Domain-specific recommendation based on deep understanding of text

Sigurd Sippel[1]

Abstract: This paper considers the process of development for a domain-specific recommender system that uses the domain of cocktail recipes as a example 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 extract over 2.000 recipes based on a ontology with over 1.000 ingredients. 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.

Keywords: Content-based recommender systems, Data Mining, Deep Understanding, Feature extraction, Ontology, Basic Categories, Validation, Domain experts

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 [ASdB08] leads to a rich semantic representation of data, which is necessary for content-based 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 deep understanding such as flavors of ingredients enriches the recommendation in the perspective of perception. Domain experts are interviewed to get feedback on the recommendation quality.

Section 2 considers the objectives. In section 3 it follows the related work. To achieve the objectives following four challenges are considered: In section 4 a domain-specific survey with domain experts is used to understand the field of cocktail recipes (challenge one) to process a huge volume of recipes (challenge two). The aim is to learn how recipes depends on recommendation. An ontology is designed to store the features such as ingredients in hierarchy. For challenge three section 5 describes domain-specific distances between classic recipes. The last experiment in section 6 considers an validation of nearest-neighbor recommendation (challenge four). The last section 7 considers the conclusion and future work.

[1] Hamburg University of Applied Sciences, Department of Computer Science, Berliner Tor 7, 20099 Hamburg, sigurd.sippel@haw-hamburg.de

2 Objective

Cocktails are written down as cocktail recipes that contain a name, ingredients including quantity, and partially short information about preferred glassware and preparation.

Manhattan Cocktail [2]
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[3], blogs[4], or cocktail databases[5]. 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 approach 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.

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 paper considers a recommendation is based on content-specific features such as ingredients and their characteristics. These features are extracted from cocktail recipes. Implicit personalization is modeled with the help of an exemplary favorite which 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.

The main question is as follows: Does a knowledge-based distance function present a sufficient precision for a cocktail recommendation? A given recommendation is subjective therefore a recommendation for a specific domain — in this case, cocktails — can only be validated by acceptability survey of domain experts such as bartenders.

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

3 Related work

Domain-specific knowledge is necessary for deeper understanding of a domain to improve content-based recommendations [LdGS11]. Based on semantic modeled knowledge in ontology a deep understanding of text is possible [ASdB08].

A ingredient substitution recommendation based on a ingredient ontology with factors of perception is validated by domain expert [Bo14]. A graph-based recommendation approach is used for cooking recipes, which is focused on ingredients as nodes and preparation as edges [Wa08]. Based on recipe recommendations, the following step is to recommend a complete meal [Ku12]. 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. Another modeling approach is on the basis of nutritional balance [KF10]. 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.

The acceptability factors of sensation of food include the following: Appearance, flavor, and texture [Bo02]. Jelinek's odor effects diagram describe flavors, which contains four main categories — acid, sweet, bitter, and animalic [Je97]. In a study uses odor databases [ZS09] that describe either semantically by a list of similar words or map a numeric value of an odor to reference materials. The result represents a database of similarities, which is called odorant object space. Besides the challenge to understand what a name of odor semantically represent, they show a high accordance between odorant object spaces and expert models such as Jelinek's.

4 Understanding the field of cocktail recipes (challenge one and two)

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 [Si16].

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. 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.

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. These experiments give the first understanding of domain in the first challenge, which forms the basis to extract features out of recipes.

The aim of the experiment is to learn how a cocktail recipe is constructed and which information is extractable for further experiments. A library of 2,155 recipes extracted successfully. The detailed process of parsing in described in [Si16, p. 60]. Following assumptions how recipes works are core aspects of the parsing process (challenge two):

- Recipes contain many different spellings such as *sugar syrup* or *simple syrup*, as well as singular and plural words. 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.
- 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* and also *vermouth* is added to basic category *red vermouth* as a synonym. The most concrete item have to be chosen by entity recognition.
- Recipes contain numbers and fractions as words such as *one−third*. It needs synonyms of numbers or fractions in the ontology. A conversion to digits is necessary. Recipes contain ranges of quantities. It often means seasoning an ingredient.
- Recipes also contain fillers such as *soda*, 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* and therefore it is tendentially the most important ingredient. The chosen concrete quantity must be realistic in terms of the glassware.
- Recipes in historic books contain or-relations such as *bourbon* or *rye*. For example, either bourbon or rye has to be used, not both. Recipes also contain optional ingredients.
- Recipes contain solid ingredients. 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 is declared in ontology.
- 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.

