognition, cleaning, contextualization, and domain-specific reasoning to get a comparable set of features. For challenge three section 4.4 describes an experiment of distances between classic recipes that contain the quantity-based ingredient distance and a quantity-based balance distance to get a first measurement of precision in manageable dataset. The last experiment in section 4.5 handle the challenge four by doing an validation of example-based recommendation based on balance and ingredient distance. Classics are chosen as examples and parsed recipes are used for recommendation. The domain experts validate how acceptable these recommendations are. The section 4.6 considers the resultant architecture including the used libraries. The last section 4.7 considers the conclusion and future work of experiments.

4.1. Understanding the field of cocktail recipes with domain experts

In the domain of cocktails, explicit assured knowledge about cocktails and the recommendation of cocktails is missing. There are manuals and cocktail recipe books, but the apprenticeship is based on voluntarism provided by accomplished bartenders who have written the books. There is no related research. Therefore, at first knowledge has to be received to find appropriate recommendations (challenge one). Domain experts are asked in a survey which parts of a cocktail recipe and which information about the guest are necessary for cocktail recommendations.

The target group comprises domain experts such as bartenders, bar owners, connoisseurs, and interested guests, who are invited to participate in the survey through online communities and social media portals such as Twitter. The survey was offered in English and in German. In this section, German answers are translated into English. Twenty domain experts aged between 22 and 48 years answer all questions of the survey. Three people claim to work in a bar or own a bar. The rest consider themselves as connoisseurs or guests in a bar. Most of them have experiences in the domain of cocktails of about 3–10 years (Figure 4.1).

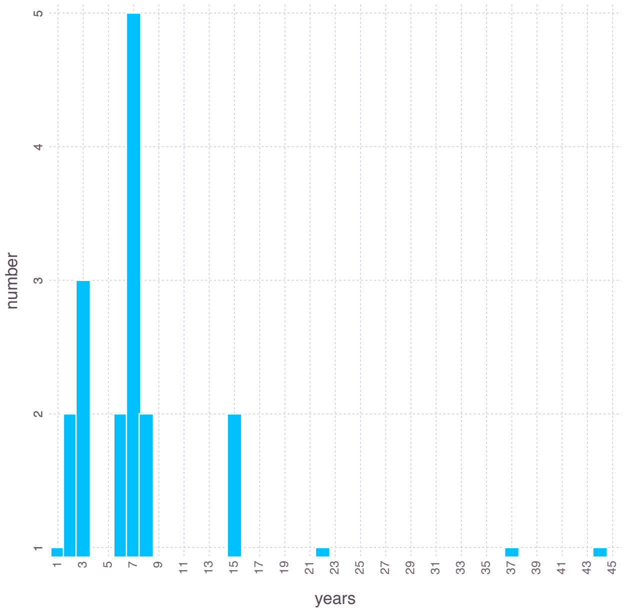
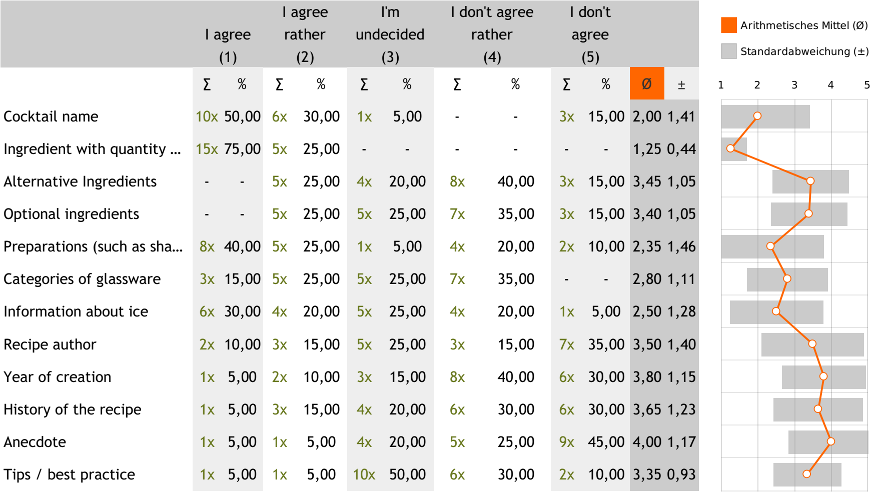


Figure 4.1.: Result of question which is located in section A.3

4.1.1. Minimal recipe

The necessary parts of a minimal recipe that can be prepared are considered in the next question (Figure 4.2). Ingredients with quantity are described as necessary. Cocktail names are agreed by most experts. A possible reason for this is that they are associated this name with a other known recipes. The mean values of preparations, glassware, and ice are almost “undecided” due to a tendential uniform distribution. This statement is unclear. Alternative and optional ingredients are logically considered not necessary. This is an indicator of the data for a good relation to reality. However, author, year of creation, history, anecdotes, and best practice are not considered necessary.

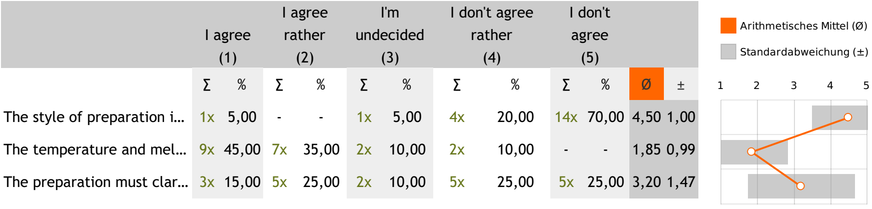


	I agree (1)		I agree rather (2)		I'm undecided (3)		I don't agree rather (4)		I don't agree (5)			
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Ø	±
Cocktail name	10x	50,00	6x	30,00	1x	5,00	-	-	3x	15,00	2,00	1,41
Ingredient with quantity ...	15x	75,00	5x	25,00	-	-	-	-	-	-	1,25	0,44
Alternative Ingredients	-	-	5x	25,00	4x	20,00	8x	40,00	3x	15,00	3,45	1,05
Optional ingredients	-	-	5x	25,00	5x	25,00	7x	35,00	3x	15,00	3,40	1,05
Preparations (such as sha...	8x	40,00	5x	25,00	1x	5,00	4x	20,00	2x	10,00	2,35	1,46
Categories of glassware	3x	15,00	5x	25,00	5x	25,00	7x	35,00	-	-	2,80	1,11
Information about ice	6x	30,00	4x	20,00	5x	25,00	4x	20,00	1x	5,00	2,50	1,28
Recipe author	2x	10,00	3x	15,00	5x	25,00	3x	15,00	7x	35,00	3,50	1,40
Year of creation	1x	5,00	2x	10,00	3x	15,00	8x	40,00	6x	30,00	3,80	1,15
History of the recipe	1x	5,00	3x	15,00	4x	20,00	6x	30,00	6x	30,00	3,65	1,23
Anecdote	1x	5,00	1x	5,00	4x	20,00	5x	25,00	9x	45,00	4,00	1,17
Tips / best practice	1x	5,00	1x	5,00	10x	50,00	6x	30,00	2x	10,00	3,35	0,93

Figure 4.2.: Result of question which is located in section A.4

4.1.2. Preparation and ice

The next question considers on what the selection of preparation depends (Figure 4.3). Most domain experts says that the preparation is important for the cocktail result, but they say also the bartender decides because of the intended temperature and melting water. If the recipes contain information about preparation, it is only a recommendation.



	I agree (1)		I agree rather (2)		I'm undecided (3)		I don't agree rather (4)		I don't agree (5)			
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Ø	±
The style of preparation i...	1x	5,00	-	-	1x	5,00	4x	20,00	14x	70,00	4,50	1,00
The temperature and mel...	9x	45,00	7x	35,00	2x	10,00	2x	10,00	-	-	1,85	0,99
The preparation must clar...	3x	15,00	5x	25,00	2x	10,00	5x	25,00	5x	25,00	3,20	1,47

Arithmetisches Mittel (Ø)
Standardabweichung (±)
1
2
3
4
5

Figure 4.3.: Result of question which is located in section A.5

The question about ice (Figure 4.4) shows that this information is mostly used only as a recommendation. It depends on the cocktails, and sometimes information about ice is useful.

	I agree (1)		I agree rather (2)		I'm undecided (3)		I don't agree rather (4)		I don't agree (5)			
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Ø	±
I always follow the instru...	1x	5,00	9x	45,00	1x	5,00	5x	25,00	4x	20,00	3,10	1,33
It is only a recommendati...	11x	55,00	4x	20,00	1x	5,00	4x	20,00	-	-	1,90	1,21
I always use the same typ...	1x	5,00	2x	10,00	2x	10,00	4x	20,00	11x	55,00	4,10	1,25
Instructions about ice are...	-	-	6x	30,00	2x	10,00	3x	15,00	9x	45,00	3,75	1,33

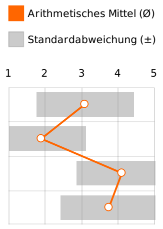


Figure 4.4.: Result of question which is located in section A.6

4.1.3. Substitution of ingredients

The next three questions consider the substitution of ingredients. The first of these questions (Figure 4.5) considers how changeable an unavailable ingredient is. It mostly says that ingredients are substitutable by appropriate ingredients, as well as that the ingredients have to be available and agreed. An interpretation of these results is that the opportunity of substitution depends strongly on the ingredient. The next two questions investigate what “appropriate” means.

	I agree (1)		I agree rather (2)		I'm undecided (3)		I don't agree rather (4)		I don't agree (5)			
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Ø	±
I can prepare a cocktail if...	5x	25,00	5x	25,00	2x	10,00	6x	30,00	2x	10,00	2,75	1,41
I cannot have all ingredie...	6x	30,00	9x	45,00	4x	20,00	1x	5,00	-	-	2,00	0,86
I cannot have all ingredie...	-	-	5x	25,00	-	-	5x	25,00	10x	50,00	4,00	1,26

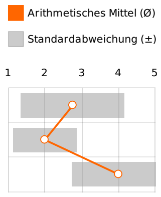


Figure 4.5.: Result of question which is located in section A.7

The second question (Figure 4.6) considers the genever as an example, which has to be substituted. It mostly says that every ingredient that is kind of a genever is appropriate. Nevertheless, there are different qualities and specialties; therefore, the bartender has to decide. It is not enough to use ingredients that contain only the same source such as gin (juniper) or the same type of production (such as distilled ingredients).

	I agree (1)		I agree rather (2)		I'm undecided (3)		I don't agree rather (4)		I don't agree (5)			
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Ø	±
I could use any ingredien...	14x	70,00	4x	20,00	1x	5,00	1x	5,00	-	-	1,45	0,83
Any ingredient that is dist...	-	-	2x	10,00	2x	10,00	8x	40,00	8x	40,00	4,10	0,97
It is a distilled ingredient...	-	-	-	-	-	-	1x	5,00	19x	95,00	4,95	0,22
The recipe is imprecise. I...	-	-	-	-	2x	10,00	8x	40,00	10x	50,00	4,40	0,68

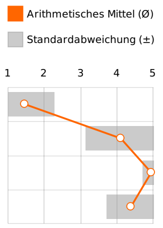
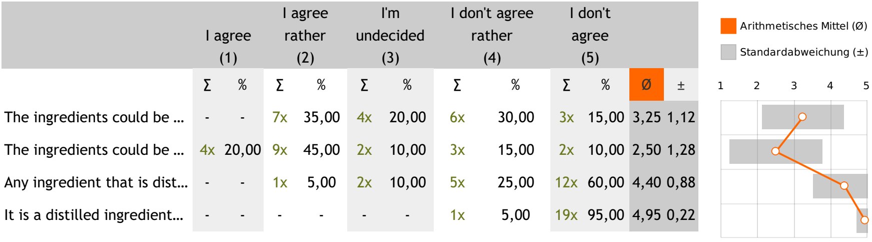


Figure 4.6.: Result of question which is located in section A.8

The third question (Figure 4.6) considers a concrete genever product that has to be substituted. The ingredients could be very special, but if this concrete product is not known, it mostly will substitute this ingredient with another genever product. The results are less ambiguous than the question before because a concrete product is considered. If a more abstract assignment is given, not just a special ingredient is needed. This increases the scope of opportunities. In this case, it is possible that a very special ingredient is needed, but it is not explicit known.

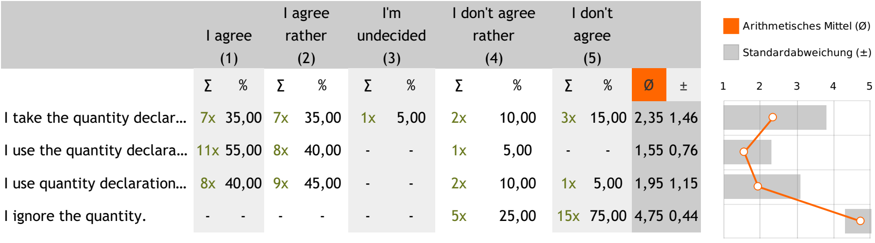


	I agree (1)		I agree rather (2)		I'm undecided (3)		I don't agree rather (4)		I don't agree (5)			
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Ø	±
The ingredients could be ...	-	-	7x	35,00	4x	20,00	6x	30,00	3x	15,00	3,25	1,12
The ingredients could be ...	4x	20,00	9x	45,00	2x	10,00	3x	15,00	2x	10,00	2,50	1,28
Any ingredient that is dist...	-	-	1x	5,00	2x	10,00	5x	25,00	12x	60,00	4,40	0,88
It is a distilled ingredient...	-	-	-	-	-	-	1x	5,00	19x	95,00	4,95	0,22

Figure 4.7.: Result of question which is located in section A.9

4.1.4. Quantity of an ingredient

The next question considers how the quantity information of a recipe is used (Figure 4.8). Mostly use the quantity literally, which is partially to understand the idea of the recipes. For qualitative units, they decide on their own. This shows that the quantities are important but have room for interpretation.

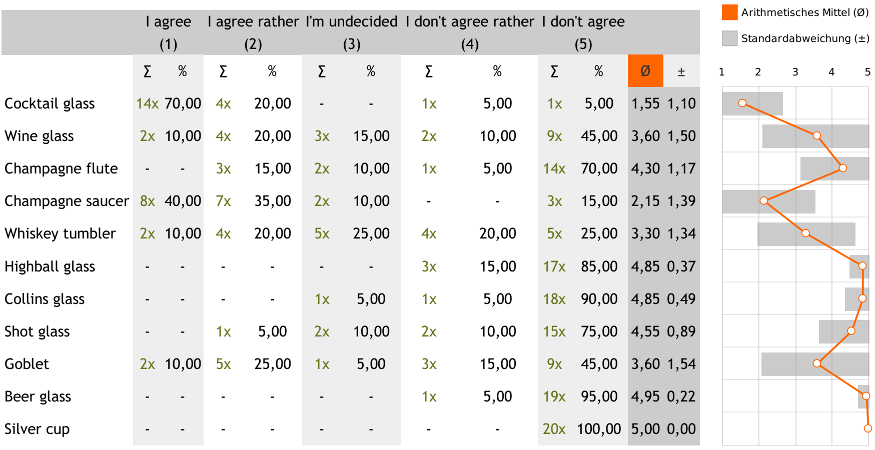


	I agree (1)		I agree rather (2)		I'm undecided (3)		I don't agree rather (4)		I don't agree (5)			
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Ø	±
I take the quantity declar...	7x	35,00	7x	35,00	1x	5,00	2x	10,00	3x	15,00	2,35	1,46
I use the quantity declara...	11x	55,00	8x	40,00	-	-	1x	5,00	-	-	1,55	0,76
I use quantity declaration...	8x	40,00	9x	45,00	-	-	2x	10,00	1x	5,00	1,95	1,15
I ignore the quantity.	-	-	-	-	-	-	5x	25,00	15x	75,00	4,75	0,44

Figure 4.8.: Result of question which is located in section A.10

4.1.5. Glassware

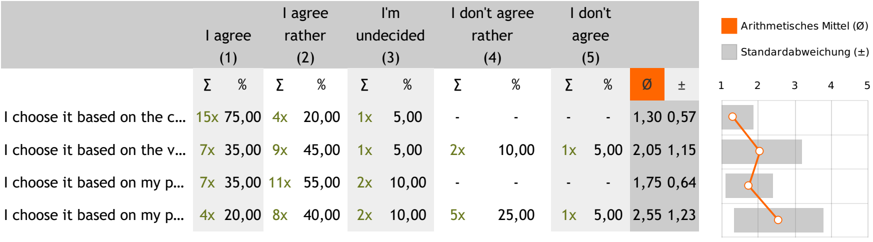
The next question considers which cocktail glass is chosen for a Daiquiri cocktail. The given recipe contains no information about glassware. Most people choose a cocktail glass or a champagne saucer, both of which contain a stem and a bowl. Some agree to use a wine glass and goblet. The figure of the glass is common; it differs only in size.



	I agree (1)		I agree rather (2)		I'm undecided (3)		I don't agree rather (4)		I don't agree (5)			
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Ø	±
Cocktail glass	14x	70,00	4x	20,00	-	-	1x	5,00	1x	5,00	1,55	1,10
Wine glass	2x	10,00	4x	20,00	3x	15,00	2x	10,00	9x	45,00	3,60	1,50
Champagne flute	-	-	3x	15,00	2x	10,00	1x	5,00	14x	70,00	4,30	1,17
Champagne saucer	8x	40,00	7x	35,00	2x	10,00	-	-	3x	15,00	2,15	1,39
Whiskey tumbler	2x	10,00	4x	20,00	5x	25,00	4x	20,00	5x	25,00	3,30	1,34
Highball glass	-	-	-	-	-	-	3x	15,00	17x	85,00	4,85	0,37
Collins glass	-	-	-	-	1x	5,00	1x	5,00	18x	90,00	4,85	0,49
Shot glass	-	-	1x	5,00	2x	10,00	2x	10,00	15x	75,00	4,55	0,89
Goblet	2x	10,00	5x	25,00	1x	5,00	3x	15,00	9x	45,00	3,60	1,54
Beer glass	-	-	-	-	-	-	1x	5,00	19x	95,00	4,95	0,22
Silver cup	-	-	-	-	-	-	-	-	20x	100,00	5,00	0,00

Figure 4.9.: Result of question which is located in section A.11

The reason for the choice of glassware is in focus of the next question (Figure 4.10). The responses are ambiguous. They choose on the basis of the features of a cocktail, but also the volume of the cocktail. The personal opinion of the bartender and the specific context also affect the choice. The figure of glassware is classifiable by a given recipe and context; therefore, it is not necessary information for a cocktail recipe.

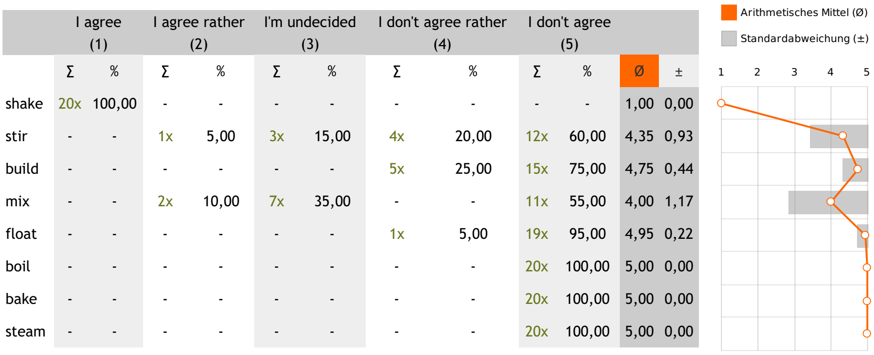


	I agree (1)		I agree rather (2)		I'm undecided (3)		I don't agree rather (4)		I don't agree (5)			
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Ø	±
I choose it based on the c...	15x	75,00	4x	20,00	1x	5,00	-	-	-	-	1,30	0,57
I choose it based on the v...	7x	35,00	9x	45,00	1x	5,00	2x	10,00	1x	5,00	2,05	1,15
I choose it based on my p...	7x	35,00	11x	55,00	2x	10,00	-	-	-	-	1,75	0,64
I choose it based on my p...	4x	20,00	8x	40,00	2x	10,00	5x	25,00	1x	5,00	2,55	1,23

Figure 4.10.: Result of question which is located in section A.12

4.1.6. Choice of preparation

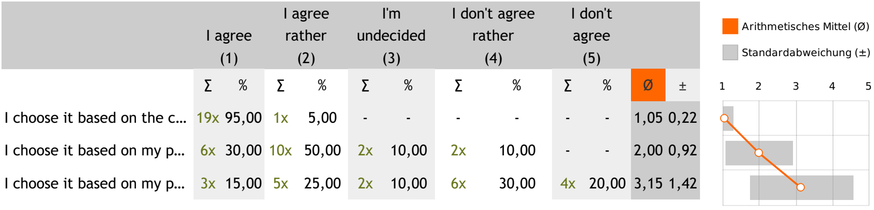
The preparation of a Daiquiri cocktail is considered in the next question (Figure 4.11). The given recipe contains no information about preparation. Nevertheless, all interviewed persons choose the preparation of shake as an appropriate preparation. Only a few persons agree on stir or mix (generalization of stir and shake). All other opportunities such as build or float are disagreed.



	I agree (1)		I agree rather (2)		I'm undecided (3)		I don't agree rather (4)		I don't agree (5)			
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Ø	±
shake	20x	100,00	-	-	-	-	-	-	-	-	1,00	0,00
stir	-	-	1x	5,00	3x	15,00	4x	20,00	12x	60,00	4,35	0,93
build	-	-	-	-	-	-	5x	25,00	15x	75,00	4,75	0,44
mix	-	-	2x	10,00	7x	35,00	-	-	11x	55,00	4,00	1,17
float	-	-	-	-	-	-	1x	5,00	19x	95,00	4,95	0,22
boil	-	-	-	-	-	-	-	-	20x	100,00	5,00	0,00
bake	-	-	-	-	-	-	-	-	20x	100,00	5,00	0,00
steam	-	-	-	-	-	-	-	-	20x	100,00	5,00	0,00

Figure 4.11.: Result of question which is located in section A.13

The choice of preparation is considered in the next question (Figure 4.12). All interviewed persons agree that the choice is based on the characteristics of a cocktail. Compared to the choice of glassware, personal opinion and context are less agreed; however, these also affect the choice.



	I agree (1)		I agree rather (2)		I'm undecided (3)		I don't agree rather (4)		I don't agree (5)			
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Ø	±
I choose it based on the c...	19x	95,00	1x	5,00	-	-	-	-	-	-	1,05	0,22
I choose it based on my p...	6x	30,00	10x	50,00	2x	10,00	2x	10,00	-	-	2,00	0,92
I choose it based on my p...	3x	15,00	5x	25,00	2x	10,00	6x	30,00	4x	20,00	3,15	1,42

Figure 4.12.: Result of question which is located in section A.14

4.1.7. Choice of volume

The resulted volume of a prepared cocktail is considered in the next question (Figure 4.13). The given Daiquiri recipe contains only relative quantities that need to be interpreted. Fifteen out of 20 persons prefer a volume in the range of 8–12 cl. There are four outliers with very low values and one that prefers 90 cl (data point excluded from diagram). The glass has to be filled adequately so as to avoid overfilling or a very small quantity. Therefore, these outliers result from missing understanding.

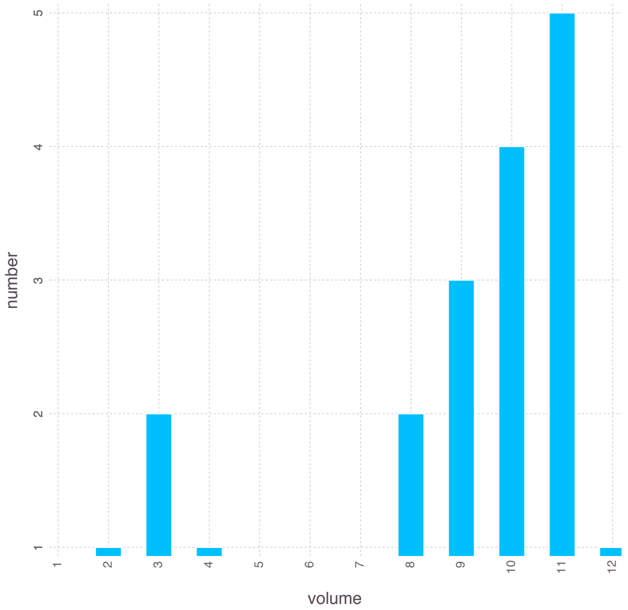
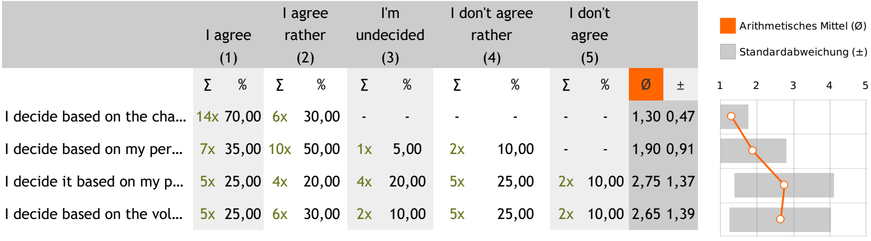


Figure 4.13.: Result of question which is located in section A.15

The reason for the chosen volume is considered in the next question (Figure 4.14). Most people agree that their choice is based on the characteristics of the given cocktail, but personal opinion also plays a role. The context and the volume of the glass are less agreed. Basically, the responses show that a volume of cocktail can be derived by the interviewed person. It depends mostly on the characteristics of the cocktail.



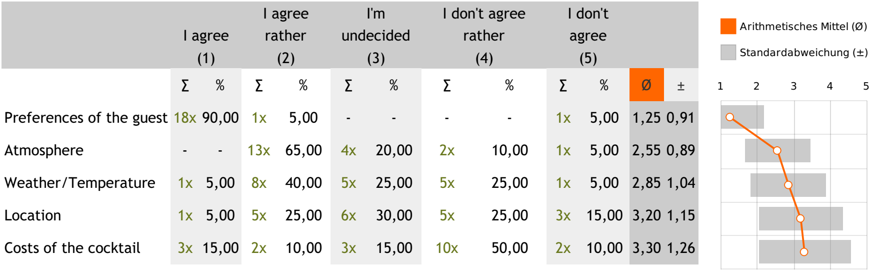
	I agree (1)		I agree rather (2)		I'm undecided (3)		I don't agree rather (4)		I don't agree (5)			
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Ø	±
I decide based on the cha...	14x	70,00	6x	30,00	-	-	-	-	-	-	1,30	0,47
I decide based on my per...	7x	35,00	10x	50,00	1x	5,00	2x	10,00	-	-	1,90	0,91
I decide it based on my p...	5x	25,00	4x	20,00	4x	20,00	5x	25,00	2x	10,00	2,75	1,37
I decide based on the vol...	5x	25,00	6x	30,00	2x	10,00	5x	25,00	2x	10,00	2,65	1,39

Arithmetisches Mittel (Ø)
Standardabweichung (±)
1
2
3
4
5

Figure 4.14.: Result of question which is located in section A.16

4.1.8. Recommendation of recipe

The next question considers the recommendation of a cocktail. The first of these considers the needed information for a recommendation for a cocktail (Figure 4.15). They mostly agree that the preference of the guest is necessary. Information about the context, such as atmosphere, weather, location, and cost of the cocktail, results in a widespread distribution of response. This information is not tangible enough to associate these with a recommendation.

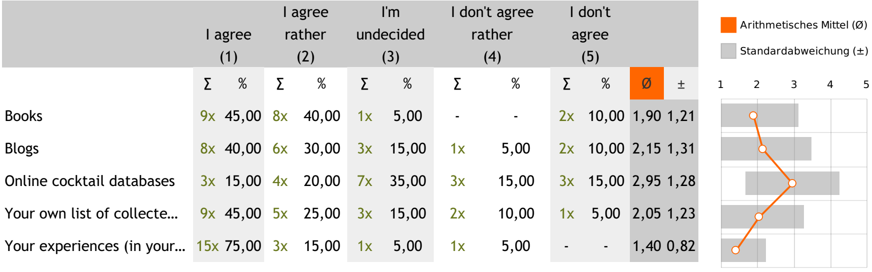


	I agree (1)		I agree rather (2)		I'm undecided (3)		I don't agree rather (4)		I don't agree (5)			
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Ø	±
Preferences of the guest	18x	90,00	1x	5,00	-	-	-	-	1x	5,00	1,25	0,91
Atmosphere	-	-	13x	65,00	4x	20,00	2x	10,00	1x	5,00	2,55	0,89
Weather/Temperature	1x	5,00	8x	40,00	5x	25,00	5x	25,00	1x	5,00	2,85	1,04
Location	1x	5,00	5x	25,00	6x	30,00	5x	25,00	3x	15,00	3,20	1,15
Costs of the cocktail	3x	15,00	2x	10,00	3x	15,00	10x	50,00	2x	10,00	3,30	1,26

🟧 Arithmetisches Mittel (Ø)
▨ Standardabweichung (±)
1
2
3
4
5

Figure 4.15.: Result of question which is located in section A.17

The next question considers the preferred sources of recipes that are used by the interviewed person. In general, books, blogs, personal lists and experiences are used, but online databases are less used.



	I agree (1)		I agree rather (2)		I'm undecided (3)		I don't agree rather (4)		I don't agree (5)			
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Ø	±
Books	9x	45,00	8x	40,00	1x	5,00	-	-	2x	10,00	1,90	1,21
Blogs	8x	40,00	6x	30,00	3x	15,00	1x	5,00	2x	10,00	2,15	1,31
Online cocktail databases	3x	15,00	4x	20,00	7x	35,00	3x	15,00	3x	15,00	2,95	1,28
Your own list of collecte...	9x	45,00	5x	25,00	3x	15,00	2x	10,00	1x	5,00	2,05	1,23
Your experiences (in your...	15x	75,00	3x	15,00	1x	5,00	1x	5,00	-	-	1,40	0,82

Figure 4.16.: Result of question which is located in section A.18

The next question (section A.19) considers which cocktail recommendation for a guest would be chosen if a favorite cocktail were known. In this case, Daiquiri is the favorite, which contains rum, lime, and sugar. Four responses contain two recommendations. The 22 resulted recommendations contain 12 recommendations with rum, lime, sugar, or sweet liquor, as well as some additional ingredients such as champagne, herbs, and bitters. The characteristic is similar and the main ingredient is equal or very similar to the given favorite.

- 2x Santa Marta: Rum, lime, sugar, shake, float Kirschwasser
- Ranglum (Rum, Falernum, lime, sugar)
- Mai Tai: Rum, lime, Orgeat, shake, tumbler, crushed Ice
- Royal Bermuda Yacht Club (Rum, lime, Falernum, sugar, Triple Sec, shake, cocktail glass)
- Rum Sour

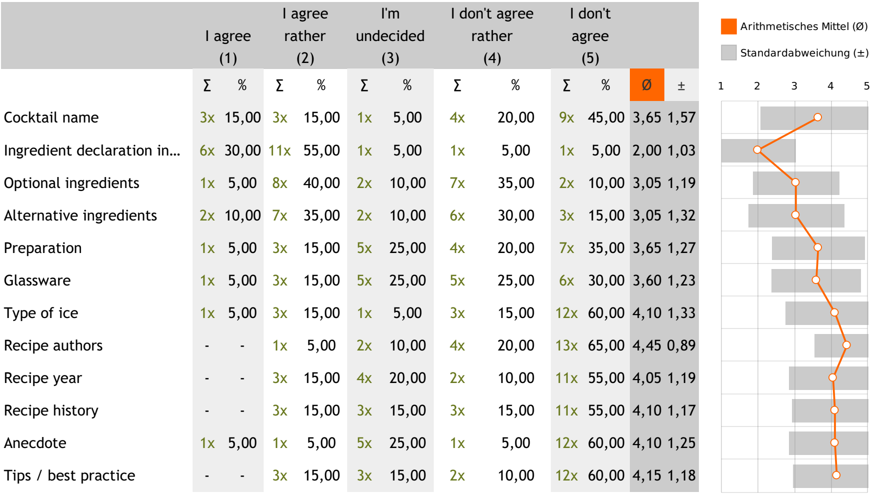
- 2x Hemingway Daiquiri, 6 cl Rum, 1 cl Maraschino, 1 cl sugar syrup, 2 cl lime juice, 3 cl pink grapefruit juice
- Pink Cuban Highball: Rum, lime, sugar, Peychaud's, stirred, filled with champagne , Highball glass with two ice bullets
- Szechuan Daiquiri: Angostura 1919, lime, sugar, Szechuan pepper, shake, cocktail glass
- Old Cuban, Rum, lime, sugar, mint, champagne, brandy snifter
- Other rum-based Sours - Rum Sour mit powerful Rum, Ranglum, Daiquiri-variants (sugar cane can be substituted by liquor), other Sours, Fizz, also Collins or Mojito

Ten recommendations contain a spirit, which is not a rum. Additionally, lime or lemon juice and also herbs, bitters, and liquors are used. An exception is the Gimlet as well as gin and a limejuice cordial, but with a higher rate of alcohol. The characteristic is similar, but the ingredients are very different to get a not-too-obvious recipe in light of the given favorite.

