A Web-Based Application for Semantic-Driven Food Recommendation with Reference Prescriptions

Devis Bianchini(✉), Valeria De Antonellis, and Michele Melchiori

Department of Information Engineering, University of Brescia, Via Branze, 38, 25123 Brescia, Italy
{devis.bianchini,valeria.deantonellis,michele.melchiori}@unibs.it

Abstract. Food recommendation, as well as searching for health-related information, presents specific characteristics if compared with conventional recommender systems, since it often has educational purposes, to improve behavioural habits of users. In this paper, we discuss the application of Semantic Web technologies in a menu generation system, that uses a recipe dataset and annotations to recommend menus according to user's preferences. Reference prescription schemes are defined to guide our system for suggesting suitable choices. The recommended menus are generated through three steps. First, relevant recipes are selected by content-based filtering, based on comparisons among features used to annotate both users' profiles and recipes. Second, menus are generated using the selected recipes. Third, menus are ranked taking into account also prescription schemes. The system has been developed within a regional project, related to the main topics of the 2015 World Exposition (EXPO2015, Milan, Italy), where the University of Brescia aims at promoting healthy behavioural habits in nutrition.

1 Introduction

Recommender systems find information of interests, properly customized according to the users' own preferences [1]. This is valid also for specific application domains, such as health and nutrition, where any choice made upon automatically provided recommendations might have an impact on users' health and wellness. Several researches on food recommendation and automatic menu generation have been carried on or are currently active (e.g., [2–4]), taking into account different aspects, such as personal and cultural preferences, health and religion constraints, menu composition and recipe co-occurrence. However, the problem within food recommender systems is still how to suggest recipes and menus that not only meet the user's preferences, but also are compliant with best food habits. Let's consider, for example, Jasmine, who is looking for recipe suggestions to have lunch during her working hours. Jasmine is registered to a food recommender system and has an associated profile. She prefers to have pasta and meat during meals. She suffers from long-term diseases, such as diabetes and high-blood-pressure, therefore white meat should be more advisable.

J. Wang et al. (Eds.): WISE 2015, Part II, LNCS 9419, pp. 32–46, 2015.
DOI: 10.1007/978-3-319-26187-4_3

She belongs to the Islamic religion, so recommendations about any food containing alcohol or pork are not acceptable, since this food is prohibited to Muslims. These aspects may be represented through features, for example a feature representing the religion which a recipe is not advisable for, the course type (e.g., first course, appetizer) and many others. The same features can be used to describe available recipes and Jasmine's preferences, associated to her profile. Features may represent either short-term, immediate preferences (e.g., when they are explicitly specified in a request for suggestions issued by the user), or long-term preferences, extracted from the history of past choices made by the user [5]. A food recommender system would be very useful, not only for the high number of available recipes to be suggested[1], but also because it is really difficult to manually check all the constraints (e.g., religion constraints) and preferences to generate proper menus. Feature-based matching between profiles and recipes is the basis for content-based filtering for food recommendation [2,6,7]. However, some Jasmine's preferences (e.g., having pasta and meat during meals, all the days throughout the week) may contrast with best habits, according to up-to-date medical prescriptions. This means that food recommendations should be able to improve the behavioural habits of the users.

Taking the opportunity of the 2015 World Exposition (EXPO2015, Milan, Italy), the University of Brescia is promoting several projects to incentivate healthy habits. Among them, within the Smart BREAK regional project, funded by the Lombardy region, Italy, we are developing *PREFer* (**P**rescriptions for **RE**commending **F**ood), a menu generation system that uses a recipe dataset and reference prescription schemes to suggest suitable menus. The recommended menus are generated through three steps: (i) relevant recipes are selected by content-based filtering, based on comparisons among features used to annotate recipes and to represent users' preferences; (ii) candidate menus are generated using the selected recipes; (iii) candidate menus are ranked also taking into account reference prescription schemes. As the contribution of this paper, we present the application of Semantic Web technologies within a food recommendation scenario, where the recommendation method is *education-oriented*, that is, aims at satisfying both user's preferences and reference prescriptions.

The paper is organized as follows: in Sect. 2 related approaches for the design of food recommender systems are presented; Sect. 3 provides detailed definitions about our ontology-based recommendation model; in Sect. 4 we describe the three steps of the menu generation procedure; Sect. 5 discusses implementation issues and preliminary experimental results; finally, in Sect. 6 we sketch conclusions and future work.

[1] The http://allrecipes.com web site lists thousands of recipes; for example, just considering appetizers, we can found more than 7,700 choices (http://allrecipes.com/recipes/appetizers-and-snacks/).