- 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.

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 1). It is a technical presentation that is necessary for the recommendation.

$$trait\, Item\{\, val\, uri : String\,\} \tag{1}$$

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 2). The result is one data structure that represents the internal and external data.

$$trait\, ValueItem\{val\, i : Item,\, val\, name : String\,\} \tag{2}$$

The assignment list 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. The cocktail data structure combined all information about a cocktail. A cocktail needs a name, but all other values such as assignments are optional. Preparation, glassware and ice are subtypes of item, which represents one taxonomy in the ontology.

5 Distances between classic recipes (challenge three)

For recommendation a distance measurement is considered in this experiment. It is assumed that classic recipes have been known for a long time, because they contain a characteristic that isolates them from each other. 52 recipes are clustered by domain experts into 19 clusters [Si16, p. 130] and extracted to measure how well the distances work. This is the first step to get an idea about how distances work.

The similarity between items is defined by shared categories in the ontology. The type is referred to the imaginable class. All classes that do not present superordinates are subclasses of the imaginable class. The basic ingredient categories and ingredient subordinates are subclasses that represent 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.

The core ontology for more abstract categories contains more than 200 ingredients and the extended ontology contains over 1.000 ingredients.

The result presents a list of ingredients showing the ingredient path in the ingredient tree. The searched ingredient is always the first item in the path. In the example (Figure 1), 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.

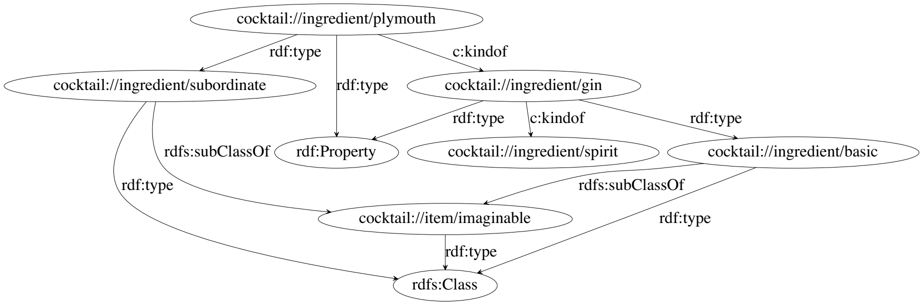


Fig. 1: Example of a ingredient categorization

The path of *Plymouth* contains itself and the parent *gin* (Equation 3). The superordinate is ignored and the types are represented by the chosen data structure such as *BasicIngredient*.

$$\begin{aligned} path_I(Plymouth) = SuboridinateIngredient(&\text{cocktail://ingredient/plymouth}) :: \\ &BasicIngredient(\text{cocktail://ingredient/gin}) :: Nil \end{aligned} \tag{3}$$

In addition 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. Pairs of ingredients and units such as *splash champagne* have default values, because these pairs are imprecise, therefore these pairs substituted [Si16, p. 78] into metric and quantitative units.

5.1 Balance

The balance is an abstract perspective on the cocktail which leans on Jelineks odor model. The result of the survey based on appropriate features for recommendation are flavors (see survey), a extract of the most important ones to describe the classic recipes 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.

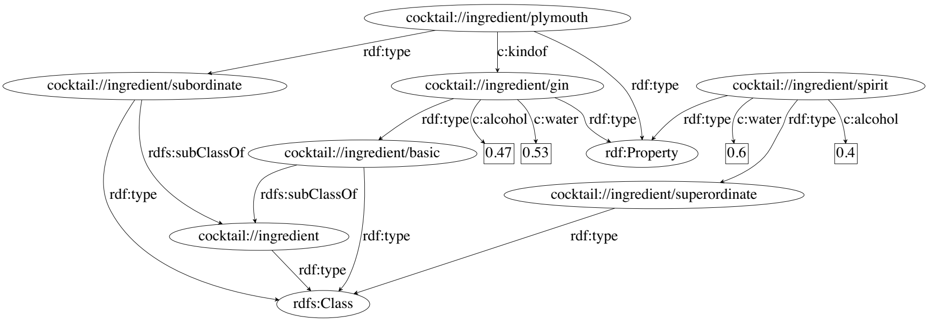


Fig. 2: Balance in ingredient categories

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

The path contains the balance information of all the single ingredients — first *Plymouth*, then *gin*, and finally *spirits* (Equation 4). The question mark is used as a symbol to indicate that the information remains unknown.