- Slivopolitan (3 Slivovic, 2 Cointreau, 2 plum jam, 1 lime juice)
- Sidecar (Cognac, lemon, Triple Sec, shake, cocktail glass)
- 2x Ti Punch: Rhum Agricole, lime, sugar, shake/stir, tumbler
- Pisco Sour, Pisco, lime, sugar, egg white, shake, thin glass, with Angostura Bitters
- Gin Basil Smash
- Tommys Margarita, Reposado Tequila, limejuice, agave syrup, shake, cocktail glass
- Tequila Sour: Tequila, limejuice, sugar, shake, cocktail glass
- Margarita
- Gimlet

The answers are not randomly chosen. Two concepts of recommendation are recognizable, which is an important basis for further experiments.

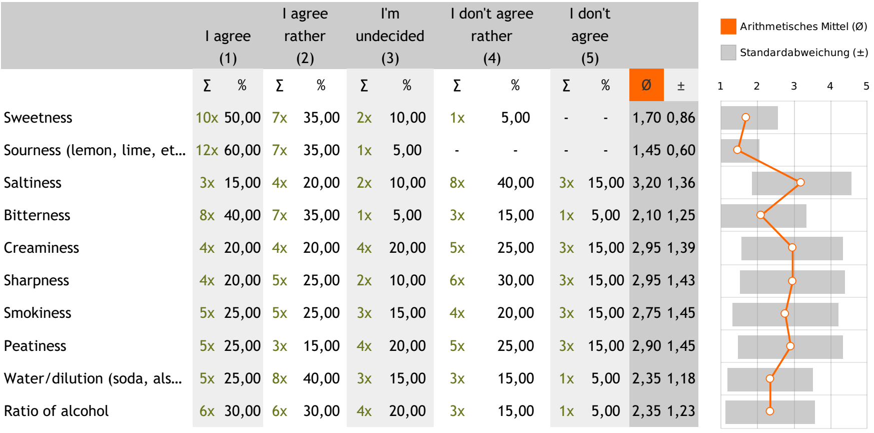
The next question considers which element of the favorite cocktail influences the recommendation (Figure 4.17). The ingredients including the quantity are most agreed. Other information such as the name or preparation and glassware is not agreed.



	I agree (1)		I agree rather (2)		I'm undecided (3)		I don't agree rather (4)		I don't agree (5)			
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Ø	±
Cocktail name	3x	15,00	3x	15,00	1x	5,00	4x	20,00	9x	45,00	3,65	1,57
Ingredient declaration in...	6x	30,00	11x	55,00	1x	5,00	1x	5,00	1x	5,00	2,00	1,03
Optional ingredients	1x	5,00	8x	40,00	2x	10,00	7x	35,00	2x	10,00	3,05	1,19
Alternative ingredients	2x	10,00	7x	35,00	2x	10,00	6x	30,00	3x	15,00	3,05	1,32
Preparation	1x	5,00	3x	15,00	5x	25,00	4x	20,00	7x	35,00	3,65	1,27
Glassware	1x	5,00	3x	15,00	5x	25,00	5x	25,00	6x	30,00	3,60	1,23
Type of ice	1x	5,00	3x	15,00	1x	5,00	3x	15,00	12x	60,00	4,10	1,33
Recipe authors	-	-	1x	5,00	2x	10,00	4x	20,00	13x	65,00	4,45	0,89
Recipe year	-	-	3x	15,00	4x	20,00	2x	10,00	11x	55,00	4,05	1,19
Recipe history	-	-	3x	15,00	3x	15,00	3x	15,00	11x	55,00	4,10	1,17
Anecdote	1x	5,00	1x	5,00	5x	25,00	1x	5,00	12x	60,00	4,10	1,25
Tips / best practice	-	-	3x	15,00	3x	15,00	2x	10,00	12x	60,00	4,15	1,18

Figure 4.17.: Result of question which is located in section A.20

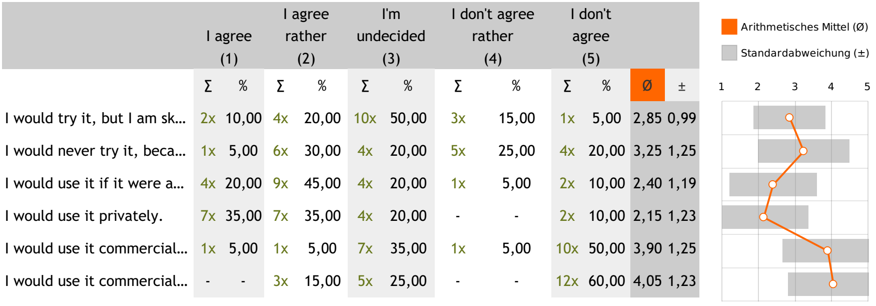
The next question considers which flavors of the favorite influence the recommendation. Sweetness and sourness are mostly agreed. Alcohol ratio, dilution, and bitterness are rather agreed. Creaminess, sharpness, peatiness, and smokiness are less agreed while saltiness is usually not agreed. Compared to the previous question, the flavors are more agreed for influencing the recommendation than the elements of the cocktail recipe. This answer sustains the results of question section A.19 that shows recommendation of the same flavors.



	I agree (1)		I agree rather (2)		I'm undecided (3)		I don't agree rather (4)		I don't agree (5)			
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Ø	±
Sweetness	10x	50,00	7x	35,00	2x	10,00	1x	5,00	-	-	1,70	0,86
Sourness (lemon, lime, et...	12x	60,00	7x	35,00	1x	5,00	-	-	-	-	1,45	0,60
Saltiness	3x	15,00	4x	20,00	2x	10,00	8x	40,00	3x	15,00	3,20	1,36
Bitterness	8x	40,00	7x	35,00	1x	5,00	3x	15,00	1x	5,00	2,10	1,25
Creaminess	4x	20,00	4x	20,00	4x	20,00	5x	25,00	3x	15,00	2,95	1,39
Sharpness	4x	20,00	5x	25,00	2x	10,00	6x	30,00	3x	15,00	2,95	1,43
Smokiness	5x	25,00	5x	25,00	3x	15,00	4x	20,00	3x	15,00	2,75	1,45
Peatiness	5x	25,00	3x	15,00	4x	20,00	5x	25,00	3x	15,00	2,90	1,45
Water/dilution (soda, als...	5x	25,00	8x	40,00	3x	15,00	3x	15,00	1x	5,00	2,35	1,18
Ratio of alcohol	6x	30,00	6x	30,00	4x	20,00	3x	15,00	1x	5,00	2,35	1,23

Figure 4.18.: Result of question which is located in section A.21

The last question considers how much a recommendation service is preferred. For private use and if the situation is passed, such service is preferred by the interviewed person. Actual the skepticism of feasibility of such recommendation service is generally high, especially for using in commercial applications. In private area, experiments are always possible and a bad result is without consequences. In a commercial area, the results have to be appropriate, which is not proved actually.



	I agree (1)		I agree rather (2)		I'm undecided (3)		I don't agree rather (4)		I don't agree (5)			
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Ø	±
I would try it, but I am sk...	2x	10,00	4x	20,00	10x	50,00	3x	15,00	1x	5,00	2,85	0,99
I would never try it, beca...	1x	5,00	6x	30,00	4x	20,00	5x	25,00	4x	20,00	3,25	1,25
I would use it if it were a...	4x	20,00	9x	45,00	4x	20,00	1x	5,00	2x	10,00	2,40	1,19
I would use it privately.	7x	35,00	7x	35,00	4x	20,00	-	-	2x	10,00	2,15	1,23
I would use it commercial...	1x	5,00	1x	5,00	7x	35,00	1x	5,00	10x	50,00	3,90	1,25
I would use it commercial...	-	-	3x	15,00	5x	25,00	-	-	12x	60,00	4,05	1,23

Figure 4.19.: Result of question which is located in section A.22

Summary

This qualitative survey shows which information a cocktail recommendation system can use to get an appropriate recommendation. The focus of a cocktail recipe is on the ingredients with their quantities. Preparation, glassware, and ice are not in focus, because this information can be derived from ingredients, opinion, and context. The recommendation for a guest needs to be appropriate to their preferences. Ingredients and in particular their flavors are useful features to describe these preferences. Using a given favorite, the interviewed person recommends either with a focus on the ingredients of the favorite or with a focus on the flavors such as sourness and sweetness and alcohol ratio. The interviewed person would use an automatic recommendation system, but actual skepticism is visible. These experiments give the first understanding of domain in the first challenge, which forms the basis to retain this information in the next experiment.

4.2. Classic recipe extraction by domain experts

The aim of the experiment is to learn how a cocktail recipe is constructed and which information is extractable for further experiments[1]. Assumed that classic recipes, which have been popular for a long time, are distinct from all other recipes. The result is a wide array of recipes from five different historic cocktail books. As the books have different authors, each book shows a unique style of recipes, different ingredients, measurement units, and spellings.

[1] The previous version is located Sections 1–4.3 in https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2015-proj/sippel.pdf

- 1862 Jerry Thomas — How to Mix Drinks: The Bon-Vivants Companion (New York, USA)
- 1882 Harry Johnson — Bartender's Manual (Chicago and New York, USA)
- 1930 Harry Craddock — Savoy Cocktail Book
- 1930 Virginia Elliott and Phil D. Stong — Shake em up!
- 1948 David A. Embury — Fine Art of Mixing Drinks

For extracting recipes out of the books a target data structure for a cocktail recipe is defined (Equation 4.1) which is based on the result of the survey. A cocktail is separated into a title, a list of ingredients, preparation, and a chosen glass. Each ingredient contains a quantity with a unit, value, and the name of the ingredient. The data structure is designed to be also understandable for domain experts, therefore preparation and glassware are not necessary, but if it given, it is a additional recommendation for the bartender.

$$\begin{aligned} Cocktail(title : String, ingredients : List[Ingredient], \\ preparation : String, glassware : String) \\ Ingredient(q : Quanitity, name : String) \\ Quantity(unit : String, value : String) \end{aligned} \tag{4.1}$$

The extraction by domain expert is done manually, which is persist in a human-readable format — XML — to get a human-readable format for the recipes (examples placed in Appendix B). Because of the given data structure, the feature extraction does not have to decide whether a string is an ingredient or preparation. This features are already classified by the domain expert.

The main task of the ontology component is to find an item for a given name and taxonomy. For this, a concrete ontology has to be designed. The ontology component contains categories separated into the following taxonomies: Ingredient, preparation, glassware, and units. These are different kinds of items that are addressed and identified by a unique URI.

The RDF model contains a set of triples $(resource, property, atomic\ value)$. Instead of atomic values, such as labels or titles, there could be other triples as well. This nested definition is used to model trees. Each property can have a URI to ensure a unique address. The property describes the edge that connects the left with the right one.

There are predefined properties. The minimal structure of RDF (Listing 1) contains the root element $rdf : RDF$ with three name spaces [HKRS07]: rdf contains elements such as $property$ or $type$, which are extended by the name space $rdfs$ and contain elements such as

```
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:c="http://www.myclassicbar.com/rdf#">
 ...
</rdf:RDF>
```

Listing 1: Declaration of RDF schemas

$Class$. The name space c is the self-invented name space for the domain-specific elements such as $factor$. The semantic of the elements will be explained in the following chapters.

4.2.1. Ingredients

Each ingredient category is a property (Listing 2) with the type of ingredient, a URI about itself, and a literal name.

```
<rdf:Property
  rdf:type="cocktail://ingredient/basic"
  rdf:about="cocktail://ingredient/gin"
  rdfs:Literal="gin"/>
```

Listing 2: Declaration of a RDF property

The type is referred to the ingredient class (Listing 3). The ingredient class contains two subclasses that represent the basic categories such as *gin* and subordinates such as *London dry gin*. Superordinates like *spirits* are completely excluded, because the shared properties between two *spirits* such as *absinthe* and *gin* are too low.

```
<rdfs:Class rdf:about="cocktail://ingredient">
  <rdfs:label>ingredient</rdfs:label>
</rdfs:Class>
<rdfs:Class rdf:about="cocktail://ingredient/basic">
  <rdfs:label>basic category of ingredient</rdfs:label>
  <rdfs:subClassOf>cocktail://ingredient</rdfs:subClassOf>
</rdfs:Class>
<rdfs:Class rdf:about="cocktail://ingredient/subordinate">
  <rdfs:label>subordinate ingredient</rdfs:label>
  <rdfs:subClassOf>cocktail://ingredient</rdfs:subClassOf>
</rdfs:Class>
```

Listing 3: Declaration of RDF classes

The query to RDF, which is written in SPARQL, has to map $(name, type(ingredient)) \rightarrow Item$ (Listing 4). As SPARQL is a kind of SQL, it uses a *select* query. It allows one to declare triples with bound and free variables. Two kinds of triples are of importance. The first one binds an ingredient $kindof$ to their literal $name$. The $name$ is bound with a filter to an uncapitalized exemplary string of $Plymouth$. The second one binds the ingredient to a *type* that is defined as a subclass of the ingredient class. This is either a basic category or a subordinate.

```
SELECT  ?type ?kindof
WHERE {
  ?kindof  rdfs:Literal ?name .
  ?kindof  rdf:type ?type .
  ?type rdfs:subClassOf "cocktail://ingredient"
  FILTER ( lcase(str(?name)) = "Plymouth" )
}
```

Listing 4: Query of a ingredient written in SPARQL

The data structure is chosen by $type$, which implements the trait (Equation 4.2). The variable $kindof$ is the value uri.

$$trait\ Item\{\ val\ uri : String\ \} \tag{4.2}$$

Names that are not found are a special kind of $item$, which also has a URI (Equation 4.3). Identification with the URI is always possible.

$$cocktail : //unknown/?name \tag{4.3}$$

4.2.2. Preparations

The preparation is represented by a small set of actions such as $stir$ or $shake$. There is no parent of an action. The query (Listing 5) contains a type check for preparation. It also uses a filter that binds an uncapitalized string of $name$ to the searched string $build$, which means stirring in the glass that is used for drinking.

```
SELECT ?preperation
WHERE { ?preperation <rdfs:Literal> ?name .
  ?preperation <rdf:type> <cocktail://cocktail/preparation>
  FILTER ( lcase(str(?name)) = "build" )
}
```

Listing 5: Query of preparation written in SPARQL

The categories (Figure 4.20) are designed as ingredients. The name and the type are checked, but there are no parent categories. $Build$ differs from $stir$ only in a practical way, which is why it is represented in the ontology only as a synonym of $stir$. Synonyms are realized with two literals that are connected to the same property.

In this case, the result of the path of $build$ is a list with a single item (Equation 4.4).

$$path_P(build) = Preparation(\text{cocktail://preparation/stir}) :: Nil \tag{4.4}$$

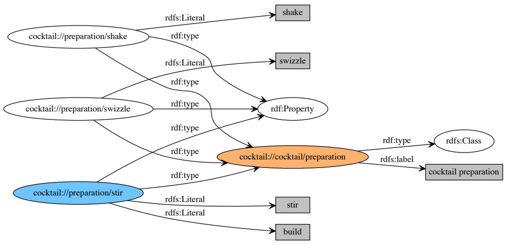


Figure 4.20.: Categories of preparation

4.2.3. Glassware

Glassware is highly diverse because of a huge collection of existing products; there are many names for the same glass or a very similar one. If the name is ignored and only the figure is considered, a glass can be classified into a small number of figures. This could be done automatically, but in this experiment, it is done manually.

Glassware is separated into classes of bottles or drinking glasses, which are realized in RDF as subclasses of glassware. The drinking glasses are manually classified into a small number of raw figures, namely *highballs*, *tumblers*, *ballon glass*, *goblets*, or *cocktail glasses* (little bowls). They are represented as properties with the type of *drinking glass*. Names, such as *julep cup* or *silver cup*, are recognized as synonyms (Figure 4.21). The hierarchy of glassware in the ontology is tendentiously flat, but examples like the *julep cup* have the same figure but are not of the same material. The *julep cup* is a special kind of *whiskey tumbler*, but is made of silver.

```
SELECT  ?type0 ?kindof0 ?type1 ?kindof1
WHERE {
  ?kindof0 <rdfs:Literal> ?name .
  ?kindof0 <rdf:type> ?type0 .
  ?type0 <rdfs:subClassOf> "cocktail://glassware"
  FILTER ( lcase(str(?name)) = "julep cup" )
  OPTIONAL {
    ?kindof0 <c:kindof> ?kindof1 .
    ?kindof1 <rdf:type> ?type1 .
    ?type1 <rdfs:subClassOf> "cocktail://glassware"
  }
}
```

Listing 6: Query of glassware written in SPARQL

The query (Listing 6) also has a type check. The type has to be a subclass of the glassware. The uncapitalized exemplary string of *julep cup* has to be the name. Only one kind of triple is allowed as an option. This also has to be a subclass of glassware.

The result (Equation 4.5) is a path with a size of 2. It first contains the silver cup and then the whiskey tumbler.

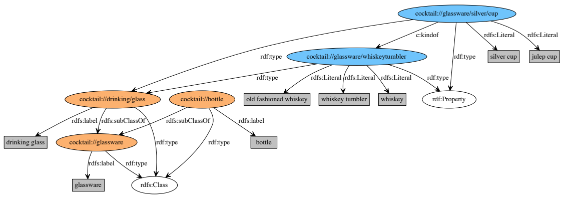


Figure 4.21.: Categories of glassware

$$
\begin{aligned}
path_G(julep\ cup) = \\
DrinkingGlass(\text{cocktail://glassware/silver/cup}) :: \\
DrinkingGlass(\text{cocktail://glassware/whiskeytumbler}) :: Nil
\end{aligned}
\tag{4.5}
$$

4.2.4. Experiences of extraction

The extraction into the data structure (Equation 4.1) has to simplify the recipes in terms of vocabulary, data structure, and knowledge:

Recipes contain anecdotes and explanations of preparation, which are not always necessary to make a cocktail. In some cases, such as when using a $Crusta$ or a $Julep$, which are more complicated, knowing these comments may prove useful. Nevertheless, it is not dependent on distances between recipes (see survey). Recipes comprise ingredients along with their origins. The ontology contains categories such as $Jamaica\ Rum$ or $Demerara\ Rum$, but if an origin is not known, the ingredient will not be recognized, because the ontology knows only the whole name. The ontology is maintained with multiple spellings.

Recipes contain many different spellings, as well as singular and plural words (Equation 4.6). These spellings are persistent in the ontology as alternative synonyms. If the spellings differs in clause position a rule is needed to convert the spellings (Equation 4.7). All are represented as synonyms to mean that these are the same.

$$
\begin{aligned}
wine - glass \rightarrow wineglass \\
wine\ glass \rightarrow wineglass \\
dashes \rightarrow dash
\end{aligned}
\tag{4.6}
$$

$$
one\ slice\ of\ lemon \rightarrow lemon\ slice \tag{4.7}
$$

Recipes contain the known default names of ingredients. Since recipes need to be short, ingredient names are as short as possible. The problem is that the names are not distinct. $Chartreuse$ is a company, but usually the product $Chartreuse\ Verte$ is meant. The $vermouth$ is a category, but $red\ vermouth$ is meant; therefore, $vermouth$ is a $superordinate$ to prevent that this is matched and $vermouth$ is added to $red\ vermouth$ as a synonym. The ontology is maintained to identify these cases.

Recipes contain numbers and fractions as words (Equation 4.8). It needs synonyms of numbers or fractions in the ontology. These are manually converted into digits.

$$\begin{aligned} one &\rightarrow 1 \\ 1/3 &\rightarrow \frac{1}{3} \\ half &\rightarrow 0.5 \\ one\ third &\rightarrow \frac{1}{3} \end{aligned} \tag{4.8}$$

Recipes contain ranges or quantities (Equation 4.9). It often means seasoning an ingredient. The average was chosen to be compatible with the given format.

$$2 - 4\ dashes\ bitters \tag{4.9}$$

Recipes also contain fillers (Equation 4.10), which are ingredients without a concrete quantity. However, that does not mean a $dash$ or a $splash$, which is always a small quantity. A filler could be about $10\ cl$. The concrete quantity chosen must be realistic in terms of the glassware.

$$fill\ with\ soda \tag{4.10}$$

Recipes in historic books contain or-relations (Equation 4.11). For example, either bourbon or rye has to be used, not both. Recipes also contain optional ingredients. The target data structure supports only one.

$$\begin{gathered} 3\ ounce\ bourbon\ or\ rye \\ optionally\ 1\ dash\ Angostura \end{gathered} \tag{4.11}$$

Recipes contain solid ingredients (Equation 4.12). The mapping of solids to liquids allows one to find better similarities with other recipes. Converting the measurements is not enough, because it is necessary to combine a qualitative unit such as $half$ with an ingredient such as

$lemon$. The ontology has to know that one $lemon$ contains about 5 cl, in order to convert this correctly. The conversion of the ingredient to liquid was done manually.

$$\begin{aligned} half\ small\ lemon &\rightarrow 2.5\ cl\ lemon\ juice \\ piece\ orange &\rightarrow 1\ cl\ orange\ juice \\ 5\ cl\ lemon\ lemonade &\rightarrow 4\ cl\ soda, \\ &\quad 0.5\ cl\ lemon\ juice, 0.5\ cl\ sugar \end{aligned} \tag{4.12}$$

Quantities are implicit if they are usual (Egg is shortened form of one piece of egg). Items of preparation such as stir or shake, drinking glass, preparation glass, or preferred ice contain many recipes, but every type of item could be missing.

Recognition and support of these issues make it possible to map a recipe more precisely. For an experiment, cocktail recipes are necessary, which are used in real life. If they are too simplified, the extracted recipe is not according to the expectation of domain experts. These are requirements for modeling in the next section. Preparation and glassware are not considered for recommendation, therefore these features are not needed in the feature vector, but for human readable format these information are provided and duplicates are removed.

Ingredients are known by names. If a name is a universal one, which is contained in dictionaries or is a public brand, the ingredient is understandable by every domain expert. If it is a very special name, a recipe for the ingredient is necessary. For this approach, ingredients are assumed to have universal names. It is also assumed that the recipes are thoroughly mixed. Zests or cherries are excepted, but recipes of molecular mixology, which result from different aggregate phases such as foams (Espuma), are not considered.

4.2.5. Target structure

The target structure is the result of manual extraction by a domain expert and describes one cocktail recipe. A flexible structure is required to extract different styles of cocktail recipes.

The extracted features represent the internal representation (Equation 4.13). It is a technical presentation that is necessary for the recommendation.

$$trait\ Item\{\ val\ uri : String\ \} \tag{4.13}$$

The URI guarantees unique identification. Different spellings, which are extracted to the same identifier, could be interpreted as the same. The user needs to understand and classify the extra information attached to the recipe such as the name, the original spelling of an assignment, and meta-information about the book and the author. The representation, which contains information for the user, is the external representation (Equation 4.14). The result is one data structure that represents the internal and external data.

$$trait\ ValueItem\{val\ i : Item,\ val\ name : String\ \} \tag{4.14}$$

The assignment list (Equation 4.15) contains a sequence of items and a quantity. The sequence shows that only one has to be chosen. This sequence is defined as a $or - relation$ of items. Allowed items are touchable such as ingredients, glassware, or ice. Preparations cannot be an assignment.

$$Assignment(items : Seq[ValueItem[Item]], quantity : Quantity) \tag{4.15}$$

The quantity contains a unit and a value. $NumVal$ is an $Item$ that also contains a numeric value. The $NumVal$ could also be a $NumRange$, which contains a minimum and a maximum value.

$$\begin{aligned} trait\ Quantity\{ \\ val\ numVal : ValueItem[NumVal] \\ val\ unit : ValueItem[UnitItem]\ \} \end{aligned} \tag{4.16}$$

The publication contains meta-information and could be a book or a collection. Some books mention the source of a recipe. Additionally, a collection contains the original book.

$$\begin{aligned} Publication\ extends\ Item\{ \\ val\ name : String \\ val\ author : String \\ val\ published : Int\ \} \end{aligned} \tag{4.17}$$

The cocktail data structure (Equation 4.18) combined all information about a cocktail. A cocktail needs a name, but all other values are optional. Preparation, glassware and ice are subtypes of item, which represents one taxonomy in the ontology.

$$\begin{aligned} Cocktail(name : String, \\ assignments : Seq[Assignment], \\ preparations : Seq[ValueItem[Preparation]], \\ glassware : Seq[ValueItem[Glassware]], \\ ice : Seq[ValueItem[Ice]], \\ publication : Option[Publication]) \end{aligned} \tag{4.18}$$

Summary

The target structure is the formal description of a cocktail for a recommendation, which is the basis to extract recipes. The ontology store the necessary items for classification and distance measurement. The understanding of the domain is necessary for both. The target structure coupled together with the ontology represent the domain-specific knowledge (challenge one) to extract recipes.

4.3. Parsing of cocktail books

For dealing with huge volumes of data (challenge two) this experiment involves the parsing of cocktail books. Seven books are chosen, which are used in a PDF format with the underlying text[2]. This is added by OCR techniques. Written in English, these books focus on cocktail recipes.

Each book contains special characteristics. Recipes in the “Cafe Royal” and “Approved Cocktail” are reduced to the most important information. Most quantities are relative. Recipes from Jerry Thomas’s “How to Mix Drinks” and Harry Johnson’s “Bartender’s Manual” contain long descriptions and different units.

- 1862 Jerry Thomas — How to Mix Drinks: The Bon-Vivants Companion (New York, USA)
- 1882 Harry Johnson — Bartender’s Manual (Chicago and New York, USA)

[2] Previous version is located Sections 1–6 in https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2015-proj/sippel2.pdf

- 1884 George Winter — How to Mix Drinks (New York, USA)
- 1917 Tom Bullock — The Ideal Bartender (St. Louis, Illinois, USA)
- 1936 Frank Meier — The Artistry of Mixing Drinks (Paris, France)
- 1937 William J. Tarling — Approved Cocktails (London, England)
- 1937 William J. Tarling — Cafe Royal (London, England)

The redesigned target structure is used to derive recipes from authentic cocktail books automatically. The books contain only raw text, which is why a parsing process is used to recognize features. The parsing process is developed for this domain explicitly because domain-specific assumptions are needed which are not implemented in common libraries. An assignment of an ingredient is not a complete sentence, therefore the quantity will not be recognized. The units such as $wine - glass$ are not expected usually. Entities have to be classified in context of domain.

The parsing process contains a preprocessing phase and several phases of feature extraction, which are used to identify tokens and refine them. A phase is defined as a list of tokens processed with a set of rules. The main phases are illustrated as red nodes in Figure 4.22. The first is preprocessing phase which tokenize the raw text and removes unwanted characters. It follows the recognition phase, which is used to classify raw tokens into typed items. The cleaning phase removes unnecessary items or changes the order of items. The context phase recognizes assignments. In the selection phase, items are chosen that have been used as features. The reasoning phase makes selected features comparable.

The output is a collection of workable recipes. These recipes are prepared for the recommendation that uses one recipe from the collection as an example to find other recipes in the collection that are classified as a recommendation.

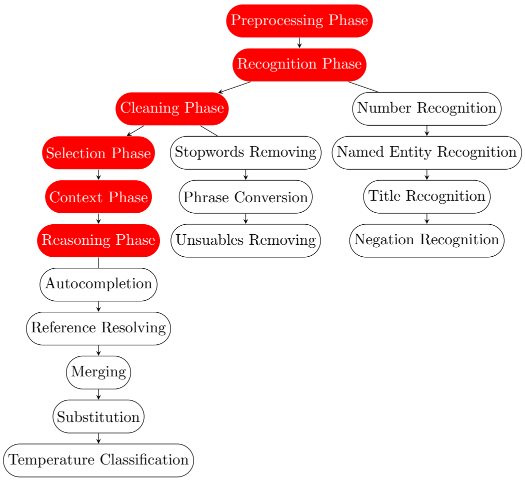


Figure 4.22.: Phases of cocktail parsing

4.3.1. Recognition phase

Preprocessing

The sources of the recipes are books written in English. The text have special characters that either have a special impact — called key characters (Equation 4.19) — or have to be ignored — called killing characters (Equation 4.20). The books are converted into raw text with optical character recognition (OCR); therefore, a special character could also show an error. In the first instance, the words and special characters have to be separated into single tokens because these characters have to be processed in particular. If a special character is combined with a word, neither can be recognized.

$$/ - \% \tag{4.19}$$

$$()| * - \backslash "\$\%\wedge\{\}, : .\S \tag{4.20}$$

For normalization, every member of the defined set of key characters (Equation 4.19) will be replaced within itself with additional white spaces (Equation 4.21). The killing characters are removed.

$$character \rightarrow whitespace\ character\ whitespace \tag{4.21}$$

The result comprises words, and the special characters are separated by white spaces. A splitting by white spaces results in a list of raw tokens. New lines explicitly act as tokens, because it is valuable information for separating the token before and after.

Number recognition

Numbers are recognized in three phases. The target numeric values are divided into several types (Figure 4.23). These types are needed for precise rules in further phases.

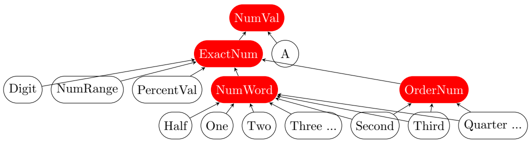


Figure 4.23.: Numeric value types (red nodes are traits; white nodes are concrete classes or objects)

At first, all items that are numbers or describe numbers are mapped to numeric items (Equation 4.22). A slash could be a part of a fraction; therefore, it is mapped to a $slash\ item$. The words *or* and *to* could also be a part of a range. *or* connects also two ingredient names, both of which are mapped to an $items$. The symbol $\%$ describes a number with a percent value; therefore, it is mapped to a $percent\ item$. Since a hyphen could be a part of a numeric range, such as $1-2$, it is mapped to a $hyphen\ item$. Digits are mapped to a $digit$ item. Written-out numbers such as $three$ are mapped to a $numeric\ word$. Ordering numbers, such as $third$, are mapped to a $ordered\ numeric\ word$.

$$\begin{aligned} or &\rightarrow Or \\ to &\rightarrow To \\ / &\rightarrow Slash \\ - &\rightarrow Hyphen \end{aligned} \tag{4.22}$$

Indefinite articles, indefinite pronouns, and qualitative declarations (Equation 4.23) are mapped to a special numeric value of a $value\ item$ of type A. These words do not mean the same thing, but are imprecise. Items such as $a\ lemon$ could also be a $small$ or a big one. Moreover, $big\ lemon$ is imprecise because the size of a big lemon is not specified. The main information is that only one lemon should be used. It must be seasoned, but this cannot be mapped to a value.

$$a, an, some, any, small, big \tag{4.23}$$

In the second phase, fractions and ranges are mapped to one numeric value.

There are special characters, such as $\frac{1}{2}$ or $one - third$, but in the used OCR results, there are only numbers combined with slashes. The rule (Equation 4.24) maps to numbers combined with a slash to one $numeric\,value$.

$$\begin{aligned} NumVal(n1) :: Slash :: NumVal(n2) &\rightarrow NumVal(\frac{n1}{n2}) \\ NumWord(n1) :: Hyphen :: OrderNum(n2) &\rightarrow NumVal(\frac{n1}{n2}) \\ NumWord(n1) :: OrderNum(n2) &\rightarrow NumVal(\frac{n1}{n2}) \end{aligned} \tag{4.24}$$

Numbers are used as ranges (Equation 4.25). Recognized numbers will be mapped to a range if an or item, to item, or a $hyphen$ item connect two numbers.

$$\begin{aligned} NumVal(n1) :: Or :: NumVal(n2) &\rightarrow NumRange(n1, n2) \\ NumVal(n1) :: To :: NumVal(n2) &\rightarrow NumRange(n1, n2) \\ NumVal(n1) :: Hyphen :: NumVal(n2) &\rightarrow NumRange(n1, n2) \end{aligned} \tag{4.25}$$

The third phase contains the final rules for number recognition:

One number with a following percent item is mapped to the representation as a fraction.

$$NumVal(n) :: Percent \rightarrow PercentVal(\frac{1}{100}) \tag{4.26}$$

If the recipe contains sentences or half-sentences (Equation 4.27), the number recognition needs additional rules because there are more than two number values, but only one of them is interesting.

$$\begin{aligned} use\ a\ half\ of\ a\ lemon \\ a\ big\ lemon \\ one\ of\ a\ big\ lemon \end{aligned} \tag{4.27}$$

The word use is a stop word, but there is more than one item of type $NumVal$. The type A is a $NumVal$ too. The following rule set (Equation 4.28) reduces the items to one item.

$$\begin{aligned} NumVal(n) :: A :: A &\rightarrow NumVal(n) \\ A :: NumVal(n) :: A &\rightarrow NumVal(n) \\ A :: A :: NumVal(n) &\rightarrow NumVal(n) \\ A :: NumVal(n) &\rightarrow NumVal(n) \\ NumVal(n) :: A &\rightarrow NumVal(n) \end{aligned} \tag{4.28}$$

The extraction of a combined number (Equation 4.27) is applied in the Equation 4.29.

$$\begin{aligned} A :: NumVal(0.5) :: A &\rightarrow NumVal(0.5) \\ NumVal(1) :: A &\rightarrow NumVal(1) \\ NumVal(1) :: A :: A &\rightarrow NumVal(1) \end{aligned} \tag{4.29}$$

A fraction or a range (Equation 4.30) has to be combined with other combinations; therefore, the fraction recognition has to be completed before the other rules can work. For example, the fraction is mapped to one numeric value (Equation 4.30) and only then is a number reduction possible.

$$1/2 \; of \; a \; lemon \rightarrow 0.5 \; lemon \tag{4.30}$$

Named entity

The raw set of tokens contains information, such as ingredient names or units, but these are yet to be recognized. The ontology is the knowledge. If the ontology contains a literal, the literal is recognizable. Initially, the empty string is ignored and a defined set of keywords are mapped to typed items (Equation 4.31).

$$\begin{aligned} \backslash n &\rightarrow HardSeparator \\ ; &\rightarrow HardSeparator \\ .\ whitespace &\rightarrow HardSeparator \\ not &\rightarrow Not \\ of &\rightarrow Of \end{aligned} \tag{4.31}$$

If no previous rule is matched, the token is looked up in the ontology to find named entities such as ingredients, which are classified in different taxonomies (Figure 4.24).