2 Related Work

Literature on recommender systems covers several domains and has been developed in parallel with the Web, to properly suggest movies, books, applications, e-learning materials, recipes, etc. (a survey on recommender systems can be found in [1]). Domain-independent categories of recommender systems hold, based on the filtering algorithm used (e.g., demographic, content-based, collaborative, knowledge- or ontology-based, context-aware, hybrid) and on the employed techniques (e.g., probabilistic approaches, nearest neighbors techniques, fuzzy models, similarity metrics). Nevertheless, given the number of domains where recommender systems have been applied and their specific features, a cross-domain comparison might be difficult and useless. Therefore, in the following we will focus on recent approaches on food recommendation domain.

Some existing approaches for recommending food and health-related information focus on content-based filtering (considering aspects like personal and cultural preferences, health and religion constraints) [2,6–8]. In [2] recipes are modelled as complex aggregations of different features, extracted from ingredients, categories, preparation directions, nutrition facts, and authors propose a content-driven matrix factorization approach to face the latent dimension of recipes, users and their features. The HealthFinland project [6] is a portal that helps the users to find relevant health information using simple keywords instead of medical vocabularies. Personalized Health Information System (PHIRS) [8] is a recommender system for health information that matches the user's profile against the retrieved health information, also considering culture and religion in the profile. Similarly, food recommendations are provided in [7].

Teng et al. [9] apply collaborative filtering for recipe recommendation: recipes taken from the `allrecipes.com` Web site are suggested on the basis of users' ratings and reviews and on the basis of co-occurrences of ingredients used to prepare them. In the paper, an interesting survey is provided on other approaches that consider ingredients, recipe ratings and cooking directions. The same information are used in [10,11], where content-based, collaborative and hybrid filtering are compared for recipe recommendation purposes.

Other approaches combine content-based and demographic filtering techniques with ontology-based and knowledge-based tools to enhance recommendation results [12,13]. Ontologies are used to model personal and cultural preferences, health and religion constraints, but no educational issues are taken into account. CarePlan [3] is a semantic representation framework for healthcare plans, that mixes the patients' health conditions with personal preferences, but ignores other aspects, such as personalization coming from educational health information, user's culture and religion, that impact on the food choice. In [4] an ontology containing fuzzy sets is used to sort recommended recipes according to prices and users' ratings, in combination with attributes like sex, age, weight, physical activity, used to calculate Basal Metabolic Rate (BMR), Activity Factor (AF) and Body Mass Index (BMI). Authors implement a demographic filtering algorithm, thus providing common suggestions to people with common attributes.

This variety of approaches demonstrates that users' profiling, in particular for sectors and domains such as the food and health recommendation, is mainly addressed in an ad-hoc manner, without aiming at providing some educational effect on the users. The papers described in [14,15] highlighted this open issue. In particular, [14] presents preliminary research on how to detect bad and correct food habits by analyzing users' ratings on `allrecipes.com`, while in [15] authors discovered that online food consumption and production are highly sensitive in time. Although these approaches do not provide a recommender system, their research could be fruitfully exploited for food recommendation purposes. Other works [16–18] explicitly address the issue of promoting healthful choices, by suggesting recipes to users based on their past food selections and nutrition intakes. We will propose a step forward compared to these approaches, promoting healthy behaviour through reference prescriptions, that are based not only on nutrition intakes, but are specifically modelled considering phenotypes, that classify ideal users' nutrition behaviour. A proper domain ontology is used to model such knowledge and is used with content-based filtering for enhanced food recommendation.

3 Recommendation Model

Let's consider the running example introduced above, where Jasmine is looking for a personalized menu for her meals. Some important aspects should be considered here. First, recipes can be combined into different menus, but not all aggregations are suitable. Specific combinations of recipes might be due to particular *menu configurations* (e.g., appetizer, first course, second course, dessert), according to user's preferences. Second, recommendations might be given according to *reference prescriptions*, that should be used as first-class citizens in recommending recipes to users who present particular profiles. Third, although prescriptions can be used to improve the habits of users for what concerns food and nutrition, they cannot be imposed to users, disregarding their own preferences. Prescriptions should *gradually* move users' choices towards more healthy recipes.

In this paper we propose a recommendation model that is based on the ontology shown in Fig. 1. Following the rationale presented in [19], we distinguish between the ontology and the recipe and menu database, that contains data such as the ones shown in Fig. 2 for the running example. The database contains specific instances of recipes, menus and prescriptions, that are annotated with concepts taken from the ontology. The adopted ontology extends the `food.owl` ontology[2] with the concepts of `CookingStyle` (e.g., Asian cuisine), `Health&CulturalConstraint` such as `Religion` (e.g., Islamic) and `Pathology` (e.g., diabetes, high-blood pressure), `CourseType` (e.g., appetizer, first course, second course, fruits, dessert), `PrescriptionType` and `Phenotype`, that will be presented in the following. The concepts defined within the `food.owl` ontology have been considered as specializations of the `RecipeType` concept. Semantic

[2] http://krono.act.uji.es/Links/ontologies/food.owl/view.