$$\begin{aligned} & bal(water, alcohol, sweet, sour, cream, bitter) \\ path_B(Plymouth) &= (?,?,?,?,?,?) :: (0.53, 0.47, ?,?,?,?) :: (0.6, 0.4, ?,?,?,?) :: Nil \\ bal(Plymouth) &= (0.53, 0.47, 0, 0, 0, 0) \end{aligned} \tag{4}$$

5.1.1 Ingredient distance

The distance of a ingredient pair (I_a, I_b) is a path distance (Equation 5), which uses a declared path of two ingredients in the ontology. 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*.

$$d_{DPI}(a,b) = stepDistance(I_a, I_b) \cdot \frac{quantity(I_a)}{volume(a)} \tag{5}$$

The distance of steps has the lowest value 0 if both ingredients remain the same but the quantity is different the quantity-based distance function (Equation 6) is used, which calculates a normalization related to the volume. 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.

$$d_{DPQ}(a,b) = \left|\frac{quan(I_a)}{vol(a)} - \frac{quan(I_b)}{vol(b)}\right| \cdot w \tag{6}$$

The distance of a ingredient pair is dependent on the distance of steps (Equation 7).

$$d_{DP}(a,b) = if(stepDistance == 0)\, d_{DPQ}(a,b)\, else\, d_{DPI}(a,b) \tag{7}$$

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 8) 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.

$$d_I(a,b) = \frac{\sum_{i=1}^{n} arg\,min(d_{DP}(I_{a_i}, I_{b_j})) + \sum_{j=1}^{m} arg\,min(d_{DP}(I_{b_j}, I_{a_i}))}{2} \tag{8}$$

5.1.2 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 9).

$$bal(c) = \sum_{i=1}^{n} bal_i(water, alcohol, sweet, sour, bitter, cream) \cdot \frac{quan(I_i)}{vol(c)} \tag{9}$$

$$d_B(bal) = water + alcohol + sweet + sour + bitter + cream \tag{10}$$

$$d_B(c_a, c_b) = d_B(|bal(c_a) - bal(c_b)|) \tag{11}$$

The difference between two balances (Equation 10) 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 11). All components will be added up to a scalar distance.

6 Validation by domain-experts (challenge four)

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.The results of recommendation are validated by domain experts to get feedback on the results. 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 a given favorite. A analysis of coherence and distinction of classic recipe clusters results a empiric value of distance, which separates the distances into too near distance and distances which shows significant differences [Si16, p. 99]. In the first instance, 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 considered as the most important and are used for recommendation. The second instance, 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.

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 — was offered the examples and a list of recommendations. The domain experts rated the validity of each recommendation on a numeric scale (Equation 12). This scale is designed to present how acceptable a recommendation is.

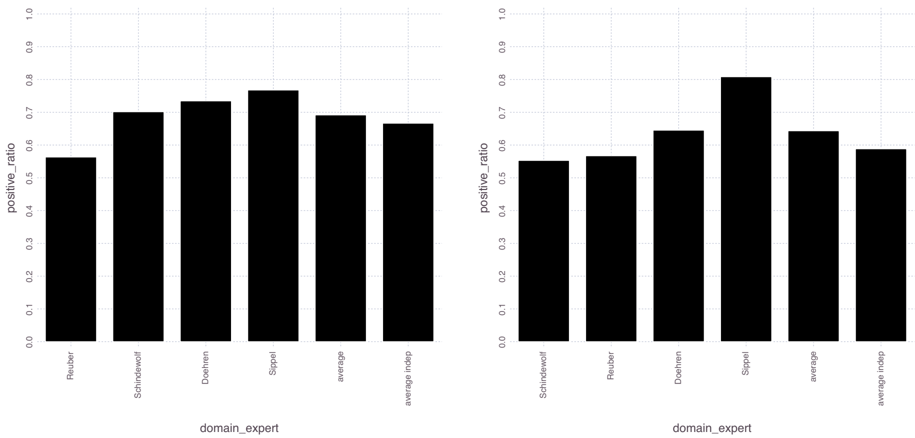
$$\left[\underset{\text{(unacceptable)}}{-2}, \underset{\text{(slightly similar)}}{-1}, \underset{\text{(obviously)}}{0}, \underset{\text{(rather appropriate)}}{1}, \underset{\text{(appropriate)}}{2}\right] \tag{12}$$