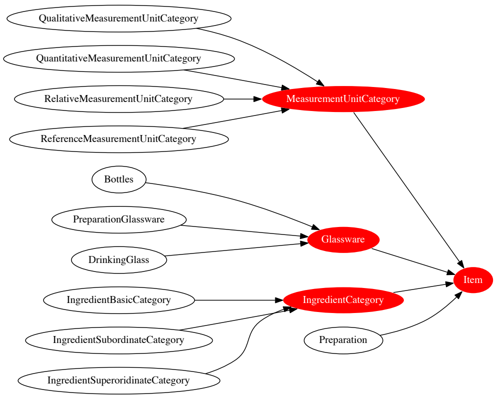

QualitativeMeasurementUnitCategory
QuantitativeMeasurementUnitCategory
RelativeMeasurementUnitCategory
ReferenceMeasurementUnitCategory
MeasurementUnitCategory
Bottles
PreparationGlassware
DrinkingGlass
Glassware
Item
IngredientBasicCategory
IngredientCategory
IngredientSubordinateCategory
Preparation
IngredientSuperoridinateCategory

Figure 4.24.: Hierarchy of item types; red types are traits

A Lucene index is used to find a literal or a part of a literal with high performance. A small ontology (Listing 7) is used to declare a Lucene index called $text$ with the key uri and the value $text$. The URI represents the URI of a property and the text declares a literal. The literal is also connected to the $rdf : type$ to identify the taxonomy.

```
<#indexLucene> a text:TextIndexLucene ;
 text:directory "mem" ;
 text:entityMap <#entMap> ;
 .
<#entMap> a text:EntityMap ;
 text:entityField      "uri" ;
 text:Field      "text" ;
 text:map (
  [ text:field "text" ; text:predicate rdfs:Literal ;
  text:predicate rdf:type ]
 ) .
```

Listing 7: Mapping of index to ontology fields, written in Turtle

The SPARQL query (Listing 8) to find named entities uses $text$, which is the previously defined Lucene query, and a chosen literal, such as $Strawberries$. As RDF has no type hierarchy, the transitivity is explicitly declared. A bottle is a subclass of glassware, and the bottle and glassware are subclasses of items. All superordinates are allowed in the entity recognition.

The found raw token could contain lexical errors or represent only a part of the appropriate literal. Therefore, the result is sorted by ascending Levenshtein distance. The item with the

```
SELECT ?uri ?label ?type {
?uri text:query (rdfs:Literal "Strawberries") ;
    rdfs:Literal ?label .
      ?uri rdf:type ?type .
      ?type rdfs:subClassOf "cocktail://item"
}
```

Listing 8: Named entity recognition written in SPARQL

smallest Levenshtein distance is used, if the acceptance criteria (Equation 4.32) are complied with.

$$Levenshtein(lowerCase(s1), lowerCase(s2)) \leq 2 \tag{4.32}$$

The found URI of type represents the type of item.

$$cocktail : //preparation/cocktail \rightarrow Preparation \tag{4.33}$$

If the acceptance criteria are not complied with, and the next token will be concatenated with a white space in between. The new string is evaluated with the named entity query. The tokens are concatenated until the acceptance criteria are complied with or all the tokens are concatenated. In this case, the first token is declared as an unknown item. If a result of a named entity query is not accepted, a spelling error is possible. Therefore, a Lucene-based spell checker with a dictionary of Project Scowl[3] is used to recommend the corrected word. The named entity query is repeated with the corrected word. Words with the right count of characters but some wrong characters are suited to correct automatically (Equation 4.34).

$$tablesp{\sim}{\sim}n \rightarrow tablespoon \tag{4.34}$$

Entities such as $lemon\ zest$ at first contain $lemon$, which is also an entity. If the token $zest$ is after $lemon$, then lemon is already recognized. Some entities are also used with a comma. $rabsperry\ syrup$ or $syrup,\ raspberry$, which is the same thing. For such cases, the combinations of the original values of recognized entities are concatenated as a string. This string is evaluated as one named entity. If the acceptance criteria is complied with, the items are merged. The combinations contain two or three items.

It result three phases for named entity recognitions. The first map strings to entities by first match of concatenated string. The second searches for existing combinations, while the third searches for reverse combinations.

[3] http://wordlist.aspell.net/

Title recognition

The title is defined by the first line of a recipe to get a simple recognizable recognition rule. A title has to be a part of a recipe. If a recipe contains no separation items, a separation item is added to a line before the first known item. In the following example without a separation, the first two items are recognized as title items. This is a fallback mechanism. In usual cases, there is a separation after the title.

> Manhattan Sweet 1 part Italian Vermouth 2 part Whisky 1 dash Angostura 1 piece Maraschino cherry (stir, cocktail glass)

Negation recognition

There are recipes that use phrases such as $do\ not\ shake$. Shake is a method of preparation. There are only two different preparations — stir or shake. In this case, it should be stirred. Therefore, rules (Equation 4.35) are needed to convert the negation. If this conversion is not done, a false positive preparation will be recognized.

$$\begin{aligned} Not :: Shake &\rightarrow NotShake \\ Not :: Stir &\rightarrow NotStir \end{aligned} \tag{4.35}$$

4.3.2. Cleaning phase

The aim of the cleaning phase is to get items in the right order (Equation 4.36). Numeric values or units are optional, but the order has to be right to find assignments in the context phase.

$$num \rightarrow unit \rightarrow (ingredients \mid glassware \mid ice \mid unknown) \rightarrow separation \tag{4.36}$$

Removing stop words

A defined set of stop words, such as use or in, are removed because these are not required for feature extraction. This list is updated when an unnecessary item is found. All unknown items that have a value contained in the stop word list are removed. After the removal, the stop words are invisible. If a stop word is important for parsing, the matching rule has to be processed before. Therefore, the recognition phase is performed before the stop words are removed. For example $fruits\ in\ season$ is recognized by recognition phase as one ingredient item, therefore it will not be removed, but a unknown item in will be removed.

Phrase conversion

Phrase conversion is a powerful phase because unnecessary named entities and numbers of a phrase are removed. Phrases are recognized and converted to right order.

Phrases containing the word of indicate a wrong order of items. In the following rules are used the item trait of probably ingredients which is morphed of unknown or ingredient items. If numeric value (Equation 4.39) or unit (Equation 4.40) and probably an ingredient connected by of, the item of type of is removed only. Phrases $such\ as\ white\ of\ (an)\ egg$ (Equation 4.37) or $juice\ of\ a\ lemon$ (Equation 4.38) needs to change positions. Additional rules are needed to understand these phrases, along with numeric values such as A or $Half$. Similar phrases contain glassware or ice.

$$egg :: Of :: ProbablyIngredient \rightarrow ProbablyIngredient :: zest \tag{4.37}$$

$$juice :: Of :: ProbablyIngredient \rightarrow ProbablyIngredient :: juice \tag{4.38}$$

$$NumVal :: Of :: ProbablyIngredient \rightarrow NumVal :: ProbablyIngredient \tag{4.39}$$

$$\begin{aligned} MeasurementUnit :: Of :: ProbablyIngredient \\ \rightarrow MeasurementUnit :: ProbablyIngredient \end{aligned} \tag{4.40}$$

Qualitative units, in combination with glassware or ice, have to be reduced to get only tokens, which are in the right order. No duplicates of units are allowed and no preparation of phrases, because they will be filtered in the selection phase. If it is in a phrase, the phrase will be broken after filtering.

$$fill :: shaker \rightarrow shaker \tag{4.41}$$

$$fill :: glass :: ice \rightarrow fill :: ice \tag{4.42}$$

$$fill :: shaker :: ice \rightarrow fill :: ice \tag{4.43}$$

$$ProbablyIngredient :: on :: top \rightarrow top :: ProbablyIngredient \tag{4.44}$$

Removement of unusable items

The last phase of cleaning involves removing all senseless items. Phrases such as $two\ seconds$ (Equation 4.45) and $(recipe)\ No.\ 10$ are explicitly identified and removed because they cannot be used as assignments. A separation between two assignments is expected, but if the recipe contains more complete sentences, a separation could be missing. Therefore, if a numeric value follows a standalone item (ingredient, glassware, unknown), a separation is added

(Equation 4.47). Before a separation, only a standalone item is allowed, which is why numeric values, units, and keywords (to, on, not, of, instead, slash) are removed (Equation 4.48). Hyphens, which are not used between two recognized words or numbers, are also removed (Equation 4.49).

$$\begin{align}
NumVal :: Time &\rightarrow \tag{4.45}\\
Not :: NumVal &\rightarrow \tag{4.46}\\
Standalone :: NumVal \rightarrow Standalone :: HardSeparator :: NumVal & \tag{4.47}\\
Or :: HardSep. \rightarrow HardSeparator & \tag{4.48}\\
NumVal :: HardSeparator \rightarrow HardSeparator & \notag\\
MeasurementUnitCategory :: HardSeparator \rightarrow HardSeparator & \notag\\
Keyword :: HardSeparator \rightarrow HardSeparator & \notag\\
Hyphen &\rightarrow \tag{4.49}
\end{align}$$

4.3.3. Selection phase

Due to the logic of combining, the processing of preparations and glassware items can be identified by the type of the item. The preparations and ice items can be filtered easily and interpreted as a list that describes an or-relation. The glassware is handled in a similar way, but this list is separated in three additional parts (bottle, preparation glassware, and drinking glasses), because these parts are independent (Equation 4.50).

$$(shake,\ cocktail\ glass, large\ bar\ glass,\ shaved\, ice) \tag{4.50}$$

Glassware and ice are also supported by the next phase in order to recognize a phrase such as $fill\ glass$. Therefore, these items are not removed from the list. Preparation items are not supported, hence their removal from the item list.

4.3.4. Context phase

The context analysis is needed to find assignments with a number, a unit, ingredients, glassware, or ice. For the context analysis, the recipe is considered an example of a domain-specific language, which has to be described according to grammar. For context analysis, the Scala parser combinators are used. This library parses a string using EBNF grammar.

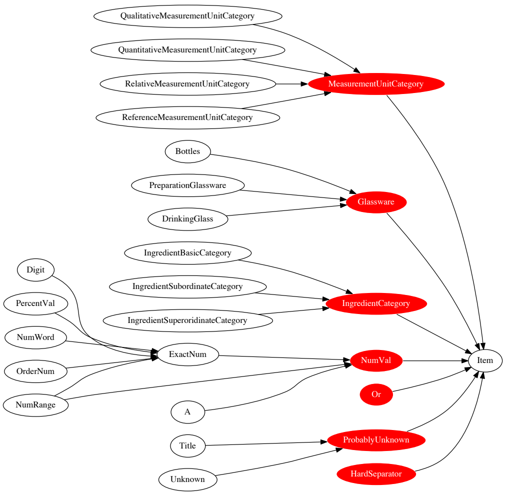


Figure 4.25.: Type hierarchy of items: Red types used for context analysis

The items are represented in a list of typed items. Therefore, the list is serialized. Every item is serialized with an identifier of type and an index of the list to a string-based notation (Equation 4.51). With this information, the original item can be found and the type can be used for parsing.

$$@\, typeidentifier\ index \tag{4.51}$$

The relevant types are mapped to a string notation. The type hierarchy of items (Figure 4.25) shows the relevant subset of types for the context analysis. The red ones are mapped to the serialized format. Subtypes of the red ones are transparent for the context analysis.

$$\begin{aligned} Or &\rightarrow @O \\ NumVal &\rightarrow @N \\ ProbablyUnknown &\rightarrow @U \\ MeasurementUnitCategory &\rightarrow @M \\ IngredientCategory &\rightarrow @I \\ Glassware &\rightarrow @G \\ Ice &\rightarrow @I \\ HardSeparator &\rightarrow @S \end{aligned} \tag{4.52}$$

The red types (Figure 4.25) are mapped in a specific order (Equation 4.52). The item Or is important for the context analysis and will be mapped to itself. Separators are used to separate two ingredients from each other.

The parser is defined by EBNF grammar. The notation is adapted to enable writing it directly in Scala: the concatenation is represented by a tilde. If a rule matches, a type transformation is possible and is written as two circumflexes. At first (Listing 9), the index id and all identifiers are declared as tokens. It is only important to find the identifier; therefore, the mapping is always to the unit type (written as ()).

```
def id              = """(\w+)""".r ^^ { _.toString }
def unknownId       = """(@U)""".r  ^^ { _ => () }
def measurementId   = """(@M)""".r  ^^ { _ => () }
def ingredientId    = """(@I)""".r  ^^ { _ => () }
def glasswareId     = """(@G)""".r  ^^ { _ => () }
def preparationId   = """(@P)""".r  ^^ { _ => () }
def numId           = """(@N)""".r  ^^ { _ => () }
def separatorId     = """(@S)""".r  ^^ { _ => () }
def hyphenId        = """(@Y)""".r  ^^ { _ => () }
def orId            = """(@O)""".r  ^^ { _ => () }
```

Listing 9: Recognition of internal identifiers

The identifier with index is mapped to the original item (Listing 10). The items can be identified with the declared rules.

```
def separator = separatorId     ~ id ^^ { case _ ~ i => item(i) }
def unknown = unknownId         ~ id ^^ { case _ ~ i => item(i) }
def measurement = measurementId ~ id ^^ { case _ ~ i => item(i) }
def ingredient = ingredientId   ~ id ^^ { case _ ~ i => item(i) }
def glassware  = glasswareId    ~ id ^^ { case _ ~ i => item(i) }
def preparation = preparationId ~ id ^^ { case _ ~ i => item(i) }
def num = numId                 ~ id ^^ { case _ ~ i => item(i) }
def or = orId                   ~ id ^^ { case _ ~ i => item(i) }
```

Listing 10: Mapping of identifier with ID of original items

An assignment is a sequence of items (Listing 11). There are four types of them. The first type contains only one item name (such as $lemon\ zest$), without any unit or numeric value. The second type is a number with an item name. The number is interpreted as part of the quantity of the item. The third contains a unit as well as an item name, such as $dash\ bitters$. The number is missing. This case is realistic, but it could be an error because the number was not recognized. The last type contains a number, a unit, and an item name, which is perfect because all the information is found.

Recipes could contain multiple item names in an assignment. Multiple items could be $or - relations$ or $and - relations$ (Listing 12). The $and - relation$ describes a list of tokens that do not contain the word or, a $number$, or a $unit$. The result is a list of names that are probably known. Since an item could be missing from the ontology, the parser would have

```
def a1 = ingredient                    ^^ { case         n => A1(List(n))}
def a2 = num ~ ingredient              ^^ { case q     ~ n => A2(q,List(n))}
def a3 = measurement ~ ingredient      ^^ { case     u ~ n => A3(u,List(n))}
def a4 = num ~ measurement ~ ingredient ^^ { case q ~ u ~ n => A4(q,u,List(n))}
```

Listing 11: Rules for assignments

to work with unknown items, if possible. The list contains more than one name. The list is mapped to the first known item with fall back to an unknown item. The $or - relation$ describes a list of ingredients or unknown words, which are separated by the word or. This rule identifies ingredients such as $rum\ or\ gin$. The mapping filters the word or, while the result list contains only names. The list needs a single element.

```
def ma = andList | orList
def andList = (ingredient | unknown) ~ ((ingredient | unknown)+) ^^
  { case f ~ l => chooseFirstKnownItem(f::l) }
def orList = (ingredient | unknown) ~ (orTail?) ^^
  {case a1 ~ a2 => a1 :: a2.getOrElse(Nil)}
def orTail  = ((or ~ (ingredient | unknown))+) ^^
  { case x => x.map(x => x._2)}
```

Listing 12: Rules for multiple ingredient names

These rules of assignments are extended with multiple item names (Listing 13) that are described by the rule ma. In this case, unknown names are allowed. This is possible if only one line is parsed. Therefore, a newline separator has to be consumed.

```
def a1m = ma ~ separator ^^                      { case         n ~ _ => A1(n)}
def a2m = num ~ ma ~ separator ^^                { case q     ~ n ~ _ => A2(q,n)}
def a3m = measurement ~ ma ~ separator ^^        { case     u ~ n ~ _ => A3(u,n)}
def a4m = num ~ measurement ~ ma ~ separator ^^  { case q ~ u ~ n ~ _ => A4(q,u,n)}
```

Listing 13: Rules for assignments with multiple ingredient names

All rules of assignments are combined in the rule a (Listing 14). The rules are ordered according to the numbers of necessary tokens. A recipe contains a list of assignments, which is described by the rule al. Between two assignments, it is possible to consume any separators or unknown items. This rule helps to parse recipes with errors or missing items in the ontology.

The cocktail name (Listing 15) lists items that do not contain a unit. The cocktail rule includes the name and the assignment list.

Some recipes contain preparation information in a phrase after the cocktail name. The recognized items are removed, but not all items contain this phrase. For example, there is an additional separator. Other recipes contain information about the author after the name (Listing 16). Therefore, any separators and unknown items can be consumed between the

```
  def a = a4m | a4 | a3m | a3 | a2m | a2 | a1m | a1
  def al = a ~ ((unknown | separator)*) ~ al  ^^
  { case h ~ _ ~ t => h :: t   } | success(List())
```

Listing 14: Rules for list of ingredients

```
def name = (num | unknown | ingredient) ~ name  ^^
  { case h ~ t => h :: t  } | success(List())
def cocktail  = name ~ ((separator | unknown)*) ~ al ~ (unknown*) ^^
  { case  n ~ _ ~ in ~ _ => C(n,in)}
```

Listing 15: Rules for a cocktail recipe

cocktail name and the ingredient list. The parser parses assignments as long as it works, because long descriptions after the assignment list should not stop the parsing workflow.

```
ROBERTA
Invented by
G. Newman
Juice of 1/2 a small lime.
1/3 maraschino.
2/3 daiquiri rum.
Shake.
```

Listing 16: Recipe with author name

If all items are recognized, then the parsing process is simple. In this case, the requirements of the recipe are the lowest. In the following recipes, there are only known items. For all assignments, Rule $a4$ is used because a number, a unit, and a name are always found. This recipe does not need a separator to be parsed successfully.

> Manhattan Sweet 1 part Italian vermouth 2 part whisky 1 dash Angostura 1 piece maraschino cherry (stir, cocktail glass)

As recipes are different, not all items can be recognized. This parser is designed to parse recipes in the worst-case scenario. The worst case of a parsable recipe occurs when an item sequence is found that contains a classifiable one and an unknown one in an alternate manner (Equation 4.53). A classifiable item is an item that matches the context.

$$classifiable\ unknown\ classifiable\ unknown\ classifiable.... \tag{4.53}$$

If a measurement unit is found, and if three unknown items and a separator follow it, then the conclusion is acceptable to map the unknown items to one ingredient. If the first assignment contains only an ingredient name, such as *orange zest*, then a separator between the cocktail name and the first assignment is necessary to find where the cocktail name ends. If there are two assignments comprising only one name, such as $orange\ zest$ and egg, then a separator is also needed to conclude that there are two assignments.

4.3.5. Domain-specific reasoning phase

Quantity autocompletion

The assignments are mapped to the types $A1$, $A2$, $A3$, and $A4$. The type $A4$ contains all information that is necessary for the distance function — a number, a measurement unit, and one or more ingredient names. The other types have to be converted into $A4$. The missing information is retrieved by default values, which are stored in the ontology. A category such as a $bitter$ is comprised of a default quantity with a unit and a volume (Listing 17).

```
<c:defaultQuantity c:unit="cocktail://unit/dash" c:volume="1" />
```

Listing 17: Declaration of a default quantity in RDF

The SPARQL query (Listing 18) for the default quantity needs the URI of the ingredient, the volume, and the unit to be bounded for a successful result.

```
SELECT DISTINCT ?type ?kindof ?volume ?unit WHERE {
 ?kindof rdf:defaultQuantity ?default .
 ?default rdf:volume ?volume .
 ?default rdf:unit ?unit
 FILTER (str(?kindof) = "cocktail://ingredient/bitters") }
```

Listing 18: Query of a default quantity written in SPARQL

If the query is unsuccessful, it is used as a default value (Equation 4.54). The most frequent case is that the number is not written if it is 1. The units for solid ingredients, such as $piece$ for the ingredient egg, are stored in the ontology. Thus, the value for fallback has to be a relative unit, such as $part$. It is possible that the measurement unit was not unrecognized, in which case this conclusion would be wrong.

$$Quantity(1, part) \tag{4.54}$$

Every assignment can contain more than one item name. In this case, all default values of these items have to be the same. Otherwise, the default value (Equation 4.54) is used. The assignments are mapped to $I4$ with default values (Equation 4.55).

$$\begin{aligned} I1(n) &\rightarrow I4(default(n), n) \\ I2(num, n) &\rightarrow I4(num, default(n).u, n) \\ I3(u, n) &\rightarrow I4(default(n).num, u, n) \\ I4 &\rightarrow I4 \end{aligned} \tag{4.55}$$

An example is the hot spiced rum recipe, which contains assignments with and without a measurement unit. The extracted features are within brackets.

HOT SPICED RUM[4]
1 or 2 lumps of sugar
4 teaspoonfuls allspice
(1) (part) water
1 (part) Jamaica rum
1 (prise) nutmeg

Referenced measurement unit resolution

Some recipes comprise a referenced unit, called $ditto$ or is short $'do.'$. This refers to the units that have been used in the assignment before.

Brandy Punch[5]
(Use large barglass)
1 table-spoonful raspberry syrnp
2 do. white sugar
1 wine-glass water
1 do. brandy
1/2 small-sized lemon
2 slices of orange
1 piece of pine-apple
Fill the tumbler with shaved ice

[4] 1882 Harry Johnson — Bartender's Manual

[5] 1862 Jerry Thomas - How to Mix Drinks - The Bon-Vivants Companion

shake well
and dress the top with berries in season
sip through a straw

The reference measurement resolution converts the filtered ingredient list of assignments into those with only qualitative, quantitative, or relative units (Equation 4.56).

$$\begin{aligned} Assignment(items1, Quantity(MeasurementUnitCategory, n1) ::\\ Assignment(items2, Quantity(RefrencedMeasurmentUnitCategory, n2) \rightarrow\\ Assignment(items1, Quantity(MeasurementUnitCategory, n1) ::\\ Assignment(items2, Quantity(MeasurementUnitCategory, n2) \end{aligned} \tag{4.56}$$

Ingredient merging

Ingredient items are often duplicated, because a recipe contains an assignment list but also a description text which contains referenced ingredients. For example the descriptions text offers further explanations or details to this ingredient.

Blue Blazer.
(Use two large silver-plated mugs, with handles.)
1 wine-glass of Scotch whiskey.
1 do. boiling water.
Put the whiskey and the boiling water in one mug,

The difference is that only one of them is in numeric values and units. After auto completion, it leads to two assignments with similar ingredient items, one full quantity, and an auto completed one. The aim is to get an assignment containing no auto completed information (Equation 4.57).

$$\begin{aligned} (1)\ (part)\ whiskey \rightarrow 1\ wine-glass\ Scotch\ whiskey\\ (1)\ (part)\ boiling\ water \rightarrow 1\ do.\ boiling\ water \end{aligned} \tag{4.57}$$

The rate of auto completion is classified into not auto completed, auto completed by ontology, and the value of Equation 4.54. The ontology is used to calculate the path of both items (Figure 4.38). If there is a common node in both paths (the parent of scotch is whiskey), the ingredients are marked to merge. The path length shows the rate of concrete terms. The longest path has the higher rate. Path resolution is described in the section on distance function. The quantity with less auto completed data and the item with the rate of concrete terms are used to replace. If all combinations of ingredient pairs are merged and similar ones are found, the ingredient list would contain duplicates. This is because they comprise both the original one and the merged one. These duplicates are removed.

Ingredient substitution

Recipes contain ingredients such as lemon or lemon juice, both referring to the same ingredient. It is not possible to consider both ingredients as synonyms, because the quantity is different. Therefore, ingredients should be used as far as possible in their liquid form, because these are comparable by volume. If a recipe requires one piece of lemon and if it is known that a lemon gives about 4 – 6 cl of juice (Equation 4.58), then the piece of lemon is substituted with lemon juice.

$$one\ (piece)\ lemon \rightarrow 4 - 6\ cl\ lemon\ juice \tag{4.58}$$

The substitution is stored in the ontology (Listing 19).

```
<c:liquidSubstitution c:sourceUnit="cocktail://unit/piece"
  c:targetUnit="cocktail://unit/cl"
  c:minVolume="4" c:maxVolume="6"
  c:substitute="cocktail://ingredient/juice/lemon" />
```

Listing 19: Query of substitution written in SPARQL

The query for resolving a substitution needs the URI of a unit and an ingredient item (Listing 20).

```
SELECT DISTINCT ?name ?type ?kindof ?minVolume ?maxVolume
 ?unit ?substitute WHERE {
   ?kindof rdfs:literal ?name .
   ?kindof rdf:type ?type .
   ?type rdfs:subClassOf "cocktail://ingredient" .
   ?kindof c:liquidSubstitution ?liquidSubstitution .
   ?liquidSubstitution c:sourceUnit ?sourceUnit .
   ?liquidSubstitution c:targetUnit ?unit .
   ?liquidSubstitution c:minVolume ?minVolume .
   ?liquidSubstitution c:maxVolume ?maxVolume .
   ?liquidSubstitution c:substitute ?substitute .
   FILTER (lcase(str(?kindof)) = "cocktail://ingredient/lemon") .
   FILTER (lcase(str(?sourceUnit)) = "cocktail://unit/piece") }
```

Listing 20: Substitution query written in SPARQL

Temperature classification

A temperature classification is needed, because recipes are mostly for cold drinks but also for hot and room-temperature drinks. Recipes contain information about preparation and the use of ice (Equation 4.59). If a recipe contains the preparation $boil$, it has to be hot. If a recipe contains $not\ boil$, the best practice is to make it hot but not boiling. But if a guest gets a boiling drink, he has to wait before he can drink. Ingredients contain the prefix hot such as $hot\ water$, which is an implicit preparation information that classifies a recipe also as hot. If a recipe hass $no\ ice$, which means it should not be cold but also not hot. Therefore, it has to be served at $room\ temperature$. The most frequently used method is to use ice, which means this information is not always explicit. If the rules are not matched, the recipe result is classified as $cold$.

$$\begin{aligned} boil &\rightarrow hot \\ not\ boil &\rightarrow hot \\ hot\ water &\rightarrow hot \\ no\ ice &\rightarrow room\ temperature \\ &\rightarrow cold \end{aligned} \tag{4.59}$$

4.3.6. Plausibility metric

After the parsing of a cocktail recipe, the recipe has to be checked for plausibility. A recipe needs a name. This is not important for the recommendation process, but is significant for reasons of clarity and comprehensibility. Recipes usually have a name. Therefore, it indicates that the parsing process was wrong or the input did not comprise a complete recipe.

A cocktail recipe needs two or more ingredients. A recipe with only one ingredient would not be of use. If there is no ingredient, then this is another indication of a parsing error or wrong input data. Additionally, the amount of recognized items that are mapped to assignments has to be above a static threshold of 0.6. It is a result of experience with testing data. Preparation and any type of glassware could be missing too. These are optional features.

The last part of the validation process involves a consistency check. With feature reasoning, the assignment is complete, but not all combinations of assignments are useful. There are two ways of using measurement units. Qualitative units are used if an ingredient such as bitter is required in small quantities or is a solid ingredient. If all assignments with qualitative units are hidden, then there are only relative units, such as Brandy Crusta, or quantitative units, such as Manhattan.

BRANDY CRUSTA.[6]
3 dashes Maraschino.
1 dash Angostura Bitters.
4 dashes Lemon Juice.
25% Curacao.
75% Brandy.
Stir and strain into prepared glass, adding slice of Orange.

MANHATTAN COCKTAIL[7]
1 dash of gum syrup, very carefully;
1 dash of bitters (orange bitters);
1 dash of curacao, if required;
1/2 wine glass of whiskey;
1/2 wine glass of sweet vermouth;
stir up well; strain into a fancy cocktail glass;

A validation criterion is that only qualitative and quantitative units or qualitative and relative units be used. If this criterion is not accomplished, the domain-specific reasoning cannot be correct. For a recommendation, all items have to be known. If an item is not known, then the distance is wrong.

4.3.7. Optimization of data quality

There are errors in the words. Often special characters are not recognized, or the noise of the photocopy came out as a symbol, such as a dot, or a random single special character, e.g. \$ (Equation 4.60). These errors are manageable, because a word with one wrong character is recognizable with the Levenstein distance-based acceptance criteria (Equation 4.32). Moreover, one senseless character between two words can be removed by a stopword list of $killing\ chracters$.

$$\begin{gathered} Cr\pounds me\ de\ Cacao \\ \$\ Shake \end{gathered} \tag{4.60}$$

[6] 1937 William J. Tarling — Approved Cocktails
[7] 1882 Harry Johnson — Bartender's Manual

These books contain many recipes, but there are areas without recipes as well. Each book has an introduction, a table of contents, an index of ingredients, as well as explanations for preparations, glassware, and ingredients. For parsing these books, all areas without cocktails are removed manually. There are cases of hardship (Listing 21), which is why the recipes are manually corrected. Incorrectly recognized words are corrected, and numbers are checked. All kinds of text that do not contain recipes or only references to recipes (such as 'see recipe No. 10 and replace ...') are removed.

```
ORGEAT LEMONADE.
(Use a large bar glass.)
1\$ wine glass of orgi~l syrup;
4 tiiblcsl)o~7ii'iil of ~ugar;
(1 to 8 di~sln.~s of \annote{Iriiion}{Is this an actual word?} juice;
2 glass of sliavril ire;
Fill the gli~i-s with water;
Mix up \annote{vvtlll ;mil orniiriient}{Are these actual words?} with grapes, berries, etc.,
in season, in 11 ti~steful manner ariil serve with a
straw.
r 1
```

Listing 21: Example of a case of hardship

4.3.8. Development process

Cocktail recipes of the chosen testing set contain many different combinations of properties. For the target structure, the properties have to be recognized. In the development process, it starts with a small and simple recipe set (Figure 4.26). If the parse result of a recipe is not expected, a recipe prototype that represents the problem is added to the test set. In this state, a toolset is used to solve the problem. The toolset combines rules to process items, phases, and type hierarchies including specific key words, categories in RDF, stopwords and killing/key characters. This toolset is developed continuously.

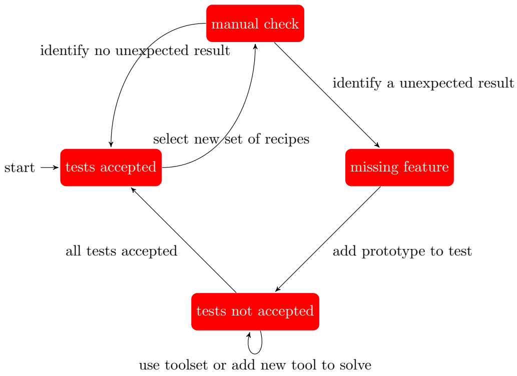

manual check
identify no unexpected result
identify a unexpected result
select new set of recipes
start
tests accepted
missing feature
all tests accepted
add prototype to test
tests not accepted
use toolset or add new tool to solve

Figure 4.26.: Development process of parsing

This process is done in several iterations that end the analysis. The development process provides more supported features, but it also raises the rate of chaos. Not every rule or phase is useful. The analysis aims to find generalized rules or unused rules. Similar rules are summarized, or one rule is changed to a more general rule. Rules have restrictions, such as types or values, which have to match. The strongest restriction is a restriction of value. A rule is more general if the number of restrictions is smaller, or a super type of a restriction in the type hierarchy is used. The generalization has to pass all tests to be accepted. The result is a test-driven development.

4.3.9. Statistics of cocktail parsing

The recipe recognition resulted in 2,566 recipes, 84 % of which are classified as valid and 16 % are invalidated by the plausibility metric. Most invalid recipes contain few recognized ingredients because the assignments are not parsable. Invalid unit combinations pose a small problem. The recognition of the title was initially an important metric to find errors. Meanwhile, it occurs infrequently.

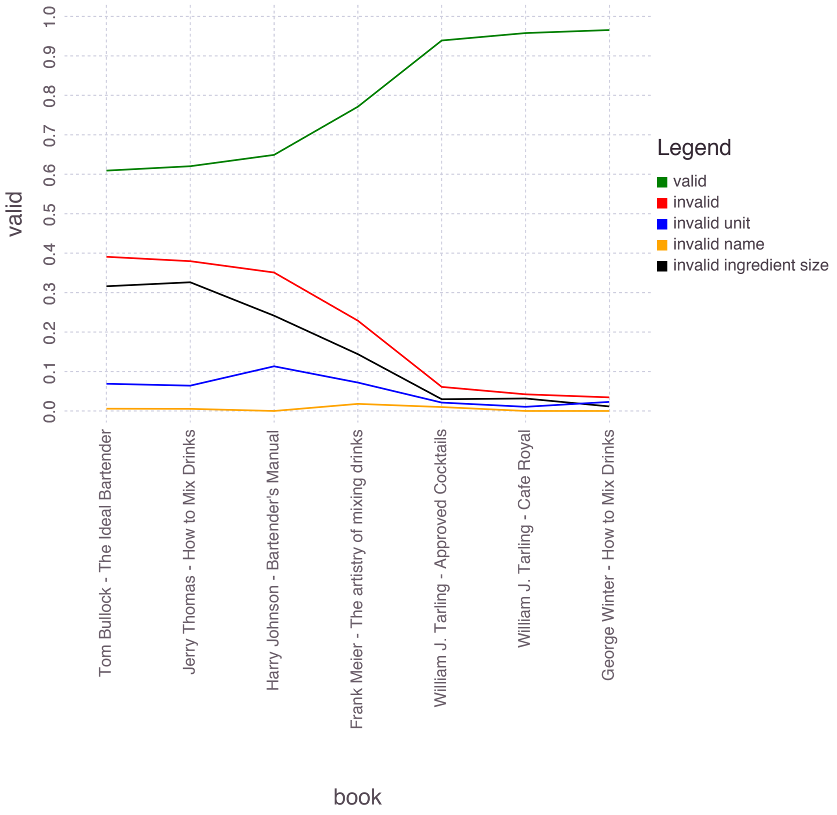


Figure 4.27.: Error rate of book parsing

The valid rate (Figure 4.27) of three books is higher than 84 %. These books contain recipes with fewer sentences and use a more formal style to present recipes. The books with the lowest plausibility rate have the highest median of recipe sizes (Figure 4.28). The recipe size is measured by the count of characters. This measurement reveals the complexity of a book. The book by Tom Bullock contains partially long phrases at the start of a recipe, resulting in a high error rate.

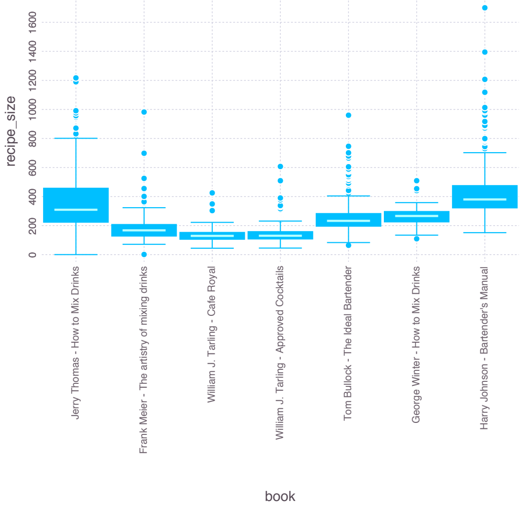


Figure 4.28.: Size of recipe distribution

There are several features that are not supported by the parser. Optional ingredients are yet to be supported. Additionally, $or - relations$ that contain commas (Equation 4.61) are not supported. This is often used in meta-recipes, which at the most contain superordinates and no measurement unit or numbers. The parser needs uniquely understandable words. $Wineglass$ is a drinking glass as well as an old measurement unit. In this case, the measurement units are more important. The ontology is aware that a wineglass is a measurement unit.

$$rum,\ gin\ or\ whiskey \tag{4.61}$$

The beginning and end of an assignment are the most important phases to extract every assignment correctly. However, the parser does not have the opportunity to recognize this if it is not explicitly given. A newline and a semicolon are explicitly separated, but if the start of an assignment is not indicated by such key words, the parser does not find these separations. The ontology and the stopword list are extendable to find more items. Default quantities and substitutions have to be maintained to obtain better results.

The parsing process needs 312 s for 2,167 validated recipes. 143 ms per recipe is an acceptable performance, which is the result of using indexer and caching mechanisms.

The recipes are varied and the data quality can be very low. The result is that recipes considered item lists need alternate parsing between a known and an unknown item. The experiment of feature extraction of unstructured recipes shows that recipes are recognizable. Nevertheless, there are opportunities to optimization.

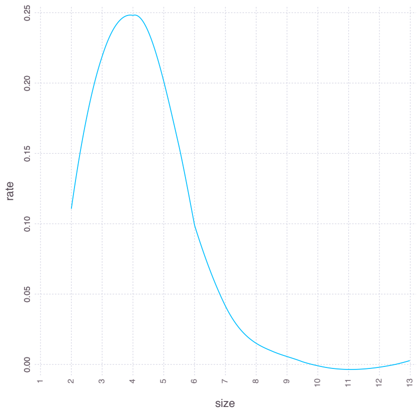


Figure 4.29.: Frequency rate of number of ingredients per recipe

The number of ingredients per recipe is basically three to five ingredients (Figure 4.29). Owing to the plausibility metric, the minimum is two ingredients. Recipes with more than seven ingredients are very rare.

The most frequent ingredients (Figure 4.30) are *lemon juice*, *syrup*, and types of *spirits*. There are also *soda* or *water* and different kinds of *vermouth*. Except for Angostura bitters, Both's Old Tom Gin, and Cointreau, there are only common ingredients without brands. All ingredients are usually known to the domain experts.

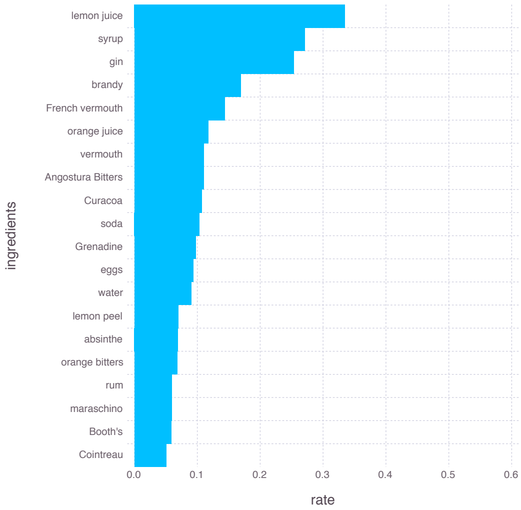


Figure 4.30.: Frequency rate of ingredients per recipe (limited to minimum rate of 5%)

Information about preparation of a recipe is concentrated on stir or shake (Figure 4.31). Mix — the generalization of both — is the third most frequent preparation. About 9 % of recipes do not contain information about preparation. Other preparations such as boil or float are negligible because of a frequency rate lower than 1 %. 1.2 % of recipes contain combinations of preparations, but these are not further considered due to their low frequency rate.

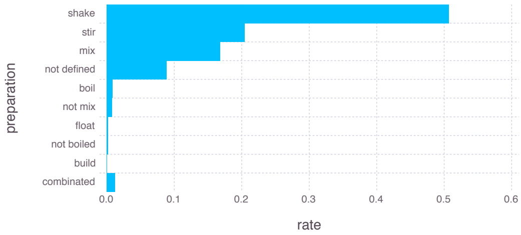


Figure 4.31.: Frequency rate of preparation per recipe

$Shake$, $stir$, or mix include using of ice implicitly. Therefore, the result is a cold drink. Hot drinks are rare, which is why it is assumed that a drink with no preparation information is a cold drink with usual preparation such as $shake$. The temperature classification based on ice and preparation states that 96 % recipes result in a cold cocktail. However, there also recipes that are served at room temperature or hot (such as toddies).

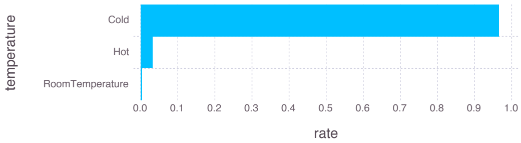


Figure 4.32.: Frequency rate of ice recommendations per recipe

Information about drinking glassware is only available in 40 % of recipes (Figure 4.33). Most are cocktail glasses, tumblers, and wineglasses. The other glasses have a low frequency. Bottles and glassware of preparation are less frequent than drinking glasses.

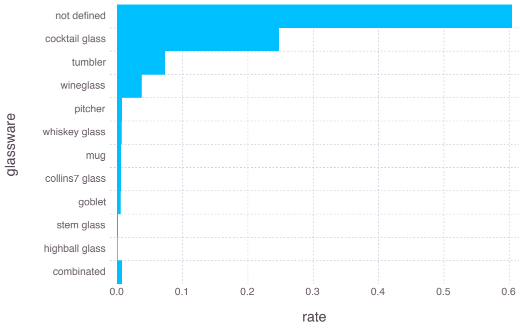


Figure 4.33.: Frequency rate of drinking glasses per recipe

Recommendations about the type of ice that should be used are also rare. 76 % of recipes have no information and 14 % contain no further information about the type of ice.

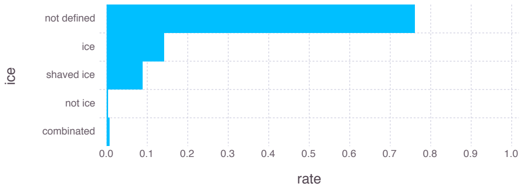


Figure 4.34.: Frequency rate of ice recommendations per recipe

Ingredients with quantity and preparation are the most frequent features, which are usable by distance functions. Each kind of glassware and ice is too rare to use it for distances.

Summary

For the feature extraction of unstructured recipes, a knowledge-based approach is used to find known items. Rules are used to transform tokens to recognized items with a higher abstraction level, such as assignments. The cleaning phase is used to get a normalized item list. The context analysis is used to find assignments with a domain-specific language. The feature reasoning transforms the extracted recipe in a comparable form.

By using the ontology based on basic categories the comparable form is stored in the target data structure which is computer-readable for recommendation and also readable for validation by domain experts. For a better recognitions rate, optimizations in the rule processing and ontology is necessary, nevertheless the dealing with huge volume of data (challenge two) is successfully archived.

4.4. Distances between classic recipes

For recommendation the a distance measurement is considered in this experiment (challenge three). It is assumed that classic recipes have been known for a long time, because they contain a characteristic that isolates them from each other. A recipe that is not different from others would be forgotten. 52 recipes are clustered by domain experts into 19 clusters and extracted in the first experiment to measure how well the distances work[8]. This is the first step to get an idea about how distances work.

One file is used for each cluster that represents the idea of one classic cocktail (Appendix B):

- lemonade.xml (three recipes)
- crusta.xml (three recipes)
- brandypunch.xml (two recipes)
- julep.xml (three recipes)
- alexander.xml (two recipes)
- aromatic.xml (two recipes)
- vermouth.xml (three recipes)
- flip.xml (two recipes)
- tomcollins.xml (two recipes)
- absinth.xml (three recipes)
- eggnogg.xml (two recipes)
- whiskeysour.xml (two recipes)
- manhattan.xml (three recipes)
- daiquiri.xml (nine recipes)
- japanesecocktail.xml (two recipes)
- jackrose.xml (two recipes)
- ginfizz.xml (two recipes)
- cloverclub.xml (three recipes)
- sidecar.xml (two recipes)

The clusters contain recipes that differ only in terms of small things, such as a different kind of gin or if they use a lemon twist or not. They use different kinds of units such as cl or $ounces$. These recipes are extracted and are constant in an XML data structure (section 4.2). The reference to the book is added so as to be able to reconstruct these recipes.

[8] Previous version is located in Sections 4.4–8: https://users.informatik.haw-hamburg.de/ ubicomp/projekte/master2015-proj/sippel.pdf

4.4.1. Item path

The similarity between items is defined by shared categories in the ontology. The type is referred to the imaginable class (Listing 22). All classes that do not present superordinates are subclasses of the imaginable class. Apart from ice and all subtypes of glassware, the basic ingredient categories and subordinates are subclasses that represent the basic categories such as *gin* and subordinates such as *London dry gin*. The superordinates such as *spirits* are clearly excluded, because the shared properties between two *spirits* such as *absinthe* and *gin* are too low.

```
<rdfs:Class rdf:about="cocktail://item/imaginable">
  <rdfs:label>imaginable item</rdfs:label>
</rdfs:Class>
<rdfs:Class rdf:about="cocktail://ingredient/basic">
  <rdfs:label>basic category of ingredient</rdfs:label>
  <rdfs:subClassOf>cocktail://item/imaginable</rdfs:subClassOf>
</rdfs:Class>
<rdfs:Class rdf:about="cocktail://ingredient/subordinate">
  <rdfs:label>subordinate ingredient</rdfs:label>
  <rdfs:subClassOf>cocktail://item/imaginable</rdfs:subClassOf>
</rdfs:Class>
```

Listing 22: Modification of RDF classes

The query written in SPARQL has to find the path of categories. Even assuming that the items are found, there could be parent categories. These are requested in optional statements. Only a triple $(?kindof_x\ kindof\ ?kindof_{x+1})$ and a type check are necessary. The type check is needed to prevent items of differing taxonomies such as superordinate ingredients from appearing in the result. The maximum tree depth is defined as 4, so only three parents could be found. The result presents a list of ingredients showing the ingredient path in the ingredient

```
SELECT  ?type0 ?kindof0 ?type1 ?kindof1 ?type2 ?kindof2 ?type3 ?kindof3
WHERE {
  ?kindof0 <rdfs:Literal> ?name .
  ?kindof0 <rdf:type> ?type0 .
  ?type0 <rdfs:subClassOf> "cocktail://item/imaginable"
  FILTER ( lcase(str(?name)) = "Plymouth" )
  OPTIONAL {
    ?kindof0 <c:kindof> ?kindof1 .
    ?kindof1 <rdf:type> ?type1 .
    ?type1 <rdfs:subClassOf> "cocktail://item/imaginable"
    OPTIONAL {
      ?kindof1 <c:kindof> ?kindof2 .
      ?kindof2 <rdf:type> ?type2 .
      ?type2 <rdfs:subClassOf> "cocktail://item/imaginable"
      OPTIONAL {
        ?kindof2 <c:kindof> ?kindof3 .
        ?kindof3 <t:type> ?type3 .
        ?type3 <rdfs:subClassOf> "cocktail://item/imaginable"
      }
    }
  }
}
```

Listing 23: Query of Ingredients written in SPARQL

tree. The searched ingredient is always the first item in the path. The variables $kindof_x$ and $type_x$ are URIs. The result is mapped to data structures. The data structure is chosen by $type_x$, which implements the trait $item$. The variable $kindof_x$ represents the value uri.

In the example (Figure 4.35), there is a subordinate ingredient $Plymouth$, which has a parent gin as a basic category of ingredients, as well as a superordinate $spirits$, which is not declared as an imaginable ingredient.

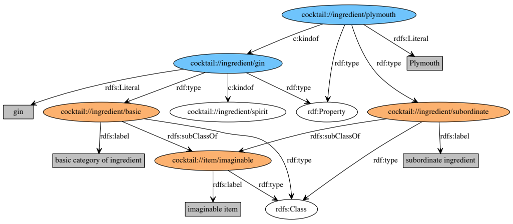


Figure 4.35.: Example of a ingredient categorization

The path of $Plymouth$ contains itself and the parent gin (Equation 4.62). The superordinate is ignored and the types are represented by the chosen data structure such as $BasicIngredient$.

$$\begin{aligned} path_I(Plymouth) = {} & SuboridinateIngredient(\text{cocktail://ingredient/plymouth}) :: \\ & BasicIngredient(\text{cocktail://ingredient/gin}) :: Nil \end{aligned} \tag{4.62}$$

4.4.2. Units

The weight of an ingredient for distance function is defined by ratio referred to the total volume, therefore the used quantity have to be extracted. For a comparable quantity, the unit has to be normalized. The main task of the unit in the ontology is to identify measurement units and to convert them into the standard unit cl. This conversion normalizes the quantity. The convertable measurement units are separated into quantitative and qualitative units. Quantitative units such as cl are scalable, while qualitative units such as $dash$ are not. There are metric units such as ml and American or British units such as $ounce$. For non-metric units, there are synonyms like singular and plural words.

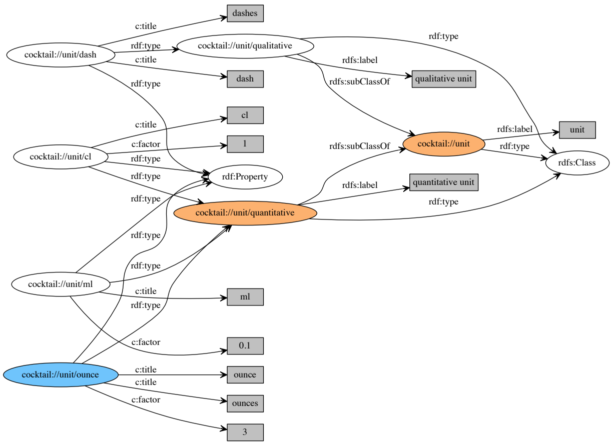


Figure 4.36.: Subset of unit categories

Quantitative units contain factors in the ontology, which are fetched by query (Listing 24). The uncapitalized string of names needs to match the exemplary name *dash*. The other types of units (referenced units, qualitative units, relative units, etc.) have default values. If these default values are too imprecise, they are converted beforehand by substitution (subsubsection 4.3.5) into metric and quantitative units.

```
SELECT  ?type ?unit ?factor
WHERE {
  ?unit <rdfs:Literal> ?name .
  ?unit <rdfs:type> ?type .
  ?type <rdfs:subClassOf> "cocktail://unit"
  FILTER ( lcase(str(?name)) = "dash" )
  OPTIONAL {
    ?unit <c:factor> ?factor
  }
}
```

Listing 24: Query of units written in SPARQL

The conversion of a quantity q, which contains a value and a unit, into another unit u, is defined with the factor to the standard unit (Equation 4.63). In the example, the 30 ml is converted into 1 $ounce$.

$$convert(q, u) = Quantity(\frac{value(q) \cdot factor(unit)}{factor(u)}, u) \tag{4.63}$$

$$convert(Quantity(30, ml), ounce) = Quantity(\frac{30 \cdot 0.1}{3}, ounce)$$

4.4.3. Balance

The balance is an abstract perspective on the cocktail which leans on Jelineks odor model (subsection 3.6.1). The result of the survey based on appropriate features for recommendation are flavors (subsection 4.1.8), a extract of the most important ones to describe the classic recipes (Appendix B) are chosen: The cocktail balance represents six pieces of information — the amounts of sweet, sour, water, cream, bitter, and alcohol. These features are developed by describing classic recipes by domain experts and are qualitatively determined information. Alcohol is an exception because the ratio is available. It is necessary to get these six features for every ingredient. However, not all of this information is always available and the ontology does not contain all the information. Therefore, it needs a default logic approach. For example, the ontology does not contain balance information for a concrete gin product, but the balance of the gin prototype is known. Besides, the balance information of gin has to be used.

A basic category or subordinate contains a specific balance. Superordinates such as spirits contain different basic categories, but they share the balance. Spirits contain usually no sugar, sour, cream, and bitterness. The ratio of alcohol about 40 %. Therefore, superordinates are appropriate for declaring default values. Subordinates may differ from these values, but it is basically the correct default information. If the difference is high, more knowledge is necessary. The superordinates are added to the ontology (Listing 25), but it is not a subclass of an ingredient to prevent useless ingredient similarities (described in subsection 4.4.1).

```
<rdfs:Class rdf:about="cocktail://ingredient/superordinate">
  <rdfs:label>superordinate ingredient</rdfs:label>
</rdfs:Class>
```

Listing 25: Delclaration of superordinate class

The balance query is designed for use as default logic of balance information. The query search is performed by known ingredient URI (Listing 26). Self-declared and domain-specific elements such as $c : sweet$ are used to declare balance information. This information could be missing, so all declarations are optional. Only the ingredient URI has to be there. The balance information must also be in a parent ingredient, so it is declared as the $kindof$ triple. If there is another balance, then the information is found. As the tree depth is limited to 4 because a higher depth is not necessary for created ontology. This query contains three nested $kindof$ triples.

In this example, the given ingredient $Plymouth$ does not have balance information. The basic category gin has alcohol and water in the proportion of 0.47 and 0.53, respectively. The superordinate has alcohol and water in the proportion of 0.4 and 0.6, respectively. As

```
SELECT DISTINCT  ?sweet0 ?sour0 ?alcohol0 ?water0 ?bitter0 ?cream0 ?sweet1 ?sour1
        ?alcohol1 ?water1 ?bitter1 ?cream1 ?sweet2 ?sour2 ?alcohol2 ?water2 ?bitter2 ?cream2
        ?sweet3 ?sour3 ?alcohol3 ?water3 ?bitter3 ?cream3
WHERE {
  ?kindof0 <rdfs:Literal> ?name
  FILTER ( str(?kindof0) = "cocktail://ingredient/plymouth" )
  OPTIONAL { ?kindof0 <c:sweet> ?sweet0 }
  OPTIONAL { ?kindof0 <c:sour> ?sour0 }
  OPTIONAL { ?kindof0 <c:alcohol> ?alcohol0 }
  OPTIONAL { ?kindof0 <c:water> ?water0 }
  OPTIONAL { ?kindof0 <c:water> ?bitter0 }
  OPTIONAL { ?kindof0 <c:water> ?cream0 }
  OPTIONAL { ?kindof0 <c:kindof> ?kindof1
    OPTIONAL { ?kindof1 <c:sweet> ?sweet1 }
    OPTIONAL { ?kindof1 <c:sour> ?sour1 }
    OPTIONAL { ?kindof1 <c:alcohol> ?alcohol1 }
    OPTIONAL { ?kindof1 <c:water> ?water1 }
    OPTIONAL { ?kindof0 <c:water> ?bitter1 }
    OPTIONAL { ?kindof0 <c:water> ?cream1 }
    OPTIONAL { ?kindof1 <c:kindof> ?kindof2
      OPTIONAL { ?kindof2 <c:sweet> ?sweet2 }
      OPTIONAL { ?kindof2 <c:sour> ?sour2 }
      OPTIONAL { ?kindof2 <c:alcohol> ?alcohol2 }
      OPTIONAL { ?kindof2 <c:water> ?water2 }
      OPTIONAL { ?kindof0 <c:water> ?bitter2 }
      OPTIONAL { ?kindof0 <c:water> ?cream2 }
      OPTIONAL { ?kindof2 <c:kindof> ?kindof3
        OPTIONAL { ?kindof3 <c:sweet> ?sweet3 }
        OPTIONAL { ?kindof3 <c:sour> ?sour3 }
        OPTIONAL { ?kindof3 <c:alcohol> ?alcohol3 }
        OPTIONAL { ?kindof3 <c:water> ?water3 }
        OPTIONAL { ?kindof0 <c:water> ?bitter3 }
        OPTIONAL { ?kindof0 <c:water> ?cream3 }
      }
    }
  }
}
```

Listing 26: Query of balance written in SPARQL

sweetness is not declared, the default value of the balance property, which is not found, stands at 0.

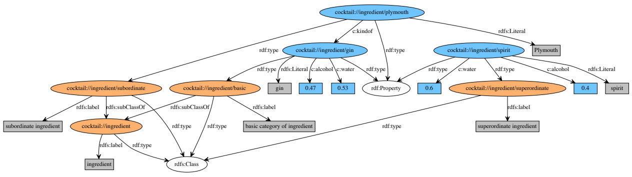


Figure 4.37.: Balance in ingredient categories

The path contains the balance information of all the single ingredients — first $Plymouth$, then gin, and finally $spirits$ (Equation 4.64). The question mark is used as a symbol to indicate that the information remains unknown. The result is a balance without any unknown element.

$$\begin{gathered} balance(water, alcohol, sweet, sour, cream, bitter) \\ path_B(Plymouth) = (?, ?, ?, ?, ?, ?) :: (0.53, 0.47, ?, ?, ?, ?) :: (0.6, 0.4, ?, ?, ?, ?) :: Nil \\ balance(Plymouth) = (0.53, 0.47, 0, 0, 0, 0) \end{gathered} \tag{4.64}$$

4.4.4. Distances for recommendation

Distances are the main part of the recommendation component because they say something about the shared facts of two values such as ingredients.

The necessary data structure for the distances is $path$. The distance of steps of two paths is the minimal count of steps to find two equal items in the path. In the example (Figure 4.38), the two paths are combined in one graph. Equal URIs are presented by a single node. The orange edges show the steps required to get the equal node b. There are three necessary steps here.

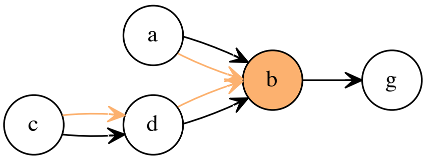


Figure 4.38.: Graph of paths with common node

The function of steps (Equation 4.65) has several aims. The first is to scale the distance of steps between 0 and 1. The second is to ensure that the distance of steps is independent of the path sizes. The last of these is that distance of steps has to approximate smoothly to 1. If no equal item is found, then the distance is 1.0.

$$stepFunction(n) = 1 - \frac{g}{\sqrt{n}} \tag{4.65}$$

The function of steps is only designed along these aims and there is no connection with knowledge. If the step count is 0, then the distance of steps must also be 0. Because of the maximum depth of 4 in the graph, only three steps per path are possible. There are six maximum steps. Therefore, the value n is defined in the range of $[1..6] \in \mathbb{N}$. The gradient g is to configure how intensively the function increases. A gradient of 0.85 is a result of the experiment and of a function (Figure 4.39), which starts with an intensive increase and then bottoms up. The distance of steps cannot be negative.

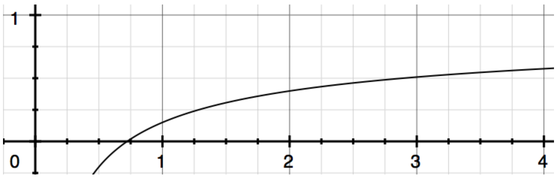


Figure 4.39.: Example of graph to visualize the step function

Ingredient distance

The distance of a distance pair (I_a, I_b) is a path distance (Equation 4.66). A quantity weighting is added because the quantity tells something about the importance. $6 cl\, gin$ are more important than $1\ cl\ sugar\ syrup$. The weight is the quantity in relation to the volume of the cocktail. The volume is the sum of quantities of all quantitatively measured ingredients. All quantities are transformed into the standard unit cl.

$$d_{DPI}(a,b) = stepFunction(I_a, I_b) \cdot \frac{quantity(I_a)}{volume(a)} \tag{4.66}$$

The distance of steps has the lowest value 0 if both ingredients remain the same. The quantity could be different. A different quantity has no effect because it is multiplied by 0. Therefore, d_{DP} has to be divided into two kinds of distance functions — the ingredient-based distance function (Equation 4.66) and the quantity-based distance function (Equation 4.67). The DPQ needs a weight w to prevent too high distances compared to DPI, because this is only used for equal ingredients. A proper weight based on the experiment is 0.25.

$$d_{DPQ}(a,b) = |\frac{quantity(I_a)}{volume(a)} - \frac{quantity(I_b)}{volume(b)}| \cdot w \tag{4.67}$$

The distance pair is dependent on the distance of steps (Equation 4.68).

$$d_{DP}(a,b) = if(stepDistance == 0)\ d_{DPQ}(a,b)\ else\ d_{DPI}(a,b) \tag{4.68}$$

A cocktail recipe contains a list of ingredients. The order must not affect the distance, because the order could be different and don't change the recipe. If there is an ingredient I_a of the cocktail a, the aim would be to find the most similar ingredient to I_a in the ingredients of cocktail b. The number of ingredients of a are n. The number of ingredients of b are m.

The distance d_I (Equation 4.69) between ingredients of recipe a and the ingredients of b represents the ingredient distance between two recipes. It uses the distance d_{DP}, which maps an ingredient to another ingredient. A mapping is not completely accurate, the distance must be calculated in both directions to catch all the ingredients in the distance. The distance d_I sums up all minimum d_{DP} distances in both directions. One direction is already scaled to 1 because the ingredients are weighted by the ratio. The sum of the two directions must be divided by 2 to scale d_I to 1.

$$d_I(a,b) = \frac{\sum_{i=0}^{n} arg\ min(d_{DP}(I_{a_i}, I_{b_j})) + \sum_{j=0}^{m} arg\ min(d_{DP}(I_{b_j}, I_{a_i}))}{2} \tag{4.69}$$

In the following example, there are two different recipes — a Negroni and a Mezcal Negroni.

Negroni
3.0 cl Punt e Mes
3.0 cl Plymouth
3.0 cl Campari
1.0 piece orange zest
(stir, whiskey tumbler)

Mezcal Negroni
3.5 cl Tlacuache silver Leyenda
3.5 cl Carpano Antica Formula
2.0 cl Gran Classico
(stir, cocktail glass)

The distance pairs are as follows (Listing 27). First, there are the mappings of ingredients of the *Mezcal Negroni* recipe to the ingredients of the *Negroni* recipe. This follows the mapping in the other direction. If there is no similar ingredient, it is shown by a question mark. The distance is displayed in the middle. All distances are based on ingredients, because there are no equal ingredients. As the *Tlacuache silver Leyenda* does not have a similar ingredient, the distance of steps is 1 and, because of the ratio to the volume of the recipe, the d_{DP} is 0.39. This has a huge impact. The *Gran Classico* and *Campari* are types of bitter liquors. *Carpano Antica Formula* and *Punt e Mes* are types of *red vermouth*. The distance of steps is low, but their ratio is also low, so their effect in the d_{DP} is not very high. Taking the other direction is also similar. The *Plymouth* does not have a similar ingredient that has a significant effect. The distances are rounded off to two decimal places.

The sums of the mappings are not equal (Equation 4.70). In this case, they are similar. The ingredient distance d_I (Equation 4.71) tells that these drinks have some similarities such as *red*

```
Mezcal Negroni => Negroni
SimpleIngredientBased[3.5 cl Tlacuache silver Leyenda <= (1.0,0.39) => ?]
SimpleIngredientBased[3.5 cl Carpano Antica Formula <= (0.40,0.16) => 3.0 cl Punt e Mes]
SimpleIngredientBased[2.0 cl Gran Classico <= (0.40,0.09) => 3.0 cl Campari]
Negroni => Mezcal Negroni
SimpleIngredientBased[3.0 cl Punt e Mes <= (0.40,0.13) => 3.5 cl Carpano Antica Formula]
SimpleIngredientBased[3.0 cl Plymouth <= (1.0,0.33) => ?]
SimpleIngredientBased[3.0 cl Campari <= (0.40,0.13) => 2.0 cl Gran Classico]
SimpleIngredientBased[1.0 piece orange zest <= (1.0,0.01) => ?]
IngredientDistance = 0.63 + 0.60 / 2 = 0.62
```

Listing 27: Ingredient distance of a Negroni

$vermouth$, because in the range of 0 to 1, it is in the middle. However, they have differences such as gin and $mezcal$.

$$sum(Negroni \rightarrow Mezcal\ Negroni) = 0.63 \tag{4.70}$$

$$sum(Mezcal\ Negroni \rightarrow Negroni) = 0.60$$

$$d_I = \frac{0.63 + 0.60}{2} = 0.62 \tag{4.71}$$

An example of two very similar recipes (Listings 28) has quantity-based and ingredient-based distances. Each distance of steps has a low value; therefore, the ingredient distance is low. It is only 0.10. One of these recipes should not lead to a recommendation of the other because they are too similar.

```
Negroni => Negroni with Punt e Mes
SimpleIngredientBased[3.0 cl red Vermouth <= (0.15,0.05) => 3.0 cl Punt e Mes]
SimpleIngredientBased[3.0 cl Gin <= (0.15,0.05) => 3.0 cl Plymouth]
SimpleQuantityBased[3.0 cl Campari <= (0.0,0.0) => 3.0 cl Campari]
SimpleQuantityBased[1.0 piece orange zest <= (0.0,0.0) => 1.0 piece orange zest]
Negroni with Punt e Mes => Negroni
SimpleIngredientBased[3.0 cl Punt e Mes <= (0.15,0.05) => 3.0 cl red Vermouth]
SimpleIngredientBased[3.0 cl Plymouth <= (0.15,0.05) => 3.0 cl Gin]
SimpleQuantityBased[3.0 cl Campari <= (0.0,0.0) => 3.0 cl Campari]
SimpleQuantityBased[1.0 piece orange zest <= (0.0,0.0) => 1.0 piece orange zest]
IngredientDistance = 0.10 + 0.10 / 2 = 0.10
```

Listing 28: Ingredient distance of two Negroni recipes

In an example of two absolutely different recipes (Listings 29), there is no similar ingredient. The ingredient distance stands at 1. Besides, one of these recipes is not a good recommendation for the other.

Balance distance

The balance distance shows how different recipes are with respect to balance. The aim is to find cocktails with the same characteristics. Every ingredient has a balance. The balance of a cocktail is the sum of balances of n ingredients (Equation 4.72).

```
Gin Fizz => Mezcal Negroni
SimpleIngredientBased[6.0 cl Gin <= (1.0,0.46) => ?]
SimpleIngredientBased[3.0 cl lemon juice <= (1.0,0.23) => ?]
SimpleIngredientBased[2.0 cl sugar syrup <= (1.0,0.15) => ?]
SimpleIngredientBased[2.0 cl soda <= (1.0,0.15) => ?]
Mezcal Negroni => Gin Fizz
SimpleIngredientBased[3.5 cl Tlacuache silver Leyenda <= (1.0,0.39) => ?]
SimpleIngredientBased[3.5 cl Carpano Antica Formula <= (1.0,0.39) => ?]
SimpleIngredientBased[2.0 cl Gran Classico <= (1.0,0.22) => ?]
CocktailDistance = 1.0 + 1.0 / 2 = 1.0
```

Listing 29: Ingredient distance of a Negroni and a Gin Fizz

$$balance(c) = \sum_{i=1}^{n} balance_i(water, alcohol, sweet, sour, bitter, cream) \cdot \frac{quantity(I_i)}{volume(c)} \tag{4.72}$$

$$d_B(balance) = water + alcohol + sweet + sour + bitter + cream \tag{4.73}$$

$$d_B(c_a, c_b) = d_B(|balance(c_a) - balance(c_b)|) \tag{4.74}$$

The difference between two balances (Equation 4.73) is a balance having a difference in each component, such as $sour$. The balance distance is the difference between the final balance of c_a and c_b (Equation 4.74). All components will be added up to a scalar distance.

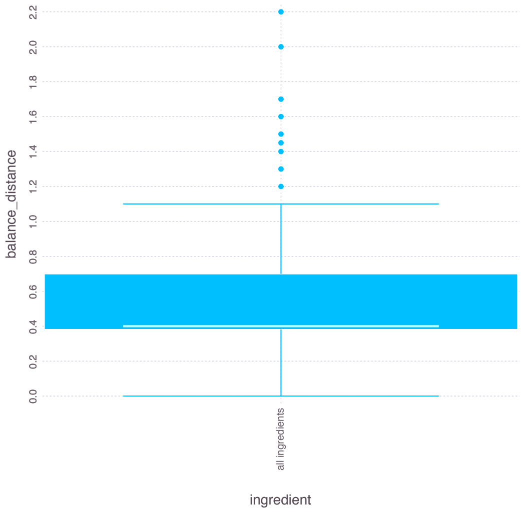


Figure 4.40.: Balance distribution of extracted ingredients

The balance distance is not scaled to 1. A balance such as $balance(1, 1, 1, 1)$ is unrealistic. An ingredient with a high ratio of alcohol such as *Absinth* does not contain a high ratio of sugar such as syrup. The distribution of ingredient balance distances in the ontology (Figure 4.40) shows that most distances are between 0 and 1. However, some ingredients such as lemon juice have a distance up to 2.2. As the values are qualitative, the precision is not measurable. Therefore, the balance distance between two recipes is not normalized to a range between 0 and 1, but it is positive.

```
Mezcal Negroni => Negroni
SimpleIngredientBased[3.5 cl Tlacuache silver Leyenda <= (1.0,0.33) => ?]
Balance(alcohol=0.4,sweet=0,sour=0,water=0.6,bitter=0,cream=0)
SimpleIngredientBased[3.5 cl Carpano Antica Formula <= (0.15,05) => 3.0 cl red Vermouth]
Balance(alcohol=0.18,sweet=0.25,sour=0,water=0.74,bitter=0,cream=0)
SimpleIngredientBased[2.0 cl Gran Classico <= (0.40,0.13) => 3.0 cl Campari]
Balance(alcohol=0.28,sweet=0.2,sour=0,water=0.6,bitter=0.8,cream=0)
MezcalNegroniBalance(alcohol=0.29,sweet=0.15,sour=0,water=0.65,bitter=0.27,cream=0)
Negroni => Mezcal Negroni
SimpleIngredientBased[3.0 cl red Vermouth <= (0.15,05) => 3.5 cl Carpano Antica Formula]
Balance(alcohol=0.18,sweet=0.2,sour=04,water=0.74,bitter=0,cream=0)
SimpleIngredientBased[3.0 cl Gin <= (1.0,0.33) => ?]
Balance(alcohol=0.47,sweet=0,sour=0,water=0.53,bitter=0,cream=0)
SimpleIngredientBased[3.0 cl Campari <= (0.40,0.13) => 2.0 cl Gran Classico]
Balance(alcohol=0.25,sweet=0.2,sour=0,water=0.6,bitter=1.0,cream=0)
SimpleIngredientBased[1.0 piece orange zest <= (1.0,01) => ?]
Balance(alcohol=0,sweet=0,sour=0,water=0,bitter=0,cream=0)
NegroniBalance(alcohol=0.30,sweet=0.13,sour=0.01,water=0.62,bitter=0.33,cream=0)
BalanceDistance = 0.13 =
DifferenceOfBalance(alcohol=0.01,sweet=0.02,sour=0.01,water=0.03,bitter=0.06,cream=0)
```

Listing 30: Balance distance of a Negroni

In the example of Negroni and Negroni Mezcal (Listing 27), the ingredient distance is very high because the two drinks do not share many properties. The balance distance shows more shared properties (Listing 30). The sum of all ingredient balances is similar, because both have the same alcohol strength and sweetness, and differ only slightly from dilution with water and bitterness. The balance distance is only 0.13. These recipes are not the same but have a similar characteristic that qualifies one of them to be a recommendation for the other.

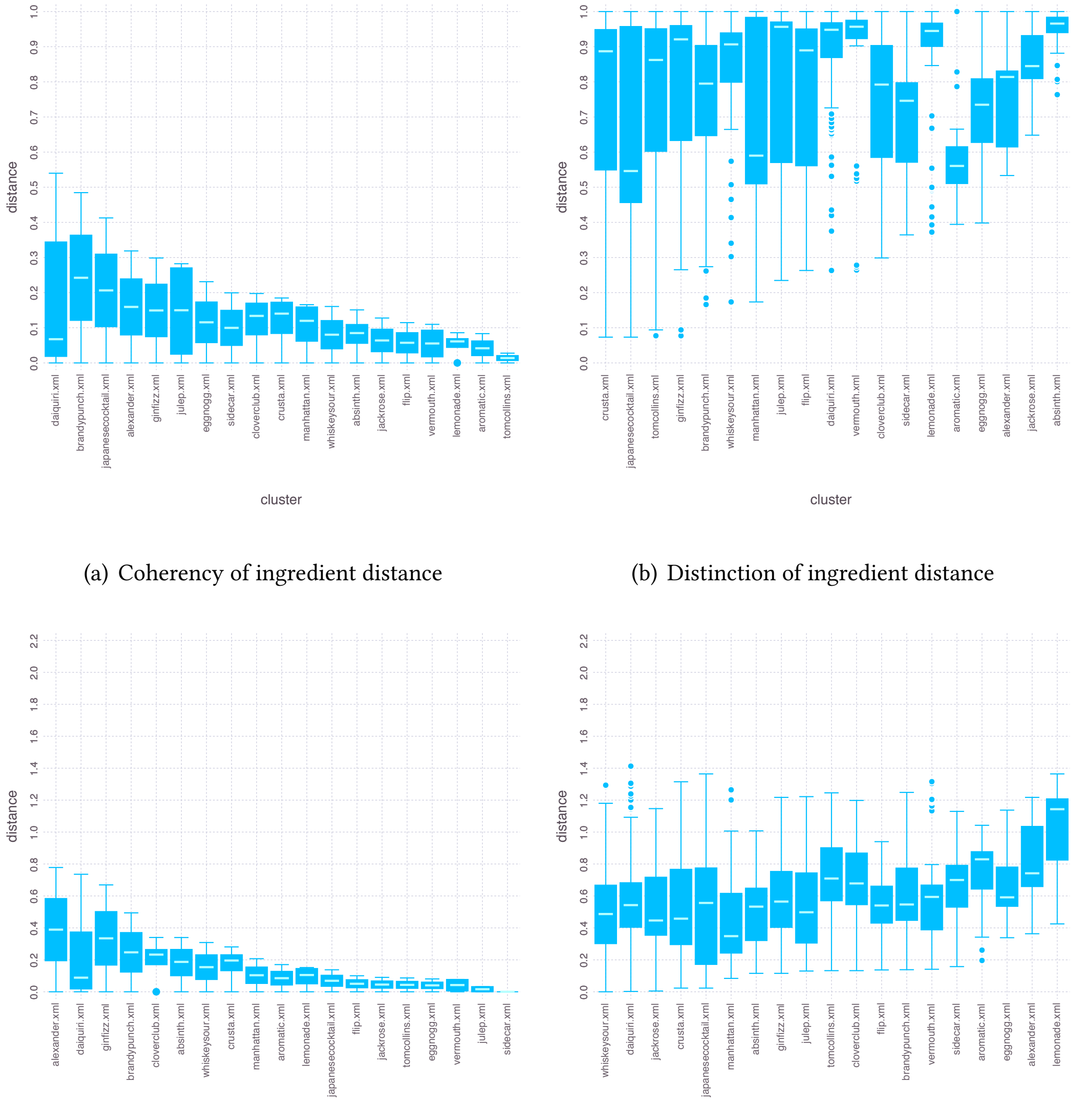
4.4.5. Evaluation of distance measurement

The balance of each ingredient is defined qualitative, which results usable data, but a quantitative measurement have to proved in future work for more precision. Techniques of chemical measurement are already developed. The ratio of alcohol is already available because each product have to show it on the label, sourness could be by ph-value, cream rate could be measurable by fat rate, bitterness is defined by international bitterness unit, sharp is defined by scoville scale, sweet could be estimated by user study, peat is measurable by phenol ratio. The opportunity to measure the balance should be always in mind.

Nevertheless, to get an first idea of how the distances work the precision of distances is evaluated in relation to the clustering. It is assumed that the maximum distance of a cocktail in one cluster must be low enough so that recipes of other clusters can get a higher distance. The measurement of coherence (Figure 4.41(a)) shows the distances between clusters. It is sorted in descending order. The first, the $daiquiri.xml$, cluster has a maximum distance of 0.54. The last one, the $tomcollins.xml$, has a maximum value of 0.03. These are positive results because recipes in the clusters are also similar in terms of distance function. The daiquiri cluster has a high distance, because there are different understandings of one recipe combined in one cluster. The name is an indication for similarity, but a deeper analysis is needed to find recipes that represent the same idea.

The next measurement (Figure 4.41(b)) shows the distance of each recipe from any recipe that is outside its own cluster. It is sorted in ascending order. Most of the recipes have distances higher than 0.5. These results confirm that the classics are distinct from each other, not in the same intensity. *Crustas* and *Japanese cocktails* have similarities, while a *Tom Collins* and a *Gin Fizz* are also a bit similar to each other. This result is according to domain expert's knowledge. This is a positive result, because the distance function represents domain knowledge.

The coherence of balance distances (Figure 4.41(c)) has to be similar to the ingredient distance coherence, and the balances should be as small as possible. The highest ingredient distances are in the *Japanese Cocktail* cluster or *Mint Julep* cluster, because these recipes contain a number of additional flavors. However, in terms of balance coherence, this cluster contains very small distances because these flavors do not change the balance. The distinction of balance distances (Figure 4.41(d)) shows that the balances are mostly higher, because other clusters often have other balances. Nevertheless, there are also low balance distances, which are the interesting ones for recommendation. These are considered in the next experiment. For the distance of ingredient and balance, the clusters are separated to other clusters by a distance of $0.3 - 0.4$. These found threshold shows that both distances shows an semantic difference between the recipes and is the key to use a nearest neighbor recommendation. In the next experiment, this value will be investigated for recommendation.



(a) Coherency of ingredient distance

(b) Distinction of ingredient distance

(c) Coherency of balance distance

(d) Distinction of balance distance

Figure 4.41.: Results of cluster experiment

Summary

The distinction and coherency metrics indicate that the domain-specific distance measurement based on path through the categories have a domain-related impact (challenge three) which is necessary for the validation in the next experiment.

A problem is that different recipe authors have different opinions. There are different special flavors (bitters, liquors, herbs, etc.), different ratios, and different products of one category, which shows that classics are not definable by one recipe; they are a collection of recipes presenting a single idea. A widespread collection of recipes is necessary to get different recipes containing one idea.

Further studies need to evaluate the clusters with a large number of domain experts in order to ensure quality of these clusters. This is a basis for validation of absolute distance measurement. The number of different recipes of each cluster has to be increased to evaluate a broader bandwidth of recipes.

4.5. Validation by domain-experts

Based on existing distance measurement this experiment validates the recommendation by domain experts (challenge four). The last experiment uses the extracted recipes represented in resulted target structure. An ingredient-based distance as well as a balance-based distance function is defined based on the extracted features by a huge number of recipes. The ingredient distance demonstrates the uniqueness of classic recipes, while the balance show a similar characteristic, which is an example of a good recommendation. This last experiment combines these results to get a working recommendation system.

Classic recipes are the popular ones. Therefore, it is assumed that these are preferred examples of recommendation. This experiment uses classic examples and the library of extracted recipes for recommendation. The results are validated by a domain expert to get feedback on the results. At first a self-proof is used to get a feeling whether the recommendation works as designed. This is an iterative process of recommendation, validation, and error analysis of feature extraction and maintaining of ontology. The resulted recommendation are rated by independent domain experts to reinforce the validation.

A recommendation needs to combine something known with something new, in context of the given distance functions there are two approaches of recommendation — the first is used to get recipes with the same balance but different ingredients and the second is used to get recipes of the same ingredients but with a different balance.

The recommendation approach uses the nearest-neighbor classification kNN of the given example e. The previous experiment results a empiric value of distance, which separates the distances into too near distance and distances which shows significant differences (Figure 4.41). In the first instance (Equation 4.75), called focus on balance, the nearest neighbors have an ingredient distance d_I higher than $t_I = 0.3$ and a balance distance lower than $t_B = 0.3$. Too low distances of ingredients are too similar while too high distances of balance are too different. The recommendation r gives a list of cocktail recipes. This is ordered increasingly according to ingredient distances. The first $k = 10$ elements are used. The second instance (Equation 4.76), called focus on ingredients, uses $t_I = 0.4$ as the maximum threshold of ingredient distance and $t_B = 0.4$ as the minimum threshold of distance of balance. If the focus is on balance, the balance distance has to be very low, because balance distance does not show which component of balance such as $sweet$ is different. If the distance is caused in only one component, the change is higher than it is distributed on all components. The focus on ingredient approach needs an higher threshold because it is more differences between the recipes necessary to get enough results.

$$r1 = kNN_B(e, t_I, t_B)) \tag{4.75}$$

$$r2 = kNN_I(e, t_I, t_B)) \tag{4.76}$$

4.5.1. Setting of offline experiment

In order to get a more reliable validation of a recommendation, it is necessary to ask several domain experts. Experts have different areas of interests (section 3.7), and so their focuses are different, resulting in their recommendations being different. If the given recommendation needed to be acceptable only to each expert, then each expert could give a different opinion but agree on the given recommendation.

The offline experiments with a static testing set and feedback by domain experts is used to test whether a recommendation is appropriate. A specific group of domain experts — such as bartenders or connoisseurs — will be offered the examples and a list of recommendations. The domain experts are able to rate the validity of each recommendation on a numeric scale (Equation 4.77). This scale is designed to present how acceptable a recommendation is.

$$[\underset{\text{(unacceptable)}}{-2}, \underset{\text{(slightly similar)}}{-1}, \underset{\text{(obviously)}}{0}, \underset{\text{(rather appropriate)}}{1}, \underset{\text{(appropriate)}}{2}] \tag{4.77}$$

For each cluster of classic recipes (Appendix B) the first recipe is used as a favorite to calculate the recommendations. These process is either done for the a focus on balance approach (in total

181 recommendations) and for the focus on ingredient approach (in total 141 recommendations). The pairs of favorite recipe and recommended recipe are stored in a file for sharing the data to domain experts easily. The domain experts add their rating of acceptance in another column and sent their results back.

4.5.2. First proof of validation approach

For the first proof of validation a set of validation criteria are created based on the given setting. The validation criteria are in respect of domain experts and specific to the approach. This qualitative measurement is independent of the used model of recommendation. It is a pretest to check the distances against assumptions. The balance should map polarizing flavors to find recipes containing similar flavors. However, the recipes are not based on this model. In the focus on balance, the following is used:

- 2: An appropriate recommendation contains two or more different ingredients. These are quantitatively measured (no dash or splash), or the ingredient with the highest quantity is different. Two different ingredients with a small quantity are rated as a single unique ingredient. Additionally, the polarizing flavors (sweet, sour, etc.) are similar but not absolutely equal.
- 1: A recommendation is appropriate if the differences between the recipes and the polarizing flavors are appropriate. However, there is one exception that breaks the general impression.
- 0: The recommendation is obvious if it is too similar to the example.
- -1: The recommendation is slightly similar if there are appropriate ingredients or appropriate polarizing flavors, but not both.
- -2: The recommendation is unacceptable if there are no appropriate ingredients or appropriate polarizing flavors.

In the focus on ingredients, the following is used:

- 2: An appropriate recommendation with two or more similar ingredients, which are quantitatively measured (no dash or splash), or the ingredient with the highest quantity, is similar. Additionally, the polarizing flavors (sweet, sour, etc.) are different.
- 1: A recommendation is appropriate if there are similarities between the recipes and the polarizing flavors are different, but there is one exception that breaks the general impression.

- 0: The recommendation is obvious if it is too similar to the example.
- -1: The recommendation is slightly similar if there are similar ingredients or different polarizing flavors, but not both.
- -2: The recommendation is unacceptable if there are no similar ingredients or different polarizing flavors.

The domain expert Sigurd Sippel, 27 years old and 9 years of experience in domain, uses the validation criteria to validate the recommendations. The recommendations of focus on balance contains 18 of 19 classics with 10 or more recipes (4.48(a)). The Alexander, which contains a lot of cream and sugar, is not very usual for classic recipes. For this example, the library of recipes is too small. Only one found recommendation (Figure 4.42) is appropriate because they use milk and strawberries instead of crème de cacao, gin, and cream. The quantity of sweetness and cream is appropriate, but the recommendation has an alcohol ratio of 0.

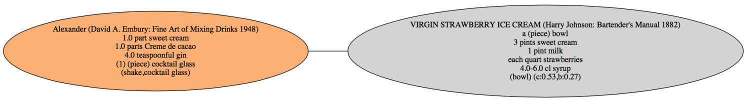


Figure 4.42.: Recommendation map of a Alexander

A Daiquiri recommendation (Figure 4.43) always contains the same sour and sugar rate. *Sunshine* contains additionally brandy, *Kick in the Pants* uses Forbidden Fruit liqueur (a pomelo liqueur). *Planters Punch* uses another type of rum, but this recipe is too similar.

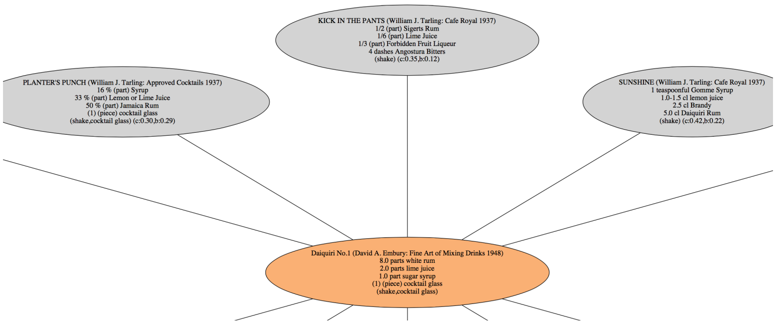

KICK IN THE PANTS (William J. Tarling: Cafe Royal 1937)
1/2 (part) Sigerts Rum
1/6 (part) Lime Juice
1/3 (part) Forbidden Fruit Liqueur
4 dashes Angostura Bitters
(shake) (c:0.35,b:0.12)
PLANTER'S PUNCH (William J. Tarling: Approved Cocktails 1937)
16 % (part) Syrup
33 % (part) Lemon or Lime Juice
50 % (part) Jamaica Rum
(1) (piece) cocktail glass
(shake,cocktail glass) (c:0.30,b:0.29)
SUNSHINE (William J. Tarling: Cafe Royal 1937)
1 teaspoonful Gomme Syrup
1.0-1.5 cl lemon juice
2.5 cl Brandy
5.0 cl Daiquiri Rum
(shake) (c:0.42,b:0.22)
Daiquiri No.1 (David A. Embury: Fine Art of Mixing Drinks 1948)
8.0 parts white rum
2.0 parts lime juice
1.0 part sugar syrup
(1) (piece) cocktail glass
(shake,cocktail glass)

Figure 4.43.: Recommendation map of a Daiquiri

The recommendation for a Manhattan (Figure 4.44) contains recipes that are very strong and a bit sweet, but free of sourness. *Whiskey Smash* contains mint instead of vermouth. *Zazerac* additionally contains lemon peel, anis, water, and orange bitters. The *King Cole* contains no vermouth and no Angostura or Maraschino cherry. Therefore, these recommendations are appropriate.

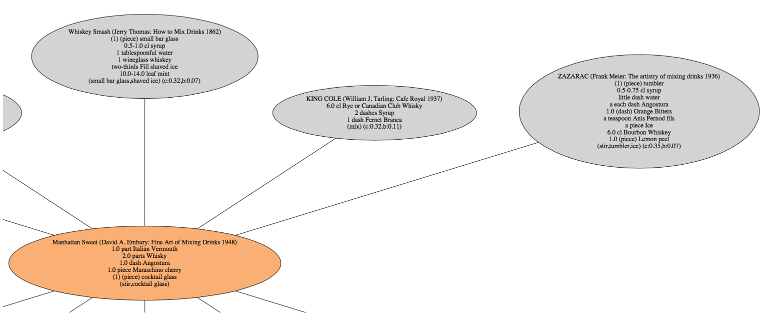

Whiskey Smash (Jerry Thomas: How to Mix Drinks 1862)
(1) (piece) small bar glass
0.5-1.0 cl syrup
1 tablespoonful water
1 wineglass whiskey
two-thirds Fill shaved ice
10.0-14.0 leaf mint
(small bar glass,shaved ice) (c:0.32,b:0.07)
KING COLE (William J. Tarling: Cafe Royal 1937)
6.0 cl Rye or Canadian Club Whisky
2 dashes Syrup
1 dash Fernet Branca
(mix) (c:0.32,b:0.11)
ZAZARAC (Frank Meier: The artistry of mixing drinks 1936)
(1) (piece) tumbler
0.5-0.75 cl syrup
little dash water
a each dash Angostura
1.0 (dash) Orange Bitters
a teaspoon Anis Pernod fils
a piece Ice
6.0 cl Bourbon Whiskey
1.0 (piece) Lemon peel
(stir,tumbler,ice) (c:0.35,b:0.07)
Manhattan Sweet (David A. Embury: Fine Art of Mixing Drinks 1948)
1.0 part Italian Vermouth
2.0 parts Whisky
1.0 dash Angostura
1.0 piece Maraschino cherry
(1) (piece) cocktail glass
(stir,cocktail glass)

Figure 4.44.: Recommendation map of a Manhattan

The medians of focus on balance (Figure 4.48(a)) are always positive, which is a good result. 16 classics have negative validation ratings, but these are in the minority.

The result of the focus on ingredients comprises 15 of 19 classics with 10 or more recipes (4.48(a)). The mint julep (Figure 4.45) contains only one recommendation, the *Camp Elysees* (Figure 4.45). Mint and cognac recipes are frequently available. These recipes are not found because the quantity of 32–48 leaves is not well extracted. The quantity declaration of mint is often very different, because the quality of mint varies and because the assignment of one or two sprigs is not quite precise. Owing to these declarations, the ingredient substitution does not work very well. The recommendation also contains cognac and a different balance because of the lemon juice, which is missing in the example. However, the herb flavor of Chartreuse is not appropriate to the mint julep, which is why it is only appropriate. Herb is not included in the balance, which is a deficit of the model.

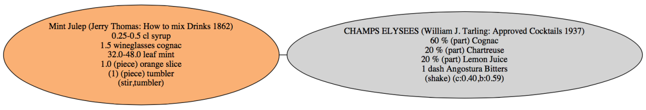


Figure 4.45.: Recommendation map of a Julep

The recommendation for the absinthe cocktail (Figure 4.46) contains different styles of mixing absinthe. The example contains sugar and a lemon twist, which are missing in *French style of mixing absinthe*. This recipe contains only absinthe and water. It is only rather appropriate because the balance is changed, but the ingredients are too similar. The *American style of mixing absinthe* uses anisette, which is also sweet and distinct from absinthe. Therefore, it is rather appropriate. The *Macaroni* uses vermouth but no water or lemon twist, which results in a stronger balance. This is appropriate. The *Pansy* contains Angostura and Grenadine additionally instead of water. It is a stronger recipe and is also appropriate.

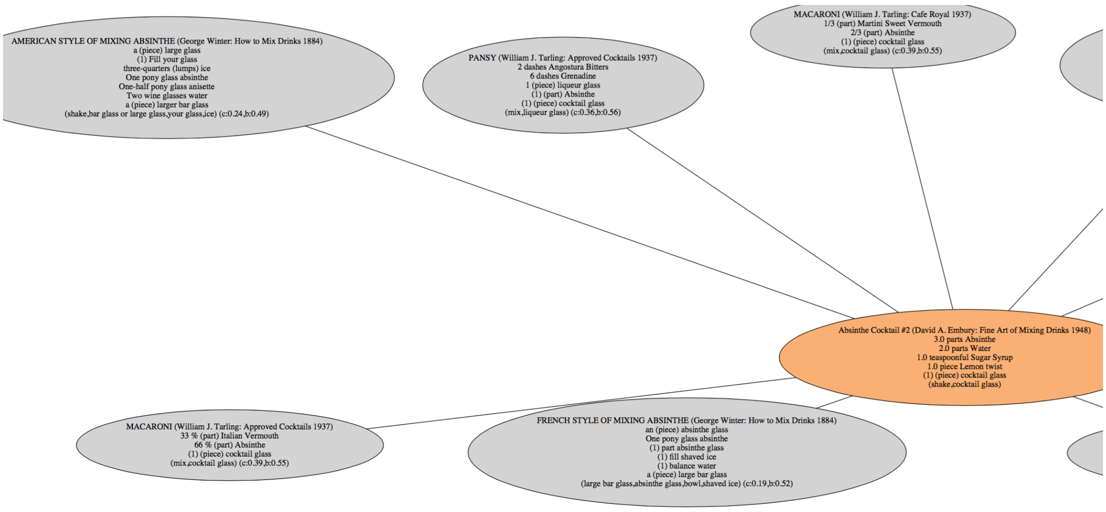

AMERICAN STYLE OF MIXING ABSINTHE (George Winter: How to Mix Drinks 1884)
a (piece) large glass
(1) Fill your glass
three-quarters (lumps) ice
One pony glass absinthe
One-half pony glass anisette
Two wine glasses water
a (piece) larger bar glass
(shake,bar glass or large glass,your glass,ice) (c:0.24,b:0.49)
PANSY (William J. Tarling: Approved Cocktails 1937)
2 dashes Angostura Bitters
6 dashes Grenadine
1 (piece) liqueur glass
(1) (part) Absinthe
(1) (piece) cocktail glass
(mix,liqueur glass) (c:0.36,b:0.56)
MACARONI (William J. Tarling: Cafe Royal 1937)
1/3 (part) Martini Sweet Vermouth
2/3 (part) Absinthe
(1) (piece) cocktail glass
(mix,cocktail glass) (c:0.39,b:0.55)
Absinthe Cocktail #2 (David A. Embury: Fine Art of Mixing Drinks 1948)
3.0 parts Absinthe
2.0 parts Water
1.0 teaspoonful Sugar Syrup
1.0 piece Lemon twist
(1) (piece) cocktail glass
(shake,cocktail glass)
MACARONI (William J. Tarling: Approved Cocktails 1937)
33 % (part) Italian Vermouth
66 % (part) Absinthe
(1) (piece) cocktail glass
(mix,cocktail glass) (c:0.39,b:0.55)
FRENCH STYLE OF MIXING ABSINTHE (George Winter: How to Mix Drinks 1884)
an (piece) absinthe glass
One pony glass absinthe
(1) part absinthe glass
(1) fill shaved ice
(1) balance water
a (piece) large bar glass
(large bar glass,absinthe glass,bowl,shaved ice) (c:0.19,b:0.52)

Figure 4.46.: Recommendation map of absinth

The recommendation for Jack Rose contains (Figure 4.47) a *Applejack Rabbit*, which also contains applejack, maple syrup, and orange juice. This results in a sweet and sour balance, which is appropriate. *Apple Jack* additionally contains raspberry syrup and lemon peel, but this recipe is too obvious. The *Pontoon* also contains rum, peach brandy, and absinth. The result is a stronger balance and different ingredients, which is appropriate.

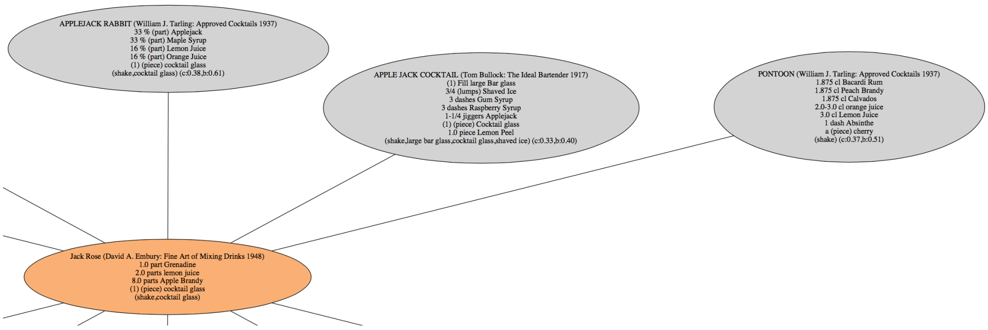
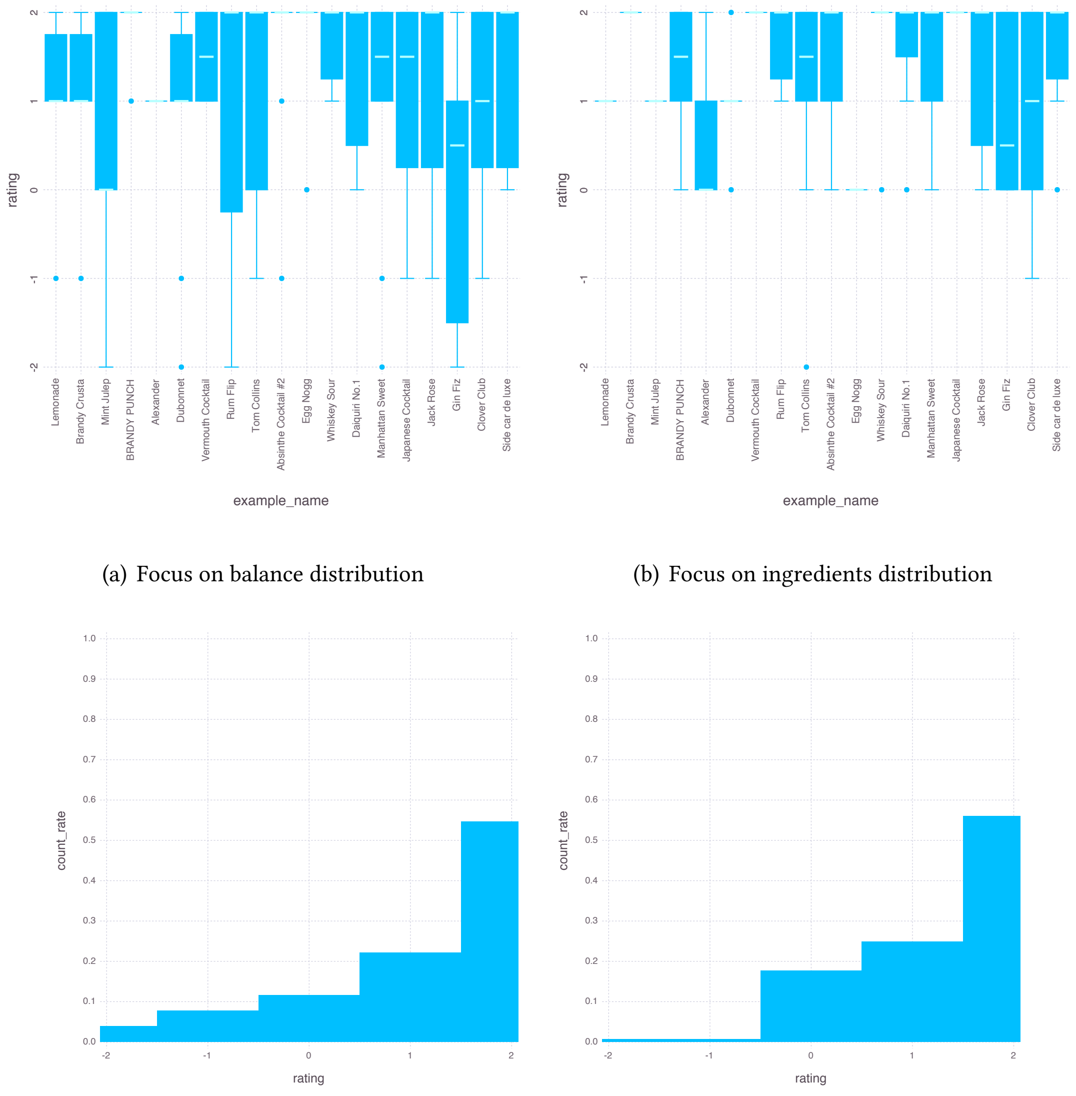


Figure 4.47.: Recommendation map of Jack Rose

The result of the focus on ingredients (Figure 4.48(a)) contains 17 of 19 positive medians where two medians are only 0. Only two classics give negative results. The positive ratings of focus on balance (Figure 4.48(c)) have a rating of about 77 % and focus of ingredients (Figure 4.48(b)) have a rating about 80 %. The negative ratings of focus on balance are about 10 %. The ratings of focus on ingredients are lower than 1 %, but the obvious results concern balance 10 %. The focus on ingredient contains 17 %. Both approaches have a positive result, and the focus on ingredient has fewer outliers but more obvious results. This is because the ingredient balance has to be small, which implies recipes with many shared properties. The focus on balance finds more unexpected results.



(a) Focus on balance distribution

(b) Focus on ingredients distribution

(c) Focus on balance count rates

(d) Focus on ingredients count rates

Figure 4.48.: Result of validation by domain expert Sigurd Sippel

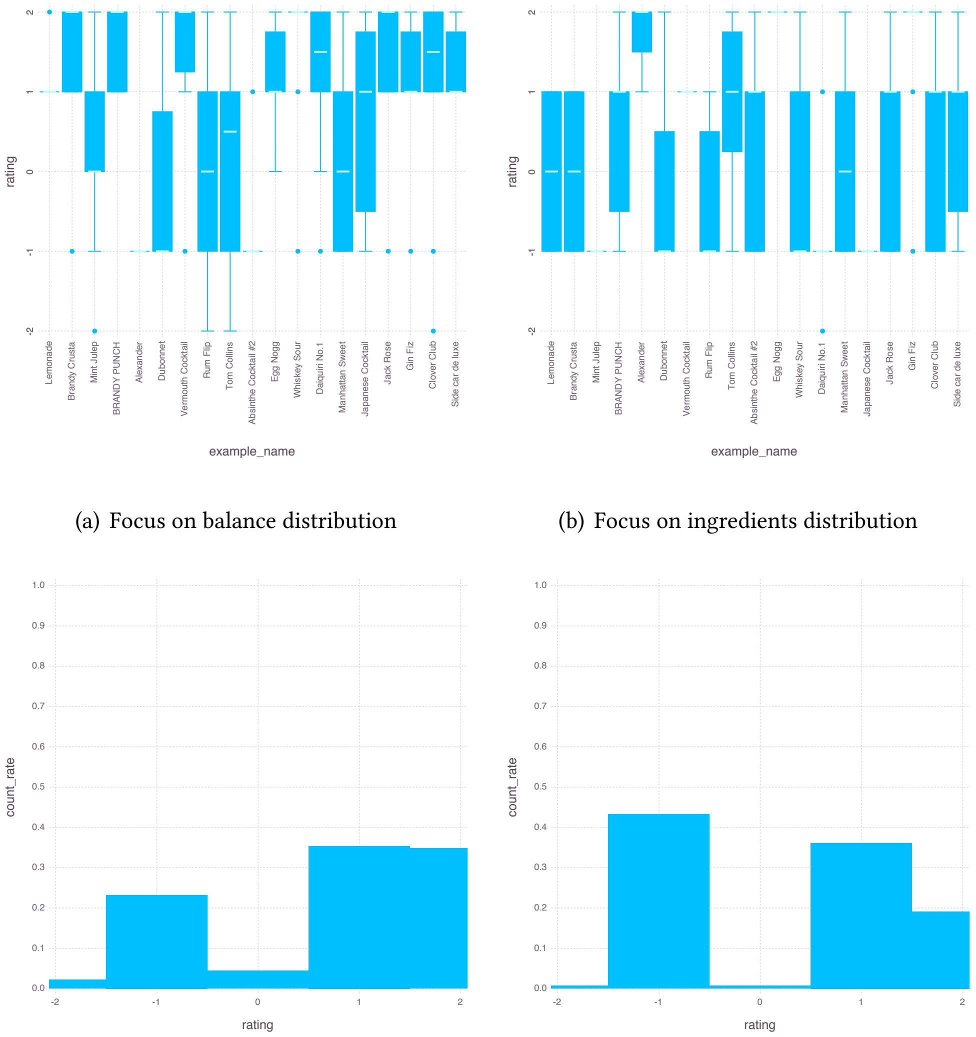
The first proof is subjective but it shows that the process of validation works, the data set is mostly clean and the resulted recommendation are at most as expected. This is a solid basis to share the validation sheet to independent domain experts.

4.5.3. Acceptability of independent domain-experts

Three independent domain experts are interviewed for validation, who rated 966 pairs of favorites and recommendations (Appendix D). They are briefed shortly, which is the idea behind the two approaches of recommendation. They are supposed to use the same numeric scale while creating their own validation criteria. If they use the given criteria, they are not independent.

- A. Schindewolf, 28 years old, physicist, six years of experience in domain
- T. Reuber, 28 years old, building engineer, nine years of experience in domain
- C. Döhren, 43 years old; freelancer for training, university lecturer and gamemaster for business games, 15 years of experience in domain

Domain expert A. Schindewolf validates the recommendations as follows: The result is that 70 % of the recommendations focused on balance are positive, which approximates the first proof. Schindewolf validates more $rather\ appropriate$ and less $appropriate$. Obvious ratings are also fewer than in the first validation, but there are more slightly similar recommendations (Figure 4.49(c)). The difference in the validation shows the cluster-wise analysis (Figure 4.49(a)). The cluster of $Dubonnet$, which contains a wine aperitif, demonstrates: Recommendations with a similar alcohol strength but without a wine aperitif are rated negatively in this validation. In the first proof, these are rated as $rather\ appropriate$. The cluster $Alexander$ contains recipes with high rating of cream, as well as rum and cognac. The recommendation contains no alcohol, which is rated in this validation negatively. In the cluster of $Gin\ Fizz$, the recommendation that differs in $egg\ white$ or $yolk$ are rated as too obvious. The domain expert rates 55 % of the recommendations focused on ingredients as positive and 43 % as only slightly appropriate. The most clusters have a median of ratings in the negative region. There are positive exceptions such as $Egg\ Nogg$ or $Gin\ Fizz$ and each cluster contains positive results. This shows less acceptance compared to the approach that is focused on balance. Specific issues that result in negative ratings are not recognizable. Schindewolf accepts focus on balance more than focus on ingredient.



(a) Focus on balance distribution

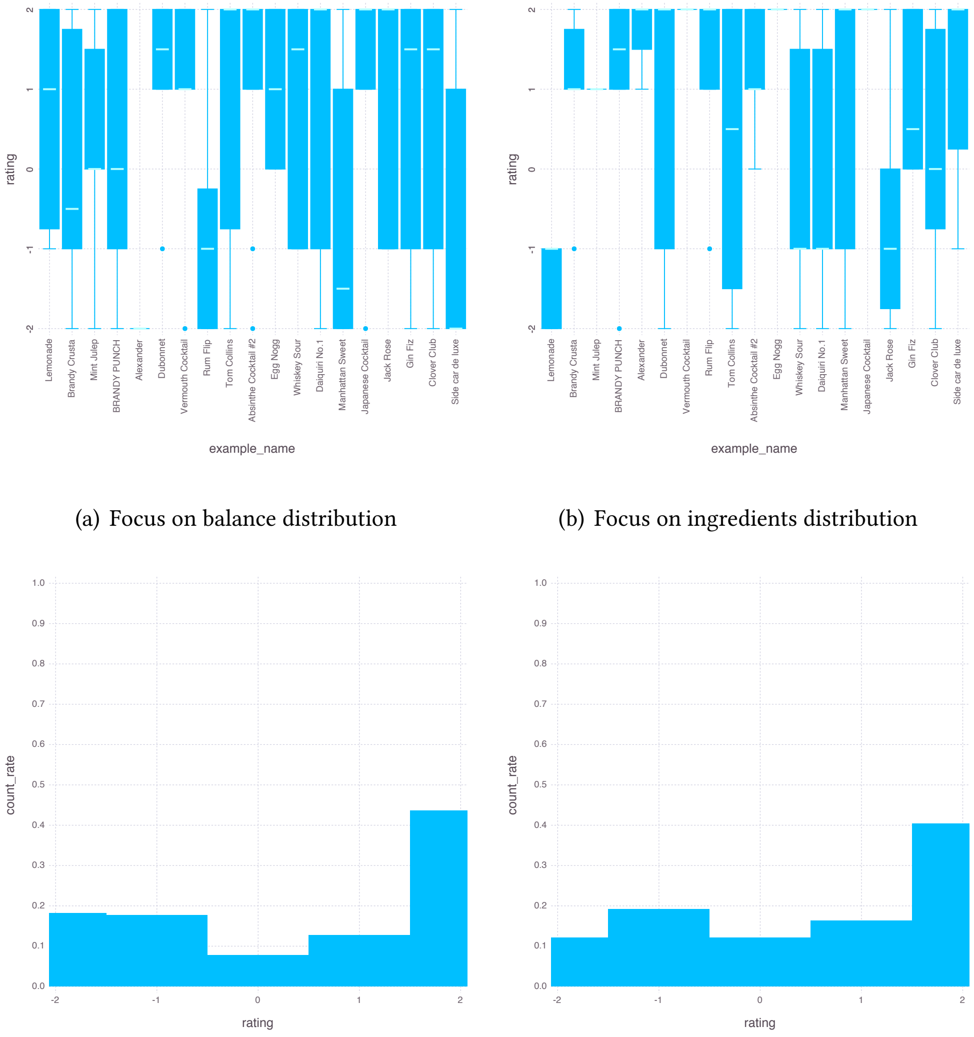
(b) Focus on ingredients distribution

(c) Focus on balance count rates

(d) Focus on ingredients count rates

Figure 4.49.: Result of validation by domain expert A. Schindewolf

Domain expert T. Reuber validates 56 % of the recommendations that are focused on balance as positive (Figure 4.50(c)). The most number of positives are more *appropriate* than *rather appropriate*. 15 out of 19 classic recipes have a positive median of rating (Figure 4.50(a)). Classics such as *Alexander*, *Rum Flip*, and *Manhattan* which have no sourness have the lowest ratings. The ratings of recommendation focused on ingredients are similar. 57 % are positive, but a bit more *rather appropriate* (Figure 4.50(d)). 16 out of 19 classic recipes have a positive median of rating. (Figure 4.50(b)). This results in a positive rating for both types of recommendations, but Reuber accepts fewer recommendations than Schindewolf. The distribution of Reuber's ratings is more widespread than Schindewolf's, showing positive and negative ones for each classic recipe. With the exception of Alexander (focused on balance) and Lemonade (focused on ingredient), there is no classic with failed recommendations. Reuber accepts focus on balance slightly more than focus on ingredient.



(a) Focus on balance distribution

(b) Focus on ingredients distribution

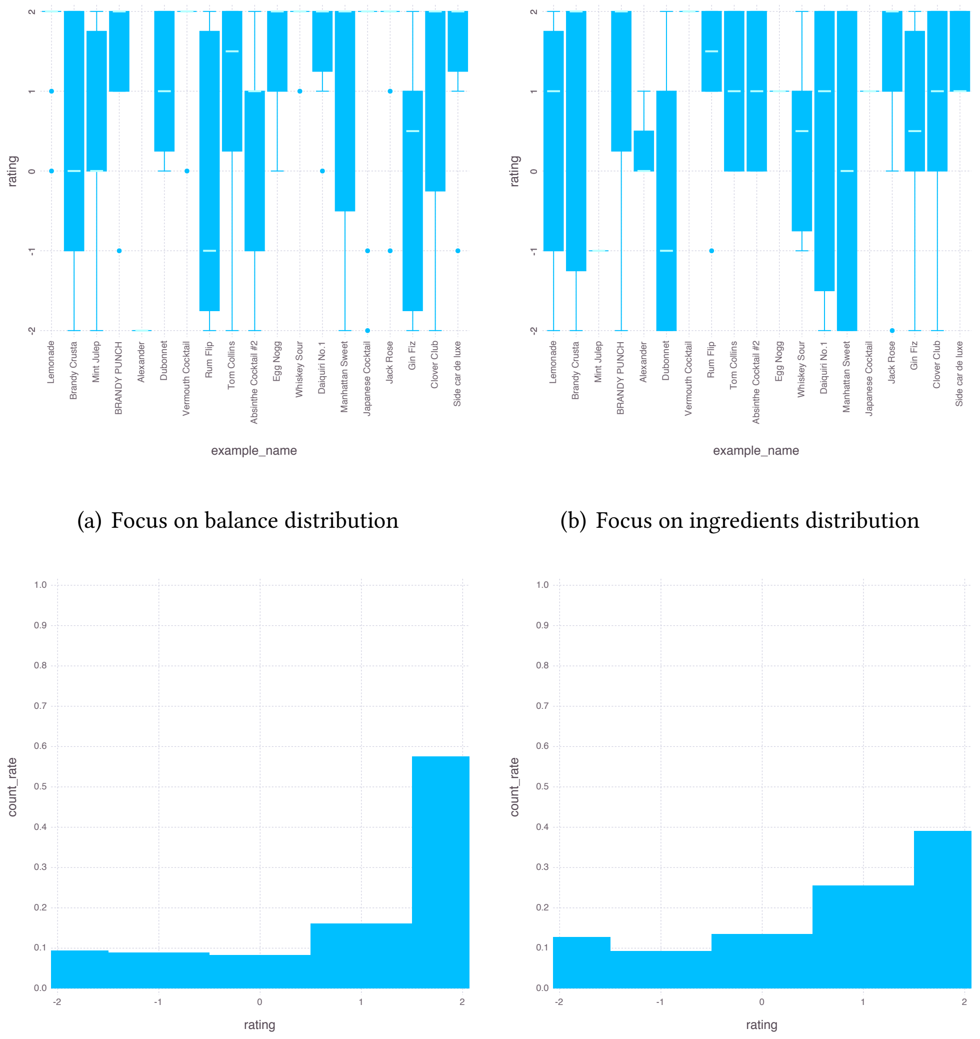
(c) Focus on balance count rates

(d) Focus on ingredients count rates

Figure 4.50.: Result of validation by domain expert T. Reuber

Domain expert C. Döhren validates 73 % of the recommendations that are focused on balance as positive (Figure 4.51(c). The most positively validated recommendations are *appropriate*. 15

out of 19 clusters have a positive median of rating (Figure 4.51(a)). A negative exception is the *Rum Flip*. The focus of ingredient (Figure 4.51(b)) has 65 % positive ratings and more *rather appropriate* than *appropriate* ratings. 14 out of 19 clusters (Figure 4.51(b)) have a positive median. Döhren accepted the focus of balance approach more than the focus of ingredient approach.



(a) Focus on balance distribution

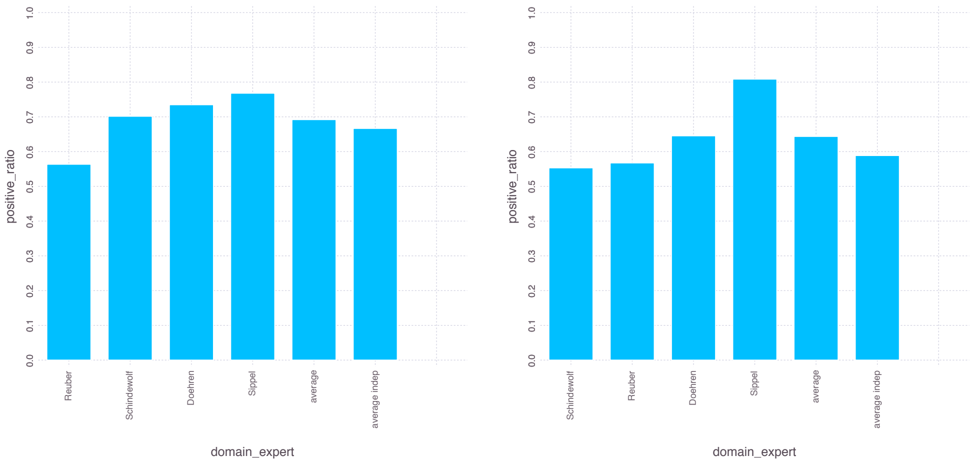
(b) Focus on ingredients distribution

(c) Focus on balance count rates

(d) Focus on ingredients count rates

Figure 4.51.: Result of validation by domain expert C. Döhren

The rate of positive ratings of focus on balance (Figure 4.52(a)) is on average about 69 %. The dependent domain expert Sippel gives the highest ratings, but the independent average value excluding Sippel's ratings has a value of about 67 %, which is very close to that. The ratio of positive rating in the focus of ingredients (Figure 4.52(b)) is about 64 %. However, the independent average is only 59 %. This shows less acceptance of this approach as well as fewer objective ratings of Sippel.

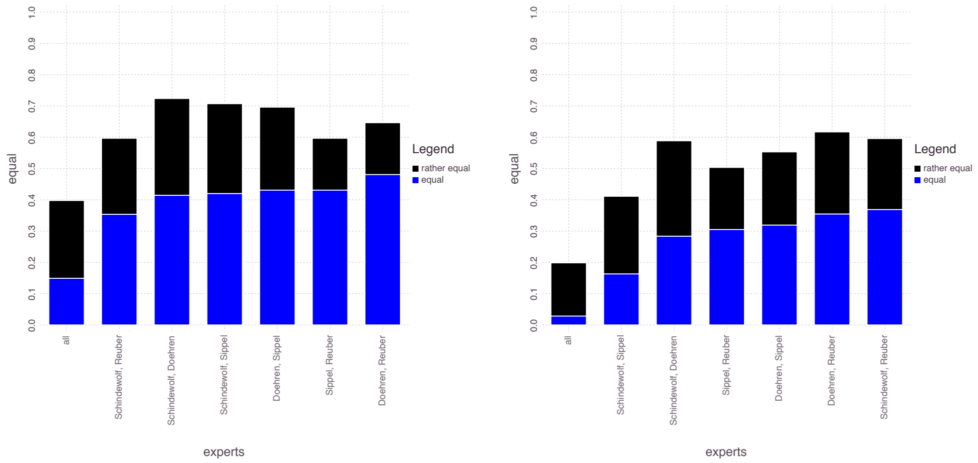


(a) Positive ratings with focus on balance

(b) Positive ratings with focus on ingredients

Figure 4.52.: Positive ratings of domain experts

Finally, the consensus of all domain experts including the first proof is assumed to get a feeling how objective the rating of a recipe recommendation is. To measure the consensus, two metrics are used: The first, called *equal*, calculates the ratio of the pair of favorites and recommendations that are rated equally by domain experts. The second is called *rather equal* and considers only positive, obvious, and negative ratings. Positive ratings are *rather appropriate* and *appropriate*, while negative are *slightly similar* and *unacceptable*.



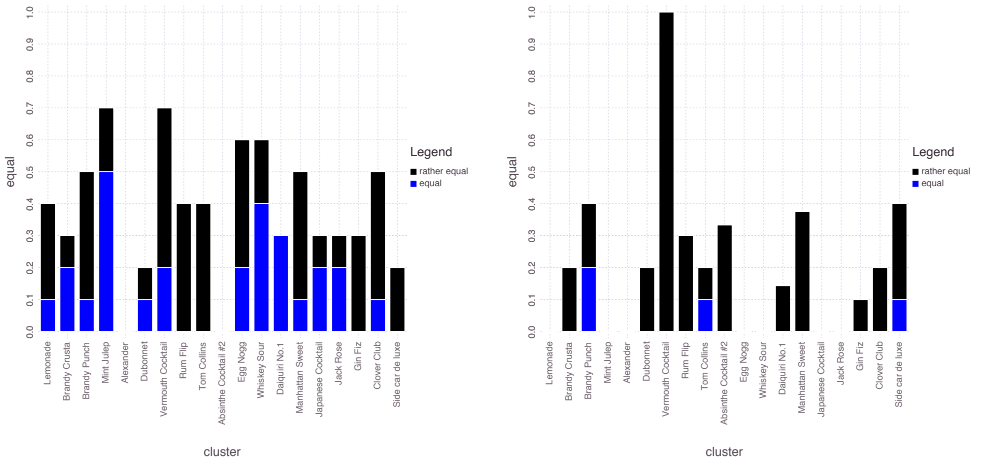
(a) Consensus in focus on balance

(b) Consensus in focus on ingredients

Figure 4.53.: Consensus of domain experts

The focus on balanced recommendation pairs of the domain experts shows *equals* ratings between 36 % and 48 %; *rather equal* ratings are between 60 % and 70 % (Figure 4.53(a)). If all domain experts are considered *equal*, the rating is about 15 % and *rather equal* ratings are about 39 %. For the classic recipe *Mint Julep*, the *equal* rating has the highest rate of about 60 % (Figure 4.54(a)). They mostly have *equal* ratings between 10 % and 40 %. The Alexander with only one recommendation has no equal ratings; the Absinthe cocktail too has no equal ratings. Hence, there is no consensus. For the other classic recipes have *rather equal* ratings between 20 % and 70 %. For the focus on ingredients, pairs of domain experts rate recommendation between 16 % and 37 % *equal* and between 41% and 61 % *rather equal* (Figure 4.53(b)). The *equal* rating is about 3 % and *rather equal* is about 20 %. The *equal* rating considered for each classic recipe is very low (Figure 4.54(b)); only for fruity *Brandy Punch*, the consensus is measurable. The vermouth cocktail has a very high consensus at about 100 %. A reason could be that this recipe represents especially one ingredient: The vermouth indicates a favorite ingredient rather than a favorite recipe.

The recommendation focused on balance has a clearly higher consensus than the recommendation focused on ingredients. This consensus indicates that there are a ratio of objective rating.



(a) Consensus for classics in focus on balance (b) Consensus for classics in focus on ingredients

Figure 4.54.: Consensus of domain experts for cluster of classics

The domain experts need 3–4 hours to fill the rating sheet, which shows how time consuming the knowledge elicitation of domain expert is. The domain experts give feedback that there recommendations which are appropriate to the favorite, but they would not recommend that because the recipe itself was not persuasive for their expectations of quality. Therefore, a quality measurement is necessary to increase the precision of recommendation.

Summary

The recommendation based on the nearest-neighbor approach and a path-based similarity measurement comprises outliers, but every example gets at least one and more recommendation, which is appropriate or almost appropriate. The validation by domain experts shows that the approach focused on ingredients is less acceptable but that the approach focused on balance has huge acceptability and consensus of rating by all domain experts (challenge four). This result qualifies the recommendation focused on balance for validation with more domain experts such as in an online study. The recommendation focused on ingredients needs higher precision. A replacement of a favorite recipe with favorite ingredient is an opportunity that needs to be proved.

4.6. Architecture

The experiment described earlier resulted in the following architecture. The experimental platform contains several components (Figure 4.55) — The extraction converts raw sources into a clean and normalized data set. It is separated into a parser for cocktail recipes and a parser for cocktail books containing a collection of recipes. The cocktail component present data structure to store recipes and offers a plausibility check for recipes. The category component helps to find ontology items with a raw string and a chosen taxonomy such as ingredients, preparations, glassware, or units. The quantity component offers unit conversion and default quantities and reference resolving. The assignment component provides substitution and merging mechanisms. The quantity component provides data structures to present units and converting units. The balance component resolves the balance of a ingredient or a recipe by ontology.

The input data is a collection of unstructured recipes. Thus, the feature extraction is a sophisticated process that has to work once for each recipe. The process is independent of recommendation, a fact that extracts the features and stores the retrieved data in an extended XML format.

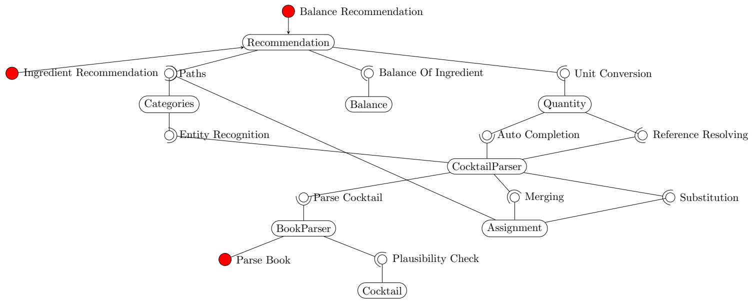


Figure 4.55.: Components

The ontology is used to identify the named entities. Indexing is necessary to find named entities or part of that with high performance. Apache Jena 2.13.0[9] is used to call SPARQL queries on the RDF ontology. The module Apache Jena Text 1.1.2[10] is used to integrate Apache Lucene 4.6.1[11] as an indexer in a SPARQL query. Scala parser combinators 1.0.3[12] are used for context analysis. The library itext 5.06[13] is used to transform the PDF format into plain text.

[9] https://jena.apache.org

[10] https://jena.apache.org/documentation/query/text-query.html

[11] https://lucene.apache.org/core

[12] https://github.com/scala/scala-parser-combinators

[13] http://itextpdf.com

4.7. Summarization and evaluation of experiments

This experiments investigates the development process of a domain-specific recommendation system where cocktail recipes serve as an example domain. The process starts with an interview of domain experts and manual analysis of real data, followed by an automatic feature extraction. After the preprocessing of raw text, the KDD process is applied to select the data and process and transform it to the needed feature vector. The extraction is developed through tests that give a plausible result for most of the recipes of the used historic cocktail books. The needed knowledge is modeled along the concept of basic categories. The necessary categories, especially ingredients, are successfully mapped to definite categories, because the domain-dependent entities such as ingredients (products or abstract names) are generally unique. If data is missing in ontology, it probably leads to poor parsing results. Definite categories are needed for the nearest-neighbor classification, because the distance measurement is based on paths through the categories. The path has to present a significant distance for a precise recommendation which have to be continuous maintained by domain expert.

This data processing is a core aspect of a content-based recommendation system, which uses a profile modeling of a single favorite recipe. The selection of the favorite works as a simple way to describe what one likes. That is also necessary for validation. The validation shows a measurable acceptability to independent domain experts. Apart from optimization opportunities, the validation shows that the used process, started with the interview of domain experts and carried out though the extraction of appropriate knowledge, is functional. The ground truth for the domain of cocktail is that the quantified ingredients are the main parts of a recipe. A process of preparation such as for a cooking recipes is not necessary. If these cocktail recipes are modeled in an abstract way of sensations such as flavors this model renders recipes comparable.

From the domain perspective, there are the opportunities for future work are as follows: A intensive online study will show how generally valid the actual validation results are. Regardless of the bartender's knowledge, not each existing recipe results in a high-quality cocktail. Quality measuring is needed for a further optimization for the recommendation process. The user model is extensible with further favorites or dislikes, in order to get a higher precision of recommendation. Each balance of ingredients is actually defined qualitatively. Therefore, the precision has to prove — and possibly increase — either with additional validation of domain experts or an alternative modeling of other sensations such as fruity or refreshing. A quantitative definition is an alternative solution, which uses declaration in an already defined unit such Scoville for sharpness. To maintain and enrich the ontology, additional sources of data have to be added to find new product names and new other categories. The YAGO process gives motivation for such processing. The types of recipes are actually restricted to cold and mostly liquid cocktails. If the experiments are extended to hot and drinks with foams and drops (molecular mixology), the model can be validated to a more generalized category of mixed drinks. Modern recipes contain more specific ingredients; the mapping to definite basic categories has to be proved again.

Assumed a huge database of recipes is given changes in process of time should be proved. This is a basis for analysis of which kinds of ingredient or recipe will be the trend of tomorrow. Assuming precise recommendations are available, a kind of meal planning is a research opportunity: An alternative to a given favorite is recommended. The transferability of meal planning of cooking recipes to cocktail recipes should be proved, which means recommending a follower of a given drink to plan the time of a guest in bar. Considering it can be measured how ingredients work together, the automatic creation of new cocktail recipes should be proved.

In review of this experiments the ground truth about a domain-specific recommender system is that the main interest of the user has to be in focus: To arouse the user's interest, it is necessary to find something known such as parts of a defined favorite and understand it in deeper way. The understanding is used to find something new. The modeling for such interests has to be according to the domain. Interviewing domain experts is a necessary precondition for extracting an abstract model. Extraction of this model in test-driven development is recommended to keep implicit requirements in mind during the whole process of development. Testing of data using plausibility metric against separate high-quality data from wasted data is necessary for obtaining precise recommendation results. Understanding of distances by coherence and distinction measurement are useful tools to get a feeling of data and distances. These have to be proceeded successfully before a validation of the recommendation by domain experts. A validation needs a lot of feedback from domain experts but it shows how acceptable this recommendation is. The personal opinion has to be dismissed to get a useful result, therefore domain expert have to evaluate the acceptance and not whether it is equal to its own chosen recommendation. If this steps are performed, then the validation will give a meaningful measurement of the quality of recommendation.

5. Conclusion and future work

Recommender systems are currently gathering a lot of attention in various areas of computer science and economy. The most common approaches are the user-oriented ones, such as collaborative filtering. However, these are not as relevant to domain-specific questions. Therefore there is a need for research to increase the precision of domain-specific recommendations. Domain-specific recommendation also requires a distance between items. Item-based distances in turn need knowledge. Now, the process of knowledge elicitation is very expensive. The process considered in this thesis shows the way to understand and extract domain-specific content to get recommendations that are accepted by independent domain experts. Therefore, it is a contribution toward the holistic approach to increase the performance of recommender systems. This thesis shows that the effort needed for deep understanding is high, but it is also worthwhile, because the quality of recommendation is very positive.

Nevertheless, there are further research questions, because the understanding of knowledge has to be automated as far as possible to find a development process that is as cheap as possible. Only then will domain-specific recommendation be used in a broad bandwidth of domains and software solutions.

There are two aspects involved. The first is to understand the key aspects of a domain. It should be shown how the content of the world wide web is usable. In particular, the content of social networks where potential domain experts can be found should be investigated. This is a uncertain area. Therefore, to deal with such users, the trustworthiness classification of a domain expert should be verified. The published content of trustworthy domain experts can be used to understand what they are interested in and what is important for them. The second aspect is to use this extracted knowledge to understand the domain-specific content. Web content is mostly domain-specific, but it is also semi-structured; the interesting features have to extracted. Therefore, it is not difficult to find content, but it is difficult to deeply understand the content. A further research question is: How is it possible to automate this process of extracting domain-specific features and validating them by using knowledge of domain experts.

A recommender system usually provides more than one recommendation because there is a greater chance to find an appropriate recommendation if a list of recommendations is offered.

The precision of a domain-specific recommendation system depends on the knowledge. If the knowledge is not only a extract of the domain but rather covers the whole area of domain, then the recommendation system can explicitly classify one recommendation with an adequate precision. In this case, the recommendation system is a digital servant. In further steps, it should be shown how it is possible to find and measure a recommendation with such high precision.

A. Survey

A.1. How old are you? (a rounded number)

_____

A.2. Why are you interested in cocktails? (2: I agree, 1: I agree rather, 0: I'm undecided, -1: I don't agree rather, -2: I don't agree)

- (-2,-1,0,1,2) I am an employee/owner of a bar.
- (-2,-1,0,1,2) I am an employee/owner of a restaurant.
- (-2,-1,0,1,2) I am a connoisseur.
- (-2,-1,0,1,2) I am a guest in a bar.

A.3. For how many years have you been interested in cocktails? (a rounded number)

_____

A.4. Which information should a simple recipe contain to be usable by an experienced bartender? (2: I agree, 1: I agree rather, 0: I'm undecided, -1: I don't agree rather, -2: I don't agree)

- (-2,-1,0,1,2) Cocktail name
- (-2,-1,0,1,2) Ingredient with quantity declaration

- (-2,-1,0,1,2) Alternative ingredients
- (-2,-1,0,1,2) Optional ingredients
- (-2,-1,0,1,2) Preparations (such as shake)
- (-2,-1,0,1,2) Categories of glassware
- (-2,-1,0,1,2) Information about ice
- (-2,-1,0,1,2) Recipe author
- (-2,-1,0,1,2) Year of creation
- (-2,-1,0,1,2) History of the recipe
- (-2,-1,0,1,2) Anecdote
- (-2,-1,0,1,2) Tips / best practice

A.5. How important is the style of preparation for the result, i.e. the cocktail?

- (-2,-1,0,1,2) The style of preparation is not relevant for the result.
- (-2,-1,0,1,2) The temperature and melting water are relevant, and both are indirectly controlled by the style of preparation. The bartender decides in individual cases whether he should confirm the recommendation.
- (-2,-1,0,1,2) The preparation must clarify how to implement the recipe.

A.6. If you prepare a given recipe, how important is information about the use of ice? (2: I agree, 1: I agree rather, 0: I'm undecided, -1: I don't agree rather, -2: I don't agree)

- (-2,-1,0,1,2) I always follow the instructions.
- (-2,-1,0,1,2) It is only a recommendation. I decide because of the characteristic of a cocktail.

- (-2,-1,0,1,2) I always use the same type of ice, because I know how to get the best cooling.
- (-2,-1,0,1,2) Instructions about ice are usually not necessary.

A.7. You prepare a cocktail based on a given recipe, but the recipe contains ingredients you do not have. How appropriate are these suggestions? (2: I agree, 1: I agree rather, 0: I'm undecided, -1: I don't agree rather, -2: I don't agree)

- (-2,-1,0,1,2) I can prepare a cocktail if I have all the ingredients.
- (-2,-1,0,1,2) I cannot have all ingredients of the world. I adopt the recipe due to its advantage. But I can only substitute ingredients with an appropriate alternative.
- (-2,-1,0,1,2) I cannot have all ingredients of the world. I adopt the recipe due to its advantage. I substitute at whim.

A.8. A recipe contains genever (more than a drop). How appropriate are these suggestions? (2: I agree, 1: I agree rather, 0: I'm undecided, -1: I don't agree rather, -2: I don't agree)

- (-2,-1,0,1,2) I could use any ingredient that is something like genever. Probably one of them is better, but the recipe does not help me. It is my personal decision.
- (-2,-1,0,1,2) Any ingredient that is distilled with similar ingredients can be used. Therefore, any kind of gin can be used.
- (-2,-1,0,1,2) It is a distilled ingredient. Any distilled ingredient can be used.
- (-2,-1,0,1,2) The recipe is imprecise. I need a concrete product to prepare it.

A.9. A recipe contains a concrete genever product (more than a drop), but you do not have this concrete product. How appropriate are these suggestions? (2: I agree, 1: I agree rather, 0: I'm undecided, -1: I don't agree rather, -2: I don't agree)

- (-2,-1,0,1,2) The ingredients could be very special. I always need the concrete product to prepare it.
- (-2,-1,0,1,2) The ingredients could be special, but if I do not know if it is special for this concrete product, I would substitute it. I could use any ingredient that is something like genever. Probably one is better than others. It is my personal decision.
- (-2,-1,0,1,2) Any ingredient that is distilled with similar ingredients can be used. Therefore, any kind of gin can be used.
- (-2,-1,0,1,2) It is a distilled ingredient. Any distilled ingredient can be used.

A.10. How important is the quantity declaration for you? (2: I agree, 1: I agree rather, 0: I'm undecided, -1: I don't agree rather, -2: I don't agree)

- (-2,-1,0,1,2) I take the quantity declaration of one ingredient literally.
- (-2,-1,0,1,2) I use the quantity declaration to understand the recipe, but I adopt the quantity to ingredients and temper. The characteristics of the cocktail recipe will always be preserved.
- (-2,-1,0,1,2) I use quantity declarations, but most of the declarations are precise (a lemon, dash bitter, filling with soda water, etc.). Therefore, even if it is imprecise, I have to make my own decision.
- (-2,-1,0,1,2) I ignore the quantity.

A.11. Which glasses are appropriate for following recipe? (2: I agree, 1: I agree rather, 0: I'm undecided, -1: I don't agree rather, -2: I don't agree)

DAIQUIRI (William J. Tarling: Cafe Royal 1937)
3 dashes Gomme Syrup
3/4 Daiquiri Rum
1/4 Juice of a Lime or Lemon

- (-2,-1,0,1,2) Cocktail glass
- (-2,-1,0,1,2) Wine glass
- (-2,-1,0,1,2) Champagne flute
- (-2,-1,0,1,2) Champagne saucer
- (-2,-1,0,1,2) Whiskey tumbler
- (-2,-1,0,1,2) Highball glass
- (-2,-1,0,1,2) Collins glass
- (-2,-1,0,1,2) Shot glass
- (-2,-1,0,1,2) Goblet
- (-2,-1,0,1,2) Beer glass
- (-2,-1,0,1,2) Silver cup

A.12. How do you choose your glassware? (2: I agree, 1: I agree rather, 0: I'm undecided, -1: I don't agree rather, -2: I don't agree)

- (-2,-1,0,1,2) I choose it based on the characteristic.
- (-2,-1,0,1,2) I choose it based on the volume of the cocktail.
- (-2,-1,0,1,2) I choose it based on my personal opinion.
- (-2,-1,0,1,2) I choose it based on my personal situation (weather, location, etc.)

A.13. Which of the following preparations would you use? (2: I agree, 1: I agree rather, 0: I'm undecided, -1: I don't agree rather, -2: I don't agree)

DAIQUIRI (William J. Tarling: Cafe Royal 1937)
3 dashes Gomme Syrup
3/4 Daiquiri Rum
1/4 Juice of a Lime or Lemon

- (-2,-1,0,1,2) shake
- (-2,-1,0,1,2) stir
- (-2,-1,0,1,2) build
- (-2,-1,0,1,2) mix
- (-2,-1,0,1,2) float
- (-2,-1,0,1,2) boil
- (-2,-1,0,1,2) bake
- (-2,-1,0,1,2) steam

A.14. How do you justify your decision about preparation? (2: I agree, 1: I agree rather, 0: I'm undecided, -1: I don't agree rather, -2: I don't agree)

- (-2,-1,0,1,2) I choose it based on the characteristic.
- (-2,-1,0,1,2) I choose it based on my personal opinion.
- (-2,-1,0,1,2) I choose it based on my personal situation (weather, location, etc.)

A.15. What volume would you like to have the cocktail (rounded, in cl, without melting water)

_____

A.16. How do you categorize the recipe? (2: I agree, 1: I agree rather, 0: I'm undecided, -1: I don't agree rather, -2: I don't agree)

- (-2,-1,0,1,2) I decide based on the characteristic.
- (-2,-1,0,1,2) I decide based on my personal opinion.
- (-2,-1,0,1,2) I decide it based on my personal situation (weather, location, etc.).
- (-2,-1,0,1,2) I decide based on the volume of the glass.

A.17. If you were asked for a cocktail recommendation, which information would be necessary for your decision? (2: I agree, 1: I agree rather, 0: I'm undecided, -1: I don't agree rather, -2: I don't agree)

- (-2,-1,0,1,2) Preferences of the guest
- (-2,-1,0,1,2) Atmosphere
- (-2,-1,0,1,2) Weather/Temperature
- (-2,-1,0,1,2) Location
- (-2,-1,0,1,2) Costs of the cocktail

A.18. How often do you use the following sources of information? (2: I agree, 1: I agree rather, 0: I'm undecided, -1: I don't agree rather, -2: I don't agree)

- (-2,-1,0,1,2) Books
- (-2,-1,0,1,2) Blogs
- (-2,-1,0,1,2) Online cocktail databases
- (-2,-1,0,1,2) Your own list of collected cocktail recipes

- (-2,-1,0,1,2) Your experiences (in your brain)

A.19. Assume that the preferences are described by favorite cocktails. You know your guest likes a daiquiri, but he wants an alternative. What would you recommend? (Cocktail name, main ingredients in short)

____

A.20. How much influence do the elements of the favorite recipe (such as daiquiri) have on the recommendation? (2: I agree, 1: I agree rather, 0: I'm undecided, -1: I don't agree rather, -2: I don't agree)

- (-2,-1,0,1,2) Cocktail name
- (-2,-1,0,1,2) Ingredient declaration including quantity
- (-2,-1,0,1,2) Alternative ingredients
- (-2,-1,0,1,2) Optional ingredients
- (-2,-1,0,1,2) Preparation
- (-2,-1,0,1,2) Glassware
- (-2,-1,0,1,2) Type of ice
- (-2,-1,0,1,2) Recipe authors
- (-2,-1,0,1,2) Recipe year
- (-2,-1,0,1,2) Recipe history
- (-2,-1,0,1,2) Anecdote
- (-2,-1,0,1,2) Tips/best practice

A.21. How do the individual flavors of the favorite (such as a daiquiri) influence the recommendation? (2: I agree, 1: I agree rather, 0: I'm undecided, -1: I don't agree rather, -2: I don't agree)

- (-2,-1,0,1,2) Sweetness
- (-2,-1,0,1,2) Sourness (lemon, lime, etc.)
- (-2,-1,0,1,2) Bitterness
- (-2,-1,0,1,2) Creaminess
- (-2,-1,0,1,2) Sharpness
- (-2,-1,0,1,2) Smokiness
- (-2,-1,0,1,2) Peatiness
- (-2,-1,0,1,2) Water/dilution (soda, also part of tonic water, etc.)
- (-2,-1,0,1,2) Ratio of alcohol

A.22. If your guest did not accept your recommendations, how much would you prefer a recommendation service that recommends recipes based on favorites. (2: I agree, 1: I agree rather, 0: I'm undecided, -1: I don't agree rather, -2: I don't agree)

- (-2,-1,0,1,2) I would try it, but I am skeptical about the results.
- (-2,-1,0,1,2) I would never try it, because it cannot work.
- (-2,-1,0,1,2) I would use it if it were according to my personal situation (weather, location).
- (-2,-1,0,1,2) I would use it privately.
- (-2,-1,0,1,2) I would use it commercially (bar, restaurant).
- (-2,-1,0,1,2) I would use it commercially (other business).

B. Domain-specific clusters

B.1. lemonade.xml

```
<cocktails>
    <cocktail><title>Soda cocktail</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>teaspoon</unit>
            </quantity><name>white sugar</name></ingredient>
            <ingredient><quantity><value>2</value><unit>dashes</unit>
            </quantity><name>bitters</name></ingredient>
            <ingredient><quantity><value>1</value><unit>bottle</unit>
            </quantity><name>soda</name></ingredient>
            <ingredient><quantity><value>1</value><unit></unit>
            </quantity><name>lemon slice</name></ingredient>
        </ingredients>
        <preparation>stir</preparation>
        <glass>soda glass</glass>
        <book><title>How to mix Drinks</title><author>Jerry Thomas</author>
            <published>1862</published></book>
    </cocktail>
    <cocktail><title>Lemonade</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>tablespoon</unit>
            </quantity><name>sugar</name></ingredient>
            <ingredient><quantity><value>2</value><unit></unit>
            </quantity><name>fruit in season</name></ingredient>
            <ingredient><quantity><value>1</value><unit>bottle</unit>
            </quantity><name>soda</name></ingredient>
            <ingredient><quantity><value>1</value><unit></unit>
            </quantity><name>lemon slice</name></ingredient>
        </ingredients>
        <preparation>stir</preparation><glass>large bar glass</glass>
        <book><title>Bartenders Manual</title><author>Harry Johnson</author>
            <published>1882</published></book>
    </cocktail>
    <cocktail><title>Soda Cocktail #2</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>lump</unit>
            </quantity><name>sugar</name></ingredient>
            <ingredient><quantity><value>4</value><unit>dashes</unit>
            </quantity><name>Angostura</name></ingredient>
            <ingredient><quantity><value>1</value><unit>wineglass</unit>
            </quantity><name>soda</name></ingredient>
            <ingredient><quantity><value>1</value><unit></unit>
            </quantity><name>lemon juice</name></ingredient>
        </ingredients>
        <preparation>stir</preparation><glass>long tumbler</glass>
        <book><title>Savoy Cocktail Book</title><author>Harry Craddock</author>
            <published>1930</published></book>
    </cocktail>
</cocktails>
```

B.2. crusta.xml

```
<cocktails>
    <cocktail><title>Brandy Crusta</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>part</unit>
            </quantity><name>Brandy</name></ingredient>
            <ingredient><quantity><value>1.5</value><unit>dashes</unit>
            </quantity><name>maraschino</name></ingredient>
            <ingredient><quantity><value>1</value><unit></unit>
            </quantity><name>orange twist</name></ingredient>
        </ingredients>
        <preparation>stir</preparation><glass>wine glass</glass>
        <book><title>Fine Art of Mixing Drinks</title><author>David A. Embury</author>
            <published>1948</published></book>
    </cocktail>
    <cocktail><title>Brandy Crusta #2</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>dash</unit>
            </quantity><name>lemon juice</name></ingredient>
            <ingredient><quantity><value>1</value><unit></unit>
            </quantity><name>lemon peel</name></ingredient>
            <ingredient><quantity><value>1</value><unit>tablespoon</unit>
            </quantity><name>white sugar</name></ingredient>
            <ingredient><quantity><value>2</value><unit>oz</unit>
            </quantity><name>brandy</name></ingredient>
        </ingredients>
        <preparation>stir</preparation>
        <glass>wine glass</glass>
        <book><title>How to mix Drinks</title><author>Jerry Thomas</author>
            <published>1862</published></book>
    </cocktail>
    <cocktail><title>Brandy Crusta #3</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>wineglass</unit>
            </quantity><name>brandy</name></ingredient>
            <ingredient><quantity><value>1</value><unit>dash</unit>
            </quantity><name>Boker’s bitters</name></ingredient>
            <ingredient><quantity><value>4.5</value><unit>drops</unit>
            </quantity><name>lemon juice</name></ingredient>
            <ingredient><quantity><value>1</value><unit></unit>
            </quantity><name>lemon peel</name></ingredient>
            <ingredient><quantity><value>2</value><unit>dashes</unit>
            </quantity><name>maraschino</name></ingredient>
            <ingredient><quantity><value>3.5</value><unit>dashes</unit>
            </quantity><name>orchard syrup</name></ingredient>
        </ingredients>
        <preparation>stir</preparation>
        <glass>wine glass</glass>
        <book><title>Bartenders Manual</title><author>Harry Johnson</author>
            <published>1882</published></book>
    </cocktail>
</cocktails>
```

B.3. brandypunch.xml

```
<cocktails>
    <cocktail><title>Brandy Punch</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>teaspoon</unit>
            </quantity><name>raspberry Syrup</name></ingredient>
            <ingredient><quantity><value>2</value><unit>tablespoons</unit>
            </quantity><name>white sugar</name></ingredient>
            <ingredient><quantity><value>1</value><unit>wineglass</unit>
            </quantity><name>water</name></ingredient>
            <ingredient><quantity><value>1.5</value><unit>wineglasses</unit>
            </quantity><name>brandy</name></ingredient>
            <ingredient><quantity><value>1</value><unit>teaspoon</unit>
            </quantity><name>orange</name></ingredient>
            <ingredient><quantity><value>1</value><unit>teaspoon</unit>
            </quantity><name>pineapple</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>tumbler</glass>
    </cocktail>
    <cocktail>
        <title>Mississippi Punch</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>wineglass</unit>
            </quantity><name>brandy</name></ingredient>
            <ingredient><quantity><value>0.5</value><unit>wineglasses</unit>
            </quantity><name>jamaica rum</name></ingredient>
            <ingredient><quantity><value>0.5</value><unit>wineglasses</unit>
            </quantity><name>bourbon</name></ingredient>
            <ingredient><quantity><value>1.5</value><unit>tablespoons</unit>
            </quantity><name>white sugar</name></ingredient>
            <ingredient><quantity><value>2</value><unit>cl</unit>
            </quantity><name>lemon juice</name></ingredient>
            <ingredient><quantity><value>0.5</value><unit>wineglass</unit>
            </quantity><name>Water</name></ingredient>
            <ingredient><quantity><value>1</value><unit>piece</unit>
            </quantity><name>orange</name></ingredient>
            <ingredient><quantity><value>1</value><unit></unit>
            </quantity><name>berries</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>tumbler</glass>
        <book><title>How to mix Drinks</title><author>Jerry Thomas</author>
            <published>1862</published></book>
    </cocktail>
</cocktails>
```

B.4. julep.xml

```
<cocktails>
    <cocktail><title>Real Georgia Mint Julep</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>teaspoon</unit>
            </quantity><name>white powdered sugar</name></ingredient>
            <ingredient><quantity><value>3</value><unit>quarters</unit>
            </quantity><name>cognac</name></ingredient>
            <ingredient><quantity><value>12</value><unit>sprigs</unit>
            </quantity><name>mint</name></ingredient>
        </ingredients>
        <preparation>stir</preparation><glass>tumbler</glass>
        <book><title>How to mix Drinks</title><author>Jerry Thomas</author>
            <published>1862</published></book>
    </cocktail>
    <cocktail><title>Mint Julep</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>teaspoon</unit>
            </quantity><name>white pulverized sugar</name></ingredient>
            <ingredient><quantity><value>1.5</value><unit>wineglasses</unit>
            </quantity><name>cognac</name></ingredient>
            <ingredient><quantity><value>4</value><unit>sprigs</unit>
            </quantity><name>mint</name></ingredient>
            <ingredient><quantity><value>1</value><unit>slice</unit>
            </quantity><name>orange</name></ingredient>
        </ingredients>
        <preparation>stir</preparation><glass>tumbler</glass>
        <book><title>How to mix Drinks</title><author>Jerry Thomas</author>
```

```
            <published>1862</published></book>
    </cocktail>
    <cocktail><title>Brandy Julep</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>teaspoon</unit>
            </quantity><name>white pulverized sugar</name></ingredient>
            <ingredient><quantity><value>1.5</value><unit>wineglasses</unit>
            </quantity><name>brandy</name></ingredient>
            <ingredient><quantity><value>4</value><unit>sprigs</unit>
            </quantity><name>mint</name></ingredient>
            <ingredient><quantity><value>1</value><unit>slice</unit>
            </quantity><name>orange</name></ingredient>
        </ingredients>
        <preparation>stir</preparation><glass>tumbler</glass>
        <book><title>How to mix Drinks</title><author>Jerry Thomas</author>
            <published>1862</published></book>
    </cocktail>
</cocktails>
```

B.5. alexander.xml

```
<cocktails>
    <cocktail><title>Alexander</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>part</unit>
            </quantity><name>sweet cream</name></ingredient>
            <ingredient><quantity><value>1</value><unit>parts</unit>
            </quantity><name>Creme de cacao</name></ingredient>
            <ingredient><quantity><value>4</value><unit>teaspoonful</unit>
            </quantity><name>gin</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>Fine Art of Mixing Drinks</title><author>David A. Embury</author>
            <published>1948</published></book>
    </cocktail>
    <cocktail><title>Alexander #2</title>
        <ingredients>
            <ingredient><quantity><value>2</value><unit>parts</unit>
            </quantity><name>gin</name></ingredient>
            <ingredient><quantity><value>1</value><unit>part</unit>
            </quantity><name>Creme de cocoa</name></ingredient>
            <ingredient><quantity><value>1</value><unit>part</unit>
            </quantity><name>lime juice</name></ingredient>
            <ingredient><quantity><value>1</value><unit>pony</unit>
            </quantity><name>thick sweet cream</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>Shake em up!</title><author>Virginia Elliott and Phil D. Stong</author>
            <published>1930</published></book>
    </cocktail>
</cocktails>
```

B.6. aromatic.xml

```
<cocktails>
    <cocktail><title>Dubonnet Cocktail</title>
        <ingredients>
            <ingredient><quantity><value>0.5</value><unit>part</unit>
            </quantity><name>Gin</name></ingredient>
            <ingredient><quantity><value>0.5</value><unit>part</unit>
            </quantity><name>Dubonnet</name></ingredient>
        </ingredients>
        <preparation>stir</preparation><glass>cocktail glass</glass>
        <book><title>Savoy Cocktail Book</title><author>Harry Craddock</author>
            <published>1930</published></book>
    </cocktail>
    <cocktail><title>Dubonnet</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>part</unit>
            </quantity><name>Dubonnet</name></ingredient>
            <ingredient><quantity><value>2</value><unit>part</unit>
            </quantity><name>Gin</name></ingredient>
        </ingredients>
        <preparation>stir</preparation><glass>cocktail glass</glass>
        <book><title>Fine Art of Mixing Drinks</title><author>David A. Embury</author>
            <published>1948</published></book>
    </cocktail>
</cocktails>
```

B.7. vermouth.xml

```
<cocktails>
    <cocktail><title>Fancy Vermouth Cocktail</title>
        <ingredients>
            <ingredient><quantity><value>2</value><unit>dashes</unit>
            </quantity><name>bitters</name></ingredient>
            <ingredient><quantity><value>2</value><unit>dashes</unit>
            </quantity><name>maraschino</name></ingredient>
            <ingredient><quantity><value>1</value><unit>wineglass</unit>
            </quantity><name>red vermouth</name></ingredient>
            <ingredient><quantity><value>1</value><unit></unit>
            </quantity><name>lemon slice</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>How to mix Drinks</title><author>Jerry Thomas</author>
            <published>1862</published></book>
    </cocktail>
    <cocktail><title>Vermouth Cocktail</title>
        <ingredients>
            <ingredient><quantity><value>2</value><unit>dashes</unit>
            </quantity><name>bitters</name></ingredient>
            <ingredient><quantity><value>1</value><unit>wineglass</unit>
            </quantity><name>red vermouth</name></ingredient>
            <ingredient><quantity><value>1</value><unit></unit>
            </quantity><name>lemon slice</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>How to mix Drinks</title><author>Jerry Thomas</author>
            <published>1862</published></book>
    </cocktail>
    <cocktail>
        <title>Vermouth Cocktail #2</title>
        <ingredients>
            <ingredient><quantity><value>4.5</value><unit>dashes</unit>
            </quantity><name>gum</name></ingredient>
            <ingredient><quantity><value>2.5</value><unit>dashes</unit>
            </quantity><name>Boker’s bitters</name></ingredient>
            <ingredient><quantity><value>1</value><unit>wineglass</unit>
            </quantity><name>vermouth</name></ingredient>
            <ingredient><quantity><value>2</value><unit>dashes</unit>
            </quantity><name>maraschino</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>Bartenders Manual</title><author>Harry Johnson</author>
            <published>1882</published></book>
    </cocktail>
</cocktails>
```

B.8. flip.xml

```
<cocktails>
    <cocktail><title>Cold Rum Flip</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>teaspoon</unit>
            </quantity><name>powdered sugar</name></ingredient>
            <ingredient><quantity><value>1</value><unit>wineglass</unit>
            </quantity><name>jamaica rum</name></ingredient>
            <ingredient><quantity><value>1</value><unit></unit>
            </quantity><name>egg</name></ingredient>
            <ingredient><quantity><value>1</value><unit>pinch</unit>
            </quantity><name>nutmeg</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>medium glass</glass>
        <book><title>How to mix Drinks</title><author>Jerry Thomas</author>
            <published>1862</published></book>
    </cocktail>
    <cocktail><title>Rum Flip</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit></unit>
            </quantity><name>egg</name></ingredient>
            <ingredient><quantity><value>1</value><unit>tablespoons</unit>
            </quantity><name>powdered sugar</name></ingredient>
            <ingredient><quantity><value>2</value><unit>oz</unit>
            </quantity><name>rum</name></ingredient>
            <ingredient><quantity><value>1</value><unit></unit>
            </quantity><name>nutmeg</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>medium size glass</glass>
        <book><title>Savoy Cocktail Book</title><author>Harry Craddock</author>
            <published>1930</published></book>
    </cocktail>
</cocktails>
```

B.9. tomcollins.xml

```
<cocktails>
    <cocktail><title>Tom Collins Gin</title>
        <ingredients>
            <ingredient><quantity><value>5.5</value><unit>dashes</unit>
            </quantity><name>gum syrup</name></ingredient>
            <ingredient><quantity><value>1</value><unit>wineglass</unit>
            </quantity><name>gin</name></ingredient>
            <ingredient><quantity><value>3</value><unit>cl</unit>
            </quantity><name>lemon juice</name></ingredient>
            <ingredient><quantity><value>1</value><unit>wineglass</unit>
            </quantity><name>seltzer water</name></ingredient>
        </ingredients>
        <preparation>stir</preparation><glass>large bar glass</glass>
        <book><title>How to mix Drinks</title><author>Jerry Thomas</author>
            <published>1862</published></book>
    </cocktail>
    <cocktail><title>Tom Collins</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>oz</unit>
            </quantity><name>lemon juice</name></ingredient>
            <ingredient><quantity><value>1</value><unit>tablespoon</unit>
            </quantity><name>powdered sugar</name></ingredient>
            <ingredient><quantity><value>2</value><unit>oz</unit>
            </quantity><name>gin</name></ingredient>
            <ingredient><quantity><value>1</value><unit>wineglass</unit>
            </quantity><name>seltzer water</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>long tumbler</glass>
        <book><title>Savoy Cocktail Book</title><author>Harry Craddock</author>
            <published>1930</published></book>
    </cocktail>
</cocktails>
```

B.10. absinth.xml

```
<cocktails>
    <cocktail><title>Absinthe Cocktail #2</title>
        <ingredients>
            <ingredient><quantity><value>3</value><unit>parts</unit>
            </quantity><name>Absinthe</name></ingredient>
            <ingredient><quantity><value>2</value><unit>parts</unit>
            </quantity><name>Water</name></ingredient>
            <ingredient><quantity><value>1</value><unit>teaspoonful</unit>
            </quantity><name>Sugar Syrup</name></ingredient>
            <ingredient><quantity><value>1</value><unit></unit>
            </quantity><name>Lemon twist</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>Fine Art of Mixing Drinks</title><author>David A. Embury</author>
            <published>1948</published></book>
    </cocktail>
    <cocktail><title>Absinthe Cocktail #3</title>
        <ingredients>
            <ingredient><quantity><value>0.75</value><unit>wineglass</unit>
            </quantity><name>absinthe</name></ingredient>
            <ingredient><quantity><value>0.25</value><unit>wineglass</unit>
            </quantity><name>water</name></ingredient>
            <ingredient><quantity><value>3.5</value><unit>dashes</unit>
            </quantity><name>gum syrup</name></ingredient>
            <ingredient><quantity><value>1</value><unit>dash</unit>
            </quantity><name>Anisette</name></ingredient>
            <ingredient><quantity><value>1</value><unit>dash</unit>
            </quantity><name>Boker’s bitters</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>Bartenders Manual</title><author>Harry Johnson</author>
            <published>1882</published></book>
    </cocktail>
    <cocktail><title>Absinthe Cocktail #4</title>
        <ingredients>
            <ingredient><quantity><value>0.5</value><unit>part</unit>
            </quantity><name>absinthe</name></ingredient>
            <ingredient><quantity><value>0.5</value><unit>part</unit>
            </quantity><name>water</name></ingredient>
            <ingredient><quantity><value>1</value><unit>dash</unit>
            </quantity><name>angostura bitters</name></ingredient>
            <ingredient><quantity><value>1</value><unit>dash</unit>
            </quantity><name>syrup</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>Savoy Cocktail Book</title><author>Harry Craddock</author>
            <published>1930</published></book>
    </cocktail>
</cocktails>
```

B.11. eggnogg.xml

```
<cocktails>
    <cocktail><title>Egg Nogg</title>
```

```
        <ingredients>
            <ingredient><quantity><value>1</value><unit>dash</unit>
            </quantity><name>fine sugar</name></ingredient>
            <ingredient><quantity><value>1</value><unit></unit>
            </quantity><name>egg</name></ingredient>
            <ingredient><quantity><value>1</value><unit>wineglass</unit>
            </quantity><name>cognac</name></ingredient>
            <ingredient><quantity><value>0.5</value><unit>wineglass</unit>
            </quantity><name>santa cruz rum</name></ingredient>
            <ingredient><quantity><value>1</value><unit>wineglass</unit>
            </quantity><name>milk</name></ingredient>
            <ingredient><quantity><value>1</value><unit>pinch</unit>
            </quantity><name>nutmeg</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>tumbler</glass>
        <book><title>How to mix Drinks</title><author>Jerry Thomas</author>
            <published>1862</published></book>
    </cocktail>
    <cocktail><title>Egg Nogg #2</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit></unit>
            </quantity><name>egg</name></ingredient>
            <ingredient><quantity><value>1</value><unit>tablespoon</unit>
            </quantity><name>sugar</name></ingredient>
            <ingredient><quantity><value>1</value><unit>pony-glass</unit>
            </quantity><name>Jamaica Rum</name></ingredient>
            <ingredient><quantity><value>1</value><unit>wineglass</unit>
            </quantity><name>brandy</name></ingredient>
            <ingredient><quantity><value>1</value><unit>wineglass</unit>
            </quantity><name>rich milk</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>tumbler</glass>
        <book><title>Bartenders Manual</title><author>Harry Johnson</author>
            <published>1882</published></book>
    </cocktail>
</cocktails>
```

B.12. whiskeysour.xml

```
<cocktails>
    <cocktail><title>Whiskey Sour</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>part</unit>
            </quantity><name>sugar syrup</name></ingredient>
            <ingredient><quantity><value>2</value><unit>parts</unit>
            </quantity><name>lemon juice</name></ingredient>
            <ingredient><quantity><value>8</value><unit>parts</unit>
            </quantity><name>Rye</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>Fine Art of Mixing Drinks</title><author>David A. Embury</author>
            <published>1948</published></book>
    </cocktail>
    <cocktail><title>Whiskey Sour #2</title>
        <ingredients>
            <ingredient><quantity><value>0.5</value><unit>tablespoon</unit>
            </quantity><name>sugar</name></ingredient>
            <ingredient><quantity><value>3.5</value><unit>dashes</unit>
            </quantity><name>lemon juice</name></ingredient>
            <ingredient><quantity><value>3.5</value><unit>dashes</unit>
            </quantity><name>gum syrup</name></ingredient>
            <ingredient><quantity><value>1</value><unit>squirt</unit>
            </quantity><name>soda</name></ingredient>
            <ingredient><quantity><value>1</value><unit>wine glass</unit>
            </quantity><name>whiskey</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>Bartenders Manual</title><author>Harry Johnson</author>
            <published>1882</published></book>
    </cocktail>
</cocktails>
```

B.13. manhattan.xml

```
<cocktails>
    <cocktail><title>Manhattan (Sweet)</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>part</unit>
            </quantity><name>Italian Vermouth</name></ingredient>
            <ingredient><quantity><value>2</value><unit>parts</unit>
            </quantity><name>Whisky</name></ingredient>
            <ingredient><quantity><value>1</value><unit>dash</unit>
            </quantity><name>Angostura</name></ingredient>
            <ingredient><quantity><value>1</value><unit></unit>
            </quantity><name>Maraschino cherry</name></ingredient>
        </ingredients>
        <preparation>stir</preparation><glass>cocktail glass</glass>
        <book><title>Fine Art of Mixing Drinks</title><author>David A. Embury</author>
            <published>1948</published></book>
    </cocktail>
    <cocktail><title>Manhattan De Luxe</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>part</unit>
            </quantity><name>Cinzano Italian Vermouth</name></ingredient>
            <ingredient><quantity><value>5</value><unit>parts</unit>
            </quantity><name>Bonded Whisky</name></ingredient>
            <ingredient><quantity><value>1</value><unit>dash</unit>
            </quantity><name>Angostura</name></ingredient>
            <ingredient><quantity><value>1</value><unit></unit>
            </quantity><name>Maraschino cherry</name></ingredient>
        </ingredients>
        <preparation>stir</preparation>
        <glass>cocktail glass</glass>
        <book><title>Fine Art of Mixing Drinks</title><author>David A. Embury</author>
            <published>1948</published></book>
    </cocktail>
    <cocktail><title>Manhattan Cocktail (No. 1)</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>wineglass</unit>
            </quantity><name>Vermouth</name></ingredient>
            <ingredient><quantity><value>1</value><unit>pony</unit>
            </quantity><name>rye</name></ingredient>
            <ingredient><quantity><value>1</value><unit>dash</unit>
            </quantity><name>Maraschino</name></ingredient>
            <ingredient><quantity><value>1</value><unit>dash</unit>
            </quantity><name>Angostura Bitters</name></ingredient>
        </ingredients>
        <preparation>stir</preparation><glass>claret glass</glass>
        <book><title>Savoy Cocktail Book</title><author>Harry Craddock</author>
            <published>1930</published></book>
    </cocktail>
</cocktails>
```

B.14. daiquiri.xml

```
<cocktails>
    <cocktail><title>Daiquiri No.1</title>
```

```
        <ingredients>
            <ingredient><quantity><value>8</value><unit>parts</unit>
            </quantity><name>white rum</name></ingredient>
            <ingredient><quantity><value>2</value><unit>parts</unit>
            </quantity><name>lime juice</name></ingredient>
            <ingredient><quantity><value>1</value><unit>part</unit>
            </quantity><name>sugar syrup</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>Fine Art of Mixing Drinks</title><author>David A. Embury</author>
            <published>1948</published></book>
    </cocktail>
    <cocktail><title>Daiquiri No.2</title>
        <ingredients>
            <ingredient><quantity><value>8</value><unit>parts</unit>
            </quantity><name>white rum</name></ingredient>
            <ingredient><quantity><value>2</value><unit>parts</unit>
            </quantity><name>lime juice</name></ingredient>
            <ingredient><quantity><value>1</value><unit>part</unit>
            </quantity><name>sugar syrup</name></ingredient>
            <ingredient><quantity><value>1</value><unit>dash</unit>
            </quantity><name>orange juice</name></ingredient>
            <ingredient><quantity><value>3</value><unit>dashes</unit>
            </quantity><name>orange curacao</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>Fine Art of Mixing Drinks</title><author>David A. Embury</author>
            <published>1948</published></book>
    </cocktail>
    <cocktail><title>Daiquiri No.3</title>
        <ingredients>
            <ingredient><quantity><value>8</value><unit>parts</unit>
            </quantity><name>white rum</name></ingredient>
            <ingredient><quantity><value>2</value><unit>parts</unit>
            </quantity><name>lime juice</name></ingredient>
            <ingredient><quantity><value>1</value><unit>part</unit>
            </quantity><name>sugar syrup</name></ingredient>
            <ingredient><quantity><value>1</value><unit>dash</unit>
            </quantity><name>grapefruit juice</name></ingredient>
            <ingredient><quantity><value>1</value><unit>teaspoonful</unit>
            </quantity><name>maraschino</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>Fine Art of Mixing Drinks</title><author>David A. Embury</author>
            <published>1948</published></book>
    </cocktail>
    <cocktail><title>Pink Daiquiri</title>
        <ingredients>
            <ingredient><quantity><value>8</value><unit>parts</unit>
            </quantity><name>white rum</name></ingredient>
            <ingredient><quantity><value>2</value><unit>parts</unit>
            </quantity><name>lime juice</name></ingredient>
            <ingredient><quantity><value>1</value><unit>part</unit>
            </quantity><name>sugar syrup</name></ingredient>
            <ingredient><quantity><value>1</value><unit>teaspoonful</unit>
            </quantity><name>grenadine</name></ingredient>
            <ingredient><quantity><value>1</value><unit>teaspoonful</unit>
            </quantity><name>maraschino</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>Fine Art of Mixing Drinks</title><author>David A. Embury</author>
            <published>1948</published></book>
    </cocktail>
    <cocktail><title>Golden Glove</title>
        <ingredients>
            <ingredient><quantity><value>8</value><unit>parts</unit>
            </quantity><name>white rum</name></ingredient>
            <ingredient><quantity><value>2</value><unit>parts</unit>
        </quantity><name>lime juice</name></ingredient>
        <ingredient><quantity><value>1</value><unit>part</unit>
        </quantity><name>sugar syrup</name></ingredient>
        <ingredient><quantity><value>1</value><unit>teaspoonful</unit>
        </quantity><name>Cointreau</name></ingredient>
    </ingredients>
    <preparation>shake</preparation><glass>cocktail glass</glass>
    <book><title>Fine Art of Mixing Drinks</title><author>David A. Embury</author>
        <published>1948</published></book>
</cocktail>
<cocktail><title>Ramoncita Lopez Special</title>
    <ingredients>
        <ingredient><quantity><value>8</value><unit>parts</unit>
        </quantity><name>white rum</name></ingredient>
        <ingredient><quantity><value>2</value><unit>parts</unit>
        </quantity><name>lime juice</name></ingredient>
        <ingredient><quantity><value>1</value><unit>part</unit>
        </quantity><name>sugar syrup</name></ingredient>
        <ingredient><quantity><value>0.5</value><unit></unit>
        </quantity><name>egg</name></ingredient>
    </ingredients>
    <preparation>shake</preparation><glass>cocktail glass</glass>
    <book><title>Fine Art of Mixing Drinks</title><author>David A. Embury</author>
        <published>1948</published>
    </book>
</cocktail>
<cocktail><title>Pineapple Bacardi</title>
    <ingredients>
        <ingredient><quantity><value>8</value><unit>parts</unit>
        </quantity><name>white rum</name></ingredient>
        <ingredient><quantity><value>2</value><unit>parts</unit>
        </quantity><name>lime juice</name></ingredient>
        <ingredient><quantity><value>1</value><unit>part</unit>
        </quantity><name>pineapple juice</name></ingredient>
        <ingredient><quantity><value>0.5</value><unit>parts</unit>
        </quantity><name>sugar syrup</name></ingredient>
    </ingredients>
    <preparation>shake</preparation><glass>cocktail glass</glass>
    <book><title>Fine Art of Mixing Drinks</title><author>David A. Embury</author>
        <published>1948</published></book>
</cocktail>
<cocktail><title>Havana Beach</title>
    <ingredients>
        <ingredient><quantity><value>1</value><unit>part</unit>
        </quantity><name>white rum</name></ingredient>
        <ingredient><quantity><value>1</value><unit>part</unit>
        </quantity><name>pineapple juice</name></ingredient>
        <ingredient><quantity><value>0.5</value><unit>teaspoonful</unit>
        </quantity><name>sugar syrup</name></ingredient>
    </ingredients>
    <preparation>shake</preparation><glass>cocktail glass</glass>
    <book><title>Fine Art of Mixing Drinks</title><author>David A. Embury</author>
        <published>1948</published></book>
</cocktail>
<cocktail><title>Nacional</title><ingredients>
        <ingredient><quantity><value>8</value><unit>parts</unit>
        </quantity><name>white rum</name></ingredient>
        <ingredient><quantity><value>8</value><unit>parts</unit>
        </quantity><name>apricot brandy</name></ingredient>
        <ingredient><quantity><value>2</value><unit>parts</unit>
        </quantity><name>lime juice</name></ingredient>
        <ingredient><quantity><value>1</value><unit>teaspoonful</unit>
        </quantity><name>sugar syrup</name></ingredient>
    </ingredients>
    <preparation>shake</preparation><glass>cocktail glass</glass>
    <book><title>Fine Art of Mixing Drinks</title><author>David A. Embury</author>
        <published>1948</published></book>
```

```xml
    </cocktail>
</cocktails>
```

B.15. japanesecocktail.xml

```xml
<cocktails>
    <cocktail><title>Japanese Cocktail</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>part</unit>
            </quantity><name>Orgeat</name></ingredient>
            <ingredient><quantity><value>8</value><unit>parts</unit>
            </quantity><name>cognac</name></ingredient>
            <ingredient><quantity><value>1</value><unit>dash</unit>
            </quantity><name>Angostura</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>Fine Art of Mixing Drinks</title><author>David A. Embury</author>
            <published>1948</published></book>
    </cocktail>
    <cocktail><title>Japanese Cocktail #2</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>tablespoon</unit>
            </quantity><name>orgeat syrup</name></ingredient>
            <ingredient><quantity><value>1.5</value><unit>piece</unit>
            </quantity><name>lemon peel</name></ingredient>
            <ingredient><quantity><value>1.5</value><unit>teaspoon</unit>
            </quantity><name>bitters</name></ingredient>
            <ingredient><quantity><value>1</value><unit>wineglass</unit>
            </quantity><name>brandy</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>How to mix Drinks</title><author>Jerry Thomas</author>
            <published>1862</published></book>
    </cocktail>
</cocktails>
```

B.16. jackrose.xml

```xml
<cocktails>
    <cocktail><title>Jack Rose</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>part</unit>
            </quantity><name>Grenadine</name></ingredient>
            <ingredient><quantity><value>2</value><unit>parts</unit>
            </quantity><name>lemon juice</name></ingredient>
            <ingredient><quantity><value>8</value><unit>parts</unit>
            </quantity><name>Apple Brandy</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>Fine Art of Mixing Drinks</title><author>David A. Embury</author>
            <published>1948</published></book>
    </cocktail>
    <cocktail><title>Apple Jack (Special) Cocktail</title>
        <ingredients>
            <ingredient><quantity><value>0.66</value><unit>part</unit>
            </quantity><name>Apple Jack</name></ingredient>
            <ingredient><quantity><value>0.16</value><unit>parts</unit>
            </quantity><name>lemon juice</name></ingredient>
            <ingredient><quantity><value>0.16</value><unit>parts</unit>
            </quantity><name>Grenadine</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>Savoy Cocktail Book</title><author>Harry Craddock</author>
            <published>1930</published></book>
    </cocktail>
</cocktails>
```

B.17. ginfizz.xml

```
<cocktails>
    <cocktail><title>Gin Fiz</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>teaspoon</unit>
            </quantity><name>powdered sugar</name></ingredient>
            <ingredient><quantity><value>1</value><unit>wineglass</unit>
            </quantity><name>Holland gin</name></ingredient>
            <ingredient><quantity><value>3</value><unit>dashes</unit>
            </quantity><name>lemon juice</name></ingredient>
            <ingredient><quantity><value>1</value><unit>pinch</unit>
            </quantity><name>nutmeg</name></ingredient>
            <ingredient><quantity><value>0.5</value><unit>wineglass</unit>
            </quantity><name>soda</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>medium bar glass</glass>
        <book><title>How to mix Drinks</title><author>Jerry Thomas</author>
            <published>1862</published></book>
    </cocktail>
    <cocktail><title>Albemarle Fizz</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>teaspoon</unit>
            </quantity><name>powdered sugar</name></ingredient>
            <ingredient><quantity><value>1</value><unit>teaspoon</unit>
            </quantity><name>raspberry syrup</name></ingredient>
            <ingredient><quantity><value>2</value><unit>oz</unit>
            </quantity><name>dry gin</name></ingredient>
            <ingredient><quantity><value>1</value><unit>oz</unit>
            </quantity><name>lemon juice</name></ingredient>
            <ingredient><quantity><value>0.5</value><unit>wineglass</unit>
            </quantity><name>soda</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>medium size glass</glass>
        <book><title>Savoy Cocktail Book</title><author>Harry Craddock</author>
            <published>1930</published></book>
    </cocktail>
</cocktails>
```

B.18. cloverclub.xml

```
<cocktails>
    <cocktail><title>Clover Club</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>part</unit>
            </quantity><name>Raspberry syrup</name></ingredient>
            <ingredient><quantity><value>2</value><unit>parts</unit>
            </quantity><name>lemon juice</name></ingredient>
            <ingredient><quantity><value>0.5</value><unit></unit>
            </quantity><name>egg</name></ingredient>
            <ingredient><quantity><value>8</value><unit>parts</unit>
            </quantity><name>Gin</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>Fine Art of Mixing Drinks</title><author>David A. Embury</author>
            <published>1948</published></book>
    </cocktail>
    <cocktail><title>Grand Royal Clover Club Cocktail</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>tablespoon</unit>
            </quantity><name>grenadine</name></ingredient>
            <ingredient><quantity><value>0.75</value><unit>oz</unit>
            </quantity><name>lemon juice</name></ingredient>
            <ingredient><quantity><value>1</value><unit></unit>
            </quantity><name>egg</name></ingredient>
            <ingredient><quantity><value>2</value><unit>oz</unit>
            </quantity><name>Gin</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>Savoy Cocktail Book</title><author>Harry Craddock</author>
            <published>1930</published></book>
    </cocktail>
    <cocktail><title>Clover Club Cocktail</title>
        <ingredients>
            <ingredient><quantity><value>2</value><unit>parts</unit>
            </quantity><name>gin</name></ingredient>
            <ingredient><quantity><value>1</value><unit>part</unit>
            </quantity><name>lemon juice</name></ingredient>
            <ingredient><quantity><value>1</value><unit>part</unit>
            </quantity><name>orange juice</name></ingredient>
            <ingredient><quantity><value>1</value><unit>oz</unit>
            </quantity><name>egg</name></ingredient>
        </ingredients>
        <preparation>shake</preparation>
        <glass>cocktail glass</glass>
        <book><title>Shake em up!</title><author>Virginia Elliott and Phil D. Stong</author>
            <published>1930</published></book>
    </cocktail>
</cocktails>
```

B.19. sidecar.xml

```
<cocktails>
    <cocktail><title>Side car de luxe</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>part</unit>
            </quantity><name>Cointreau</name></ingredient>
            <ingredient><quantity><value>2</value><unit>parts</unit>
            </quantity><name>lemon juice</name></ingredient>
            <ingredient><quantity><value>3</value><unit>parts</unit>
            </quantity><name>cognac</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>Fine Art of Mixing Drinks</title><author>David A. Embury</author>
            <published>1948</published></book>
    </cocktail>
    <cocktail><title>Side car de luxe No.2</title>
        <ingredients>
            <ingredient><quantity><value>1</value><unit>part</unit>
            </quantity><name>Cointreau</name></ingredient>
            <ingredient><quantity><value>2</value><unit>parts</unit>
            </quantity><name>lemon juice</name></ingredient>
            <ingredient><quantity><value>3</value><unit>parts</unit>
            </quantity><name>amagnac</name></ingredient>
        </ingredients>
        <preparation>shake</preparation><glass>cocktail glass</glass>
        <book><title>Fine Art of Mixing Drinks</title><author>David A. Embury</author>
            <published>1948</published></book>
    </cocktail>
</cocktails>
```

C. Extract of ontology

```
<rdf:RDF
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
        xmlns:c="http://www.myclassicbar.com/rdf#">
    <!-- Classes definition excluded -->
    <rdf:Property rdf:type="cocktail://ingredient/superordinate" rdf:about="cocktail://ingredient/spirit"
                  rdfs:Literal="spirit" c:alcohol="0.4" />
    <rdf:Property rdf:type="cocktail://ingredient/basic" rdf:about="cocktail://ingredient/juniper/spirit"
                  rdfs:Literal="juniper spirit" >
        <c:kindof rdf:resource="cocktail://ingredient/spirit"/>
    </rdf:Property>
    <rdf:Property rdf:type="cocktail://ingredient/basic" rdf:about="cocktail://ingredient/gin/oldtom"
                  rdfs:Literal="old tom gin">
        <c:kindof rdf:resource="cocktail://ingredient/juniper/spirit"/>
    </rdf:Property>
    <rdf:Property rdf:type="cocktail://ingredient/basic" rdf:about="cocktail://ingredient/genever"
                  rdfs:Literal="genever">
        <c:kindof rdf:resource="cocktail://ingredient/juniper/spirit"/>
    </rdf:Property>
    <rdf:Property rdf:type="cocktail://ingredient/superordinate" rdf:about="cocktail://ingredient/fruit"
                  rdfs:Literal="fruit" c:sweet="0.2" c:sour="0.2">
        <c:defaultQuantity c:unit="cocktail://unit/piece" c:volume="1"/>
    </rdf:Property>
    <rdf:Property rdf:type="cocktail://ingredient/basic" rdf:about="cocktail://ingredient/juice/lemon"
                  rdfs:Literal="lemon juice" c:sour="1" c:water="1">
        <c:kindof rdf:resource="cocktail://ingredient/juice"/>
        <c:liquidSubstitution c:sourceUnit="cocktail://unit/dash" c:targetUnit="cocktail://unit/cl" c:minVolume="0.5"
                              c:maxVolume="1" c:substitute="cocktail://ingredient/juice/lemon"/>
    </rdf:Property>
    <rdf:Property rdf:type="cocktail://ingredient/basic" rdf:about="cocktail://ingredient/lemon"
                  rdfs:Literal="lemon" c:sour="1" c:sweet="0.01">
        <c:defaultQuantity c:unit="cocktail://unit/piece" c:volume="1"/>
        <c:liquidSubstitution c:sourceUnit="cocktail://unit/piece" c:targetUnit="cocktail://unit/cl" c:minVolume="4"
                              c:maxVolume="6" c:substitute="cocktail://ingredient/juice/lemon"/>
        <c:liquidSubstitution c:sourceUnit="cocktail://unit/quarter" c:targetUnit="cocktail://unit/cl" c:minVolume="1"
                              c:maxVolume="1.5" c:substitute="cocktail://ingredient/juice/lemon"/>
        <c:kindof rdf:resource="cocktail://ingredient/fruit"/>
    </rdf:Property>
    <rdf:Property rdf:type="cocktail://ingredient/basic" rdf:about="cocktail://ingredient/soda"
                  rdfs:Literal="soda water"/>
    <rdf:Property rdf:type="cocktail://preparation/cocktail" rdf:about="cocktail://preparation/stir"
                  rdfs:Literal="stirring"/>
    <rdf:Property rdf:type="cocktail://preparation/cocktail" rdf:about="cocktail://preparation/stir"
                  rdfs:Literal="stired"/>
    <rdf:Property rdf:type="cocktail://drinking/glass" rdf:about="cocktail://glassware/collins"
                  rdfs:Literal="Collins glass"/>
    <rdf:Property rdf:type="cocktail://drinking/glass" rdf:about="cocktail://glassware/highball"
                  rdfs:Literal="highball glass"/>
    <rdf:Property rdf:type="cocktail://unit/quantitative" rdf:about="cocktail://unit/cl" rdfs:Literal="cl"
                  c:factor="1"/>
    <rdf:Property rdf:type="cocktail://unit/quantitative" rdf:about="cocktail://unit/ounce" rdfs:Literal="ounce"
                  c:factor="3"/>
    <rdf:Property rdf:type="cocktail://unit/qualitative" rdf:about="cocktail://unit/quarter" rdfs:Literal="quarter"/>
    <rdf:Property rdf:type="cocktail://unit/reference" rdf:about="cocktail://unit/ditto" rdfs:Literal="ditto"/>
    <rdf:Property rdf:type="cocktail://unit/relative" rdf:about="cocktail://unit/part" rdfs:Literal="part"/>
    <rdf:Property rdf:type="cocktail://unit/qualitative" rdf:about="cocktail://unit/lump" rdfs:Literal="lump"/>
    <rdf:Property rdf:type="cocktail://ice" rdf:about="cocktail://ice/cube" rdfs:Literal="ice"/>
</rdf:RDF>
```

D. Extract of acceptability ratings by domain experts

Favorites are chosen out of the clusters (Appendix B) and paired with the recommended recipe. This pair is rated by domain expert and displayed as a triple (favorite,recommendation,rating):

```
Mint Julep (Jerry Thomas: How to mix Drinks 1862)
0.25-0.5 cl syrup
1.5 wineglasses cognac
32.0-48.0 leaf mint
1.0 (piece) orange slice
(1) (piece) tumbler
(stir,tumbler)
BRANDY SMASH (Frank Meier: The artistry of mixing drinks 1936)
small (piece) tumbler
0.13-0.25 cl syrup
little dash water
12.5-17.5 leaf mint
6.0 cl Brandy
(1) fill shaved Ice
1.0 slice Lemon
(stir,tumbler,shaved ice)
Rating:0
BRANDY PUNCH (Frank Meier: The artistry of mixing drinks 1936)
1.0 wineglass brandy
1.0 tablespoons raspberry syrup
2.5 cl lemon juice
1.0 dash Water
1.0 (piece) orange slice
1.0 piece berries
0.25 cl syrup
(1) (piece) tumbler
(stir,tumbler)
CHAMPS ELYSEES (William J. Tarling: Cafe Royal 1937)
1/2 (part) Cognac
1/4 (part) Chartreuse
1/4 (part) Lemon Juice
1 dash Angostura Bitters
(1) (piece) cocktail glass
(shake,cocktail glass)
Rating:2
BRANDY PUNCH (Frank Meier: The artistry of mixing drinks 1936)
1.0 wineglass brandy
1.0 tablespoons raspberry syrup
2.5 cl lemon juice
1.0 dash Water
1.0 (piece) orange slice
1.0 piece berries
0.25 cl syrup
(1) (piece) tumbler
(stir,tumbler)
SPEED (William J. Tarling: Approved Cocktails 1937)
33 % (part) Brandy
33 % (part) Apricot Brandy
16 % (part) Orange Juice
16 % (part) Lemon Juice
1.0 (piece) orange peel
(shake)
Rating:1
```

Bibliography

[AJOP11] Amatriain, Xavier ; Jaimes, Alejandro ; Oliver, Nuria ; Pujol, Josep M.: Data mining methods for recommender systems. In: *Recommender Systems Handbook*. Springer, 2011, S. 39–71

[ANH13] Al-Nazer, Ahmed ; Helmy, Tarek: Semantic Query-manipulation and Personalized Retrieval of Health, Food and Nutrition Information. In: *Procedia Computer Science* 19 (2013), Nr. 0, 163 - 170. http://dx.doi.org/10.1016/j.procs.2013.06.026. – DOI 10.1016/j.procs.2013.06.026. – ISSN 1877–0509. – The 4th International Conference on Ambient Systems, Networks and Technologies (ANT 2013), the 3rd International Conference on Sustainable Energy Information Technology (SEIT-2013)

[ASB08] Allen, James F. ; Swift, Mary ; Beaumont, Will de: Deep Semantic Analysis of Text. In: *Proceedings of the 2008 Conference on Semantics in Text Processing*. Stroudsburg, PA, USA : Association for Computational Linguistics, 2008 (STEP '08), 343–354

[BMCMB$^+$10] Barragáns-Martínez, Ana B. ; Costa-Montenegro, Enrique ; Burguillo, Juan C. ; Rey-López, Marta ; Mikic-Fonte, Fernando A. ; Peleteiro, Ana: A hybrid content-based and item-based collaborative filtering approach to recommend {TV} programs enhanced with singular value decomposition. In: *Information Sciences* 180 (2010), Nr. 22, 4290 - 4311. http://dx.doi.org/10.1016/j.ins.2010.07.024. – DOI 10.1016/j.ins.2010.07.024. – ISSN 0020–0255

[Bou02] Bourne, Malcolm: *Food texture and viscosity: concept and measurement*. Academic press, 2002

[CHHH06] Chen, Yu-Hsin ; Huang, Ting-hsiang ; Hsu, David C. ; Hsu, Jane Yung-jen: Color-Cocktail: An Ontology-Based Recommender System. In: Liu, Hugo (Hrsg.) ; Mihalcea, Rada (Hrsg.): *2006 AAAI Workshop on Computational Aesthetics: Artificial Intelligence Approaches to Beauty and Happiness (AAAI 2006)*. Menlo Park, California : The AAAI Press, 2006 (Technical Report WS-06-04), 79–82

[CLS01] Candan, K. S. ; Liu, Huan ; Suvarna, Reshma: Resource Description Framework: Metadata and Its Applications. In: *SIGKDD Explor. Newsl.* 3 (2001), Juli, Nr. 1, 6–19. http://dx.doi.org/10.1145/507533.507536. – DOI 10.1145/507533.507536. – ISSN 1931–0145

[CMR07] Carvalho, Gracinda ; Matos, David M. ; Rocio, Vitor: Document Retrieval for Question Answering: A Quantitative Evaluation of Text Preprocessing. In: *Proceedings of the ACM First Ph.D. Workshop in CIKM.* New York, NY, USA : ACM, 2007 (PIKM '07). – ISBN 978–1–59593–832–9, 125–130

[Coo94] Cooke, Nancy J.: Varieties of knowledge elicitation techniques. In: *International Journal of Human-Computer Studies* 41 (1994), Nr. 6, S. 801–849

[CWML13] Chang, Yi ; Wang, Xuanhui ; Mei, Qiaozhu ; Liu, Yan: Towards Twitter Context Summarization with User Influence Models. In: *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining.* New York, NY, USA : ACM, 2013 (WSDM '13). – ISBN 978–1–4503–1869–3, 527–536

[DK11] Desrosiers, Christian ; Karypis, George: A comprehensive survey of neighborhood-based recommendation methods. In: *Recommender systems handbook.* Springer, 2011, S. 107–144

[Edw16] Edwards: *The Fragrance Wheel.* Online visit (22.02.2016) http://www.fragrancesoftheworld.com/downloads/TheFragranceWheel2013.jpg : Website, 2016

[FB10] Freyne, Jill ; Berkovsky, Shlomo: Intelligent Food Planning: Personalized Recipe Recommendation. In: *Proceedings of the 15th International Conference on Intelligent User Interfaces.* New York, NY, USA : ACM, 2010 (IUI '10). – ISBN 978–1–60558–515–4, 321–324

[FKPT07] Fahrmeir, Ludwig ; Künstler, Rita ; Pigeot, Iris ; Tutz, Gerhard: *Statistik.* Springer-Verlag Berlin Heidelberg, 2007 (Springer-Lehrbuch). http://books.google.de/books?id=ZinjP103iRcC. – ISBN 9783540697398

[FPSS96] Fayyad, Usama ; Piatetsky-Shapiro, Gregory ; Smyth, Padhraic: The KDD Process for Extracting Useful Knowledge from Volumes of Data. In: *Commun. ACM* 39 (1996), November, Nr. 11, 27–34. http://dx.doi.org/10.1145/240455.240464. – DOI 10.1145/240455.240464. – ISSN 0001–0782

[FPSSU96] Fayyad, Usama M. ; Piatetsky-Shapiro, Gregory ; Smyth, Padhraic ; Uthurusamy, Ramasamy: Advances in knowledge discovery and data mining. (1996)

[GEFT+15] Ge, Mouzhi ; Elahi, Mehdi ; Fernaández-Tobías, Ignacio ; Ricci, Francesco ; Massimo, David: Using Tags and Latent Factors in a Food Recommender System. In: *Proceedings of the 5th International Conference on Digital Health 2015.* New York, NY, USA : ACM, 2015 (DH '15). – ISBN 978–1–4503–3492–1, 105–112

[Glö91] Glöss, W: *H&R fragrance guide: Feminine notes, masculine notes.* 1991

[HKRS07] Hitzler, Pascal ; Krötzsch, Markus ; Rudolph, Sebastian ; Sure, York: *Semantic Web: Grundlagen.* Springer-Verlag, 2007

[HLE12] Harvey, Morgan ; Ludwig, Bernd ; Elsweiler, David: Learning user tastes: a first step to generating healthy meal plans? In: *First International Workshop on Recommendation Technologies for Lifestyle Change (LIFESTYLE 2012)* (2012), S. 18

[Jel97] Jellinek, Paul: Perfumery and eroticism. In: Jellinek, J.Stephan (Hrsg.): *The Psychological Basis of Perfumery.* Springer Netherlands, 1997. – ISBN 978–94–010–7200–7, S. 3–3

[JMF99] Jain, A. K. ; Murty, M. N. ; Flynn, P. J.: Data Clustering: A Review. In: *ACM Comput. Surv.* 31 (1999), September, Nr. 3, 264–323. `http://dx.doi.org/10.1145/331499.331504`. – DOI 10.1145/331499.331504. – ISSN 0360–0300

[KF10] Karikome, Shihono ; Fujii, Atsushi: A System for Supporting Dietary Habits: Planning Menus and Visualizing Nutritional Intake Balance. In: *Proceedings of the 4th International Conference on Uniquitous Information Management and Communication.* New York, NY, USA : ACM, 2010 (ICUIMC '10). – ISBN 978–1–60558–893–3, 56:1–56:6

[KLSL12] Kuo, Fang-Fei ; Li, Cheng-Te ; Shan, Man-Kwan ; Lee, Suh-Yin: Intelligent Menu Planning: Recommending Set of Recipes by Ingredients. In: *Proceedings of the ACM Multimedia 2012 Workshop on Multimedia for Cooking and Eating Activities.* New York, NY, USA : ACM, 2012 (CEA '12). – ISBN 978–1–4503–1592–0, 1–6

[KRSW09] Kasneci, Gjergji ; Ramanath, Maya ; Suchanek, Fabian ; Weikum, Gerhard: The YAGO-NAGA Approach to Knowledge Discovery. In: *SIGMOD Rec.* 37 (2009), März, Nr. 4, 41–47. `http://dx.doi.org/10.1145/1519103.1519110`. – DOI 10.1145/1519103.1519110. – ISSN 0163–5808

[LWL14] Li, Xin ; Wang, Mengyue ; Liang, T.-P.: A multi-theoretical kernel-based approach to social network-based recommendation. In: *Decision Support Systems* 65 (2014), Nr. 0, 95 - 104. `http://dx.doi.org/10.1016/j.dss.2014.05.006`. – DOI 10.1016/j.dss.2014.05.006. – ISSN 0167–9236. – Crowdsourcing and Social Networks Analysis

[McD83] McDermott, John: Extracting Knowledge from Expert Systems. In: *Proceedings of the Eighth International Joint Conference on Artificial Intelligence - Volume 1.* San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1983 (IJCAI'83), 100–107

[MHC06] Maulik, U. ; Holder, L.B. ; Cook, D.J.: *Advanced Methods for Knowledge Discovery from Complex Data.* Springer, 2006 (Advanced Information and Knowledge Processing). `http://books.google.de/books?id=OOOSx1X2-_sC`. – ISBN 9781846282843

[mhl14] mhlw.go.jp: *Japanese Food Guide Spinning Top.* Website, 2014. – Online Abruf (06.06.2014) www.mhlw.go.jp/bunya/kenkou/pdf/eiyou-syokuji5.pdf

[Mil95] Miller, George A.: WordNet: A Lexical Database for English. In: *Commun. ACM* 38 (1995), November, Nr. 11, 39–41. http://dx.doi.org/10.1145/219717.219748. – DOI 10.1145/219717.219748. – ISSN 0001–0782

[PGK11] Pinxteren, Youri van ; Geleijnse, Gijs ; Kamsteeg, Paul: Deriving a Recipe Similarity Measure for Recommending Healthful Meals. In: *Proceedings of the 16th International Conference on Intelligent User Interfaces*. New York, NY, USA : ACM, 2011 (IUI '11). – ISBN 978–1–4503–0419–1, 105–114

[pri15] princeton.edu: *Wordnet ontology example*. Online visit (16.01.2015) http://wordnetweb.princeton.edu/perl/webwn?s=whiskey&i=1&h=1000#c : Website, 2015

[PROA12] Papaioannou, Thanasis G. ; Ranvier, Jean-Eudes ; Olteanu, Alexandra ; Aberer, Karl: A Decentralized Recommender System for Effective Web Credibility Assessment. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. New York, NY, USA : ACM, 2012 (CIKM '12). – ISBN 978–1–4503–1156–4, 704–713

[RMG$^+$76] Rosch, Eleanor ; Mervis, Carolyn B. ; Gray, Wayne D. ; Johnson, David M. ; Boyes-Braem, Penny: Basic objects in natural categories. In: *Cognitive Psychology* 8 (1976), Nr. 3, 382 - 439. http://dx.doi.org/10.1016/0010-0285(76)90013-X. – DOI 10.1016/0010–0285(76)90013–X. – ISSN 0010–0285

[RRSK10] Ricci, Francesco ; Rokach, Lior ; Shapira, Bracha ; Kantor, Paul B.: *Recommender Systems Handbook*. 1st. New York, NY, USA : Springer-Verlag New York, Inc., 2010. – ISBN 0387858199, 9780387858197

[RV97] Resnick, Paul ; Varian, Hal R.: Recommender Systems. In: *Commun. ACM* 40 (1997), März, Nr. 3, 56–58. http://dx.doi.org/10.1145/245108.245121. – DOI 10.1145/245108.245121. – ISSN 0001–0782

[SG11] Shani, Guy ; Gunawardana, Asela: Evaluating recommendation systems. In: *Recommender systems handbook*. Springer, 2011, S. 257–297

[SS15] Shadbolt, Nigel ; Smart, Paul R.: Knowledge elicitation: Methods, tools and techniques. (2015)

[STIM09] Shidochi, Yuka ; Takahashi, Tomokazu ; Ide, Ichiro ; Murase, Hiroshi: Finding Replaceable Materials in Cooking Recipe Texts Considering Characteristic Cooking Actions. In: *Proceedings of the ACM Multimedia 2009 Workshop on Multimedia for Cooking and Eating Activities*. New York, NY, USA : ACM, 2009 (CEA '09). – ISBN 978–1–60558–763–9, 9–14

[SZ15] Santos, Tiago R. ; Zárate, Luis E.: Categorical data clustering: What similarity measure to recommend? In: *Expert Systems with Applications* 42 (2015), Nr. 3, 1247

- 1260. http://dx.doi.org/10.1016/j.eswa.2014.09.012. – DOI 10.1016/j.eswa.2014.09.012. – ISSN 0957–4174

[w3.15] w3.org: *RDF - Resource Description Framework.* Online visit (06.01.2015) www.w3.org/RDF : Website, 2015

[WLL+08] Wang, Liping ; Li, Qing ; Li, Na ; Dong, Guozhu ; Yang, Yu: Substructure similarity measurement in chinese recipes. In: *Proceedings of the 17th international conference on World Wide Web* ACM, 2008, S. 979–988

[XYL10] Xie, Haoran ; Yu, Lijuan ; Li, Qing: A Hybrid Semantic Item Model for Recipe Search by Example. In: *Multimedia (ISM), 2010 IEEE International Symposium on*, 2010, S. 254–259

[ZS06] Zarzo, Manuel ; Stanton, David T.: Identification of Latent Variables in a Semantic Odor Profile Database Using Principal Component Analysis. In: *Chemical Senses* 31 (2006), Nr. 8, 713-724. http://dx.doi.org/10.1093/chemse/bjl013. – DOI 10.1093/chemse/bjl013

[ZS09] Zarzo, Manuel ; Stanton, DavidT.: Understanding the underlying dimensions in perfumers odor perception space as a basis for developing meaningful odor maps. In: *Attention, Perception, Psychophysics* 71 (2009), Nr. 2, 225-247. http://dx.doi.org/10.3758/APP.71.2.225. – DOI 10.3758/APP.71.2.225. – ISSN 1943–3921

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 21. April 2016 Sigurd Sippel