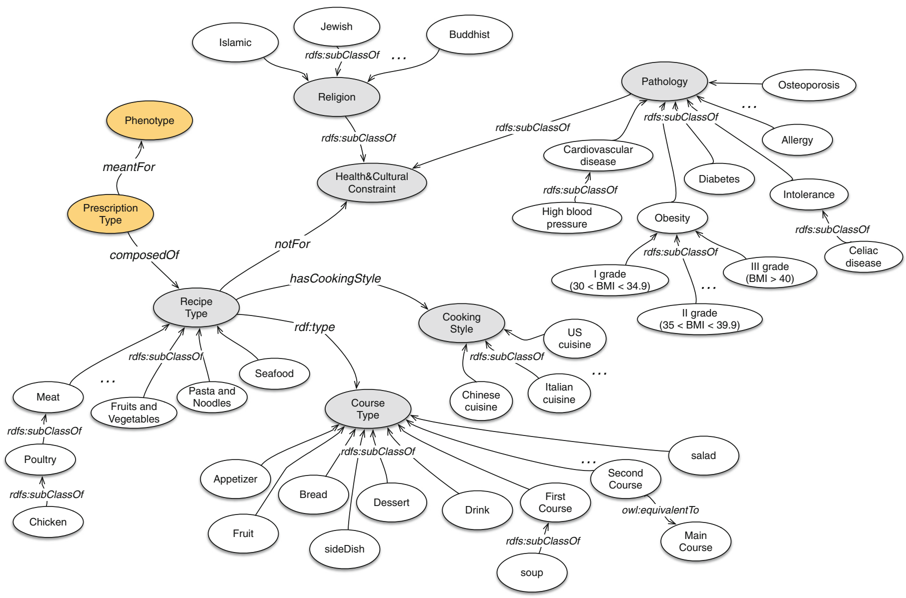

Fig. 1. Main concepts of the ontology adopted for food recommendation.

relationships in the ontology are used to provide food recommendation as discussed in the rest of the paper.

Recipes. Recipes represent the most fine-grained items to be recommended. A recipe is stored in the database as a record $r_i = \langle R_i, n_i, \mathcal{C}_i \rangle$ $(\forall i = 1, \ldots N)$, where: R_i is the unique identifier of the recipe (we denote with $\mathcal{R}$ the overall set of N recipes available within the dataset); n_i is the name of the recipe; $\mathcal{C}_i$ is a set of concepts taken from the ontology, used to characterize the recipe. In particular, in our approach each recipe can be classified through the `CourseType`, the `CookingStyle`, the `RecipeType`, the `Health&CulturalConstraint` (for which the recipe is not advised) and their sub-concepts shown in Fig. 1. In Fig. 2 eight different recipes are depicted, with concepts extracted from the ontology.

In our approach, semantic annotation is supported using the semantic disambiguation techniques we applied in other Semantic Web applications [20]. When a new recipe is published, a text field is provided to enter concepts to annotate it. As the user inputs the characters of the concept name he/she wants to use for annotating the recipe, the system provides an automatic completion mechanism based on the set of concepts contained within the ontology. Starting from the name specified by the user, the system queries the ontology, retrieves the concept with the specified names and/or other concepts related to the specified one through semantic relationships, in order to enable the user to explore the ontology and refine the annotation. Other candidate concepts are also provided according to the string distance between concept names and terms contained in the recipe name and descriptions. A thesaurus (WordNet) is also used in this phase to identify candidate concepts for annotation, using lists of synonyms within WordNet synsets, to face polisemy (that is, the same term refers to different concepts) and synonymy (i.e., the same concept is pointed out using different terms).

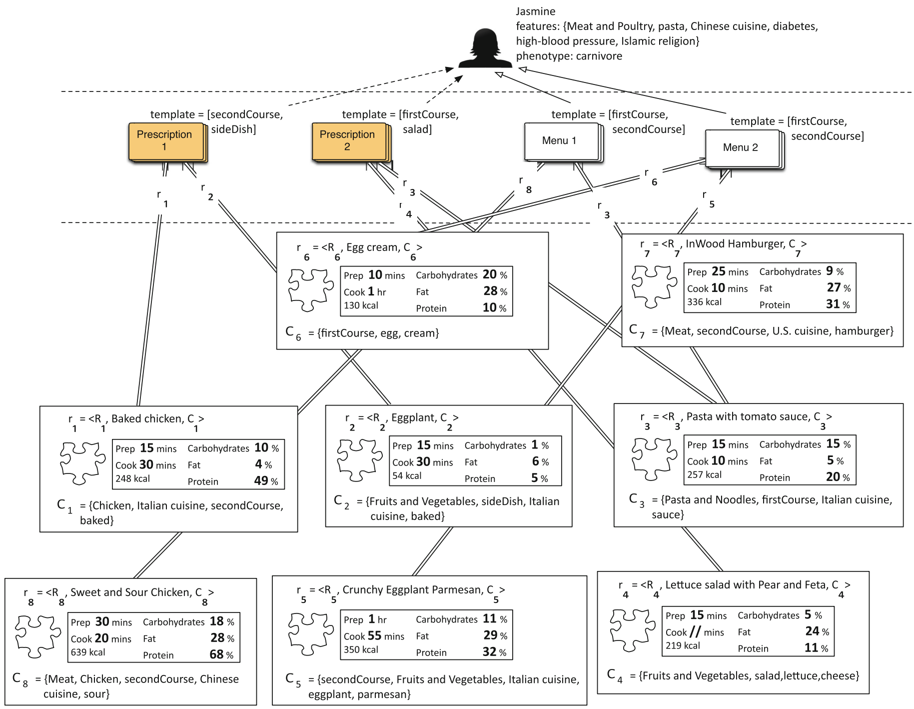

Fig. 2. Recipes to recommend, menus and prescriptions of the running example.

Menus and Prescriptions. Recipes are aggregated to be proposed in a combined way. In the context of our food recommendation approach, we distinguish two kinds of aggregations: (a) available *menus*, that is, combinations of recipes chosen in the past by the users of the system (these menus are used to extract the preferences of the users, exploiting them during the recommendation phase, see for details Sect. 4.1); (b) *prescriptions*, that is, proper combinations of recipes that are advisable for specific kinds of users. Formally, we define an aggregation (either a menu or a prescription) $a_j \in \mathcal{A}$ as $a_j = \langle n_{a_j}, \mathcal{R}[a_j], \tau_{a_j} \rangle$, where: $\mathcal{A}$ denotes the overall set of aggregations; n_{a_j} is the name of the aggregation; $\mathcal{R}[a_j] \subseteq \mathcal{R}$ is the set of recipes aggregated in a_j; τ_{a_j} is the template of the aggregation, expressed in terms of values of a specific concept. In our approach, given an aggregation a_j, τ_{a_j} is identified considering the `CourseType` concept and corresponding sub-concepts (e.g., `Appetizer`, `Fruit`, `sideDish`, etc., see Fig. 1).

Examples of templates may be [`Appetizer`, `FirstCourse`, `SecondCourse`, `Dessert`] or [`FirstCourse`, `Fruit`]. Templates play an important role for the formulation of the request for suggestions (see Sect. 4.1) and to speed up the generation of the recommendation output (see Sect. 4.4). The way prescriptions are associated to users depends on the features used to describe users' profiles. In our food recommendation approach, Food Frequency Questionnaires (FFQ) are issued to collect users' habits and BMI (Body Mass Index), in order to automatically classify users within specific phenotypes [21]. Given a phenotype, one or more prescription types are advisable for it, and each prescription type is composed of a set of recipes types, as specified in the ontology. For example, Jasmine's features identify her phenotype as *carnivore* (Fig. 2). Within the ontology, one of the prescriptions advisable for this phenotype should contain a second course based on chicken, fruits and vegetables. Therefore, `prescription1` in Fig. 2, composed of recipes r_1 and r_2, is compliant with these constraints. The prescription and other compliant ones are automatically generated within the database, given the available recipes. Specification of phenotypes and admissible prescription types for a given phenotype is supervised by medical doctors, who participate to the Smart BREAK project (see Sect. 5 on implementation issues). The point here is that this information is given in the ontology and will be used, as shown in Sect. 4.5.

Users' Profiles. Users are profiled according to their preferences and past menu choices, that are collected to represent the history of recipe and menu selections made by the user in the past. Formally, we define the profile $p(u)$ of a user $u \in \mathcal{U}$ as $p(u) = \langle ID_u, \mathcal{C}[u], \mathcal{M}[u], \mathcal{P}[u] \rangle$, where: $\mathcal{U}$ denotes the overall set of users; ID_u is used to identify the user u; $\mathcal{C}[u]$ is the set of ontological concepts used to denote the preferences of u; $\mathcal{M}[u]$ is the set of menus chosen by the user in the past, that in turn may represent the preferences of the user u about recipes to be recommended; $\mathcal{P}[u]$ is the set of prescriptions assigned to the user in the system, given his/her phenotype and corresponding prescription types.

4 Menu Recommendation System

4.1 Formulating a Request for Suggestions

When Jasmine is looking for menu suggestions, she generates a request $r_r(u)$ formulated as $r_r(u) = \langle \mathcal{C}_r, \tau_r \rangle$, where: $\mathcal{C}_r$ is a set of concepts that represent immediate, short-term preferences of Jasmine; τ_r is the menu template Jasmine is searching for. The recommender system takes into account the profile $p(u)$ of the user u (Jasmine), whom the request comes from. To this aim, the request $r_r(u)$ is expanded with the concepts that are present within the Jasmine's profile $p(u)$. We denote with $\widehat{r}_r(u)$ the expanded version of the request, where $\widehat{r}_r(u) = \langle \widehat{\mathcal{C}}_r, \tau_r \rangle$. The set $\widehat{\mathcal{C}}_r$ contains both the concepts specified in $\mathcal{C}_r$ and the concepts within $p(u)$. Concepts used to characterize $p(u)$ represent long-term preferences of the user, that might be collected and updated using traditional techniques from the literature [5]. The set $\mathcal{C}_r$ might also be empty, thus denoting that the system should exclusively rely on the preferences contained within $p(u)$. Each concept $c_r \in \widehat{\mathcal{C}}_r$ is weighted by means of a coefficient $\omega_r \in [0,1]$ such that:

$$\omega_r = \begin{cases} 1 & if\ c_r \in \mathcal{C}_r \\ freq(c_r) \in [0,1] & otherwise \end{cases} \tag{1}$$

The value of ω_r means that a concept explicitly specified in the request $r_r(u)$ will be considered the most for identifying candidate recipes. The term $freq(c_r)$ computes the frequency of concept c_r among all the concepts that annotate the recipes contained in the profile $p(u)$. Less frequent concepts will be considered as less important for identifying candidate recipes. If a concept c_r is present both in $\mathcal{C}_r$ and in the profile, then $\omega_r = 1$. If u is a new user, without a history of past choices, then $\widehat{r}_r(u) = r_r(u)$ (no expansion). In this case, prescriptions are used to differentiate the user's choices, as explained in Sect. 4.5. In future versions of the *PREFer* system we aim at integrating here further collaborative filtering and demographic filtering recommendation techniques [1].

Example 1. Let's consider the recipes and Jasmine's profile of the running example (Fig. 2), and the following request, issued to search for menus and recipes containing `baked poultry`, according to [`FirstCourse`, `SecondCourse`] template: $r_r(u) = \langle\{\texttt{poultry}, \texttt{baked}\}, [\texttt{FirstCourse}, \texttt{SecondCourse}]\rangle$. The following expanded version of the request is generated (frequency values are specified between parenthesis):

$$\widehat{\mathcal{C}}_r = \{\texttt{poultry}(1.0), \texttt{meat}(0.5), \texttt{chicken}(0.5), \texttt{SecondCourse}(1.0), \texttt{Chinesecuisine}(0.5), \\ \texttt{PastaandNoodles}(0.5), \texttt{FirstCourse}(0.5), \texttt{Italiancuisine}(1.0), \texttt{FruitsandVegetables}(0.5)\}$$
$$\widehat{\mathcal{T}}_r = \{\texttt{baked}(1.0), \texttt{sour}(0.5), \texttt{cream}(0.5), \texttt{egg}(0.5), \texttt{eggplant}(0.5), \texttt{parmesan}(0.5)\}$$

As can be noticed, frequencies are computed on a menu basis, since recipes are recommended only within aggregations, represented as menus.

4.2 Menu Recommendation Steps

The approach followed here for food recommendation is articulated over a set of steps, that are summarized in Fig. 3:

- *feature-based recipe filtering* - the overall set of recipes $\mathcal{R}$ is properly pruned taking into account the menu template τ_r, specified in the request (all recipes that do not present a `CourseType` that is included within τ_r are filtered out from the set of recommendation results) and the features, using proper ontology-based similarity metrics; let's denote with $\mathcal{R}' \subseteq \mathcal{R}$ the set of filtered recipes;
- *candidate menu generation* - candidate menus that are compliant with τ_r are generated, only considering the recipes included within $\mathcal{R}'$; let's denote with $\mathcal{A}^*$ the set of generated candidate menus;

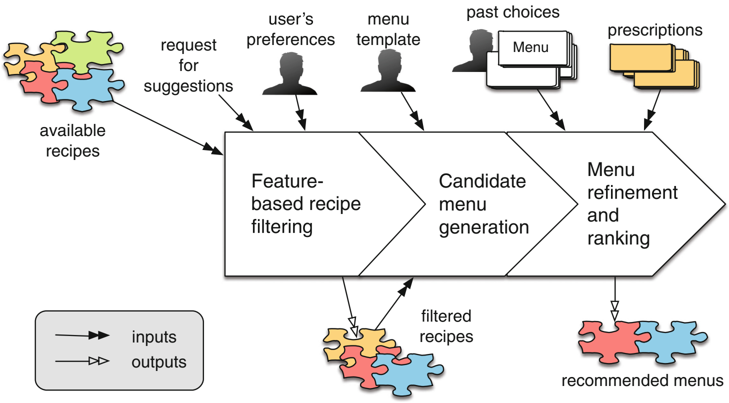

Fig. 3. The three steps of food recommendation approach driven by user's preferences and prescriptions.

– *menu refinement and ranking* - candidate menus contained in $\mathcal{A}^*$ are properly ranked according to their average similarity with the past menu choices made by the user, who is looking for suggestions, and with the prescriptions advertised for that user.

4.3 Feature-Based Recipe Filtering

The input of this step is the set $\mathcal{R}$ of all the available recipes and the request $\widehat{r}_r(u)$. First, τ_r element specified in the request is considered. Recipes such that their `CourseType` is not included within τ_r will not pass the feature-based filtering step. With reference to the running example, only the r_1, r_3, r_5, r_6, r_7 and r_8 recipes will be further considered. To speed up the pre-selection based on `CourseType`, recipes are stored in the underlying dataset indexed with respect to the `CourseType` feature. Another important aspect to be considered is that not all features can be exploited in the same way to filter out not relevant recipes. For instance, let's consider some constraints imposed by the Islamic religion or by some allergies. Recipes that do not respect these constraints must be excluded before any other kind of comparison. These constraints, to keep our model as more general as possible, are defined within the domain ontology and are expressed in terms of other features. For example, the Islamic religion within the Jasmine's profile excludes all recipes that are annotated with `pork` or `alcohol`. Modeling of such constraints must be accurate; this explains why we inserted them within the domain ontology, that is developed in a controlled way.

After τ_r and ontological constraints have been used to pre-select recipes, the filtering based on remaining features is applied, according to the *concept-based relevance*. This metric is computed as follows.

Concept-Based Relevance. The relevance of a recipe $r_i = \langle R_i, n_i, \mathcal{C}_i \rangle$ with respect to the request $\widehat{r}_r(u) = \langle \widehat{\mathcal{C}}_r, \tau_r \rangle$ taking into account concepts in $\mathcal{C}_i$ and $\widehat{\mathcal{C}}_r$, denoted with $Sim(\widehat{r}_r, r_i) \in [0, 1]$, is computed as:

$$Sim(\widehat{r}_r, r_i) = \frac{2 \cdot \sum_{c_r, c_i} \omega_r \cdot ConceptSim(c_r, c_i)}{|\mathcal{C}_i|} \in [0, 1] \tag{2}$$

where c_r ranges over the set $\widehat{\mathcal{C}}_r$, c_i ranges over the set $\mathcal{C}_i$, $|\mathcal{C}_i|$ denotes the number of concepts in the set $\mathcal{C}_i$, ω_r denotes the weight of concept $c_r \in \widehat{\mathcal{C}}_r$, as assigned according to Eq. (1), to take into account both short-term and long-term preferences (see Sect. 4.1). $ConceptSim(c_r, c_i)$ represents the *concept similarity* between c_r and c_i:

$$ConceptSim(c_r, c_i) = \frac{2 \cdot |c_r \cap c_i|}{|c_r| + |c_i|} \in [0, 1] \tag{3}$$

In Eq. (3), we consider the two concepts c_r and c_i as more similar as the number of recipes that have been annotated with both the concepts, denoted with $|c_r \cap c_i|$, increases with respect to the overall number of recipes annotated with c_r, denoted with $|c_r|$, and with c_i, denoted with $|c_i|$. The domain ontology is considered in this case as well: in fact, given two concepts c_i and c_j such that $c_i \sqsubseteq c_j$ (c_i is `subclassOf` c_j), due to the semantics of the `subclassOf` relationship, all recipes annotated with c_i are considered as annotated with c_j as well. For example, $|\texttt{Chicken}| = |\{r_1, r_8\}| = 2$, $|\texttt{Poultry}| = |\{r_1, r_8\}| = 2$, since $\texttt{Chicken} \sqsubseteq \texttt{Poultry}$, $|\texttt{Chicken} \cap \texttt{Poultry}| = |\{r_1, r_8\}| = 2$, therefore $ConceptSim(\texttt{Chicken}, \texttt{Poultry}) = 1.0$.

Pairs of concepts to be considered in the $Sim(\widehat{r}_r, r_i)$ computation are selected according to a maximization function, that relies on the assignment in bipartite graphs and ensures that each concept in $\mathcal{C}_i$ participates in at most one pair with one of the concepts in $\widehat{\mathcal{C}}_r$ and the pairs are selected in order to maximize the overall $Sim(\widehat{r}_r, r_i)$. The rationale behind Eq. (2) is that the closer $Sim()$ to 1.0, the more concepts in $\mathcal{C}_i$ are similar to one of the concepts in $\widehat{\mathcal{C}}_r$. In the running example, for computing $Sim(\widehat{r}_r, r_1)$, the pair $\langle \texttt{Poultry}, \texttt{Chicken} \rangle$ ($\omega_r = 1.0$) is considered instead of $\langle \texttt{Chicken}, \texttt{Chicken} \rangle$ ($\omega_r = 0.5$) in order to maximize the final result, therefore $Sim(\widehat{r}_r, r_1) = (1.0 + 1.0 + 1.0)/3 = 1.0$.

The recipes included in the set $\mathcal{R}' \subseteq \mathcal{R}$, as output of the *feature-based recipe filtering*, are those whose concept-based relevance with respect to the request $\widehat{r}_r(u)$ is equal or greater than a threshold $\gamma \in [0, 1]$ set by the user.

4.4 Candidate Menu Generation

In this step, recipes are aggregated into menus that must be compliant with the template τ_r specified in the request $\widehat{r}_r(u)$. This significantly reduces the number of menu configurations to be generated: in fact, a candidate menu can not contain two recipes r_i and r_j annotated with the same `CourseType`. If we consider, for example, m `CourseTypes`, with an average number of n candidate recipes for each `CourseType`, the number of possible menu configurations without considering the constraint imposed by the menu template would be equal to $f_1(n,m) = \frac{(n \cdot m!)}{m!(n \cdot m - m)!}$ (since we have $n \cdot m$ elements from which m elements must be selected to be composed, without repetitions). In our approach, the number of possible menu configurations is equal to $f_2(n,m) = n^m$. Moreover, the menu generation in our approach is not performed through a brute force procedure, where all possible n^m configurations are generated and, only after generation, properly ranked. The candidate recipes are already sorted, according to the concept-based similarity $Sim(\widehat{r}_r, r_i)$, therefore the candidate menus are generated as illustrated in Fig. 4 with the running example.

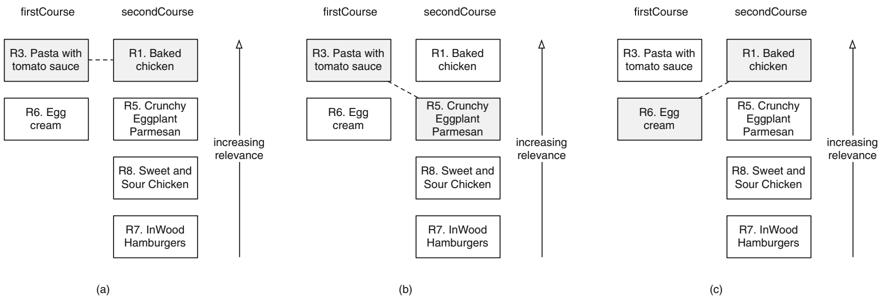

Fig. 4. The menu generation step.

The first candidate menu that is generated is the one where candidate recipes are the best ranked ones for each `CourseType` (Fig. 4(a)). The next candidate menus that are generated are the ones shown in Fig. 4(b–c), where, for instance, $Sim(\widehat{r}_r, r_5) > Sim(\widehat{r}_r, r_6)$. This explains why we choose the combination $r_3 - r_5$, before $r_6 - r_1$. This procedure does not ensure that the list of generated menus will be properly ranked as well. The final ranking of menus is performed in the next step.

4.5 Menu Refinement and Ranking

Menus that have been generated in the previous step are ranked according to their similarity with: (i) past menu choices made by the user u who is issuing the request for suggestions, represented by the set $\mathcal{M}[u]$; (ii) prescriptions prepared for the user u according to his/her profile, represented by the set $\mathcal{P}[u]$. Since both menus and prescriptions are formally defined as sets of recipes, the building block in this step is the similarity measure between items aggregations (*item aggregation similarity*), that is computed as follows:

$$Sim_{agg}(a_i, a_j) = \frac{2 \cdot \sum_{r_i, r_j} Sim(r_i, r_j)}{|\mathcal{R}[a_i]| + |\mathcal{R}[a_j]|} \in [0,1] \tag{4}$$

where a_i and a_j represent the two compared aggregations (menus or prescriptions), r_i (resp., r_j) is a recipe included within a_i (resp., within a_j), $|\mathcal{R}[a_i]|$ (resp., $|\mathcal{R}[a_j]|$) denotes the number of recipes included within a_i (resp., within a_j). The rationale behind $Sim_{agg}()$ computation is the same as the one of the concept-based relevance: we consider two aggregations as more similar as the number of similar items in the two aggregations increases.

The final ranking of a generated menu $a_k \in A^*$, recommended to the user u who issued a request for suggestions, is performed through a ranking function $\rho : A^* \mapsto [0, 1]$, computed as follows:

$$\rho(a_i) = \omega_m \cdot \frac{\sum_{a[u] \in \mathcal{M}[u]} Sim_{agg}(a_i, a[u])}{|\mathcal{M}[u]|} + \omega_s \cdot \frac{\sum_{\widehat{a}[u] \in \mathcal{P}[u]} Sim_{agg}(a_i, \widehat{a}[u])}{|\mathcal{P}[u]|} \quad (5)$$

where $\omega_m, \omega_p \in [0, 1]$, $\omega_m + \omega_p = 1.0$, are weights used to balance the impact of past menu choices and prescriptions on the ranking of recommended menus. We have chosen $\omega_m < \omega_p$ (i.e., $\omega_m \cong 0.4$ and $\omega_p \cong 0.6$) in order to stimulate users on improving their food and nutrition habits, without recommending menus and recipes that are too much distant from users' preferences. This is one the most innovative aspects of our approach compared with recent food recommendation literature (see Sect. 2).

5 Implementation and Experimental Issues

We implemented the *PREFer* system as a web application, whose functional architecture is shown in Fig. 5. The *PREFer Web Interface* guides the user through the registration process, the menu recommendation, the publication of new recipes, also supporting semantic annotation (through the *Sense Disambiguation module*), both during the publication of new recipes and the formulation of a request for suggestions, using a wizard similar to the one described in [20]. The Jena reasoner is used to access knowledge stored within the ontology, that is formalised using the Web Ontology Language (OWL). Registration is performed by answering a food frequency survey (FFQ), that is used to collect information about the user in order to compute his/her BMI and identify his/her phenotype [21], in order to prepare suggested prescriptions. This task is executed by medical doctors, who participate to the regional project where PREFer is being developed. The description of this task is out of the scope of this paper. To just give an idea, medical doctors are supported in the identification of phenotypes and have a simple web interface at their disposal (*Prescription Manager*) to prepare and insert prescription types as sets of recipe types. Specific instances of prescriptions, that are compliant with prescription types specified in the ontology, are automatically generated starting from available recipes. These prescriptions are finally assigned to users classified in the phenotype for which prescription types have been built in the ontology. FFQ results are also stored to enable data analysis by doctors for statistical purposes. *Menu recommendation module* implements the recommendation process described in Sect. 4. It supports

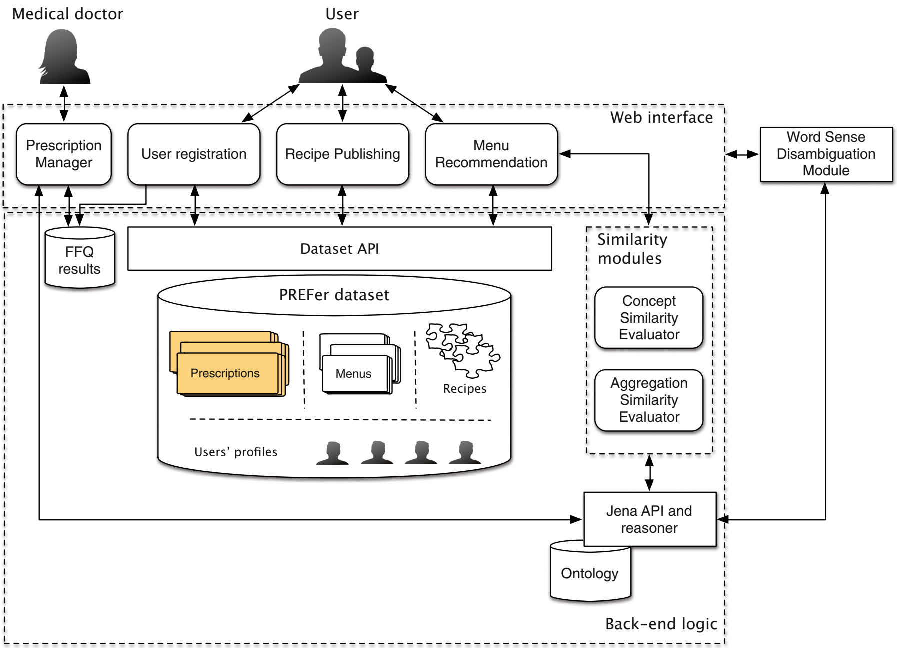

Fig. 5. The functional architecture of the PREFer system for food recommendation.

the user throughout the formulation of the request for suggestions through a proper wizard.

Preliminary Experimentation. Experiments on our food recommendation approach, that are being carried on within the Smart BREAK project, are twofold: (a) to demonstrate the performances of the approach in terms of average precision of the proposed recommendations; (b) to verify the impact of the approach in improving the users' habits concerning food and nutrition. With respect to the former objective, our work has been meant as a complementary approach to recent food recommendation efforts, where content-based filtering techniques based on recipes, ingredients, cultural and contextual features have been implemented. Performance tests are being performed on a dataset that extends an