19 classic recipes [Si16, Appendix B] used as a favorite to calculate the recommendations. These process is either done for the a focus on balance approach (in total 181 recommen-

dations) and for the focus on ingredient approach (in total 141 recommendations). The pairs of favorite recipe and recommended recipe are rated by the domain experts.

6.1 Acceptability of domain-experts

Four domain experts are interviewed for validation, three are independent and additionally one is dependent to development, who rated in total 1288 pairs of favorites and recommendations (extract in [Si16, p. 145]). The independents 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) Positive ratings with focus on balance

(b) Positive ratings with focus on ingredients

Fig. 3: Positive ratings of domain experts

The rate of positive ratings of focus on balance (Figure 3(a)) is on average about 69 %. The dependent domain expert gives the highest ratings, but the independent average value has a value of about 67 %, which is very close to that. The ratio of positive rating in the focus of ingredients (Figure 3(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 the dependent one.

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.

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.

7 Conclusion and future work

The experiments uses a semi-automated and domain-specific process for recommendation which shows first acceptable results. Deep understanding is possible because it is used a limited domain with available background knowledge.

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. The extraction process is done with a huge volume of recipes. 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. The validation shows that the used process is functional.

For optimizations also the combination with contextualization and individualization should be considered. In perspective of individualization the user model is extensible with further favorites or dislikes, in order to get a higher precision of recommendation. Assumed a huge database of recipes is given, the contextualization such as changes in process of time should be considered. 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: 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.

For specific domains such as news deep understanding could be working, therefore it is a possible research question to prove how it is possible to integrate several domain-specific recommender systems in a bigger recommender system, which classify automatically which specialized reocmmender system is qualified for a specific query of recommendation.

Bibliography

[ASdB08] Allen, James F.; Swift, Mary; de Beaumont, Will: Deep Semantic Analysis of Text. In: Proceedings of the 2008 Conference on Semantics in Text Processing. STEP '08, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 343–354, 2008.

[Bo02] Bourne, Malcolm: Food texture and viscosity: concept and measurement. Academic press, 2002.

[Bo14] Boscarino, Corrado; Nedović, Vladimir; Koenderink, Nicole J. J. P.; Top, Jan L.: Automatic Extraction of Ingredient's Substitutes. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication. UbiComp '14 Adjunct, ACM, New York, NY, USA, pp. 559–564, 2014.

[Je97] Jellinek, Paul: Perfumery and eroticism. In (Jellinek, J.Stephan, ed.): The Psychological Basis of Perfumery, pp. 3–3. Springer Netherlands, 1997.

[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. ICUIMC '10, ACM, New York, NY, USA, pp. 56:1–56:6, 2010.

[Ku12] 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. CEA '12, ACM, New York, NY, USA, pp. 1–6, 2012.

[LdGS11] Lops, Pasquale; de Gemmis, Marco; Semeraro, Giovanni: Recommender Systems Handbook. Springer US, Boston, MA, chapter Content-based Recommender Systems: State of the Art and Trends, pp. 73–105, 2011.

[Si16] Sippel, Sigurd: Domain-specific recommendation based on deep understanding of text http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/sippel.pdf. 2016.

[Wa08] 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, pp. 979–988, 2008.

[ZS09] Zarzo, Manuel; Stanton, DavidT.: Understanding the underlying dimensions in perfumers odor perception space as a basis for developing meaningful odor maps. Attention, Perception, Psychophysics, 71(2):225–247, 2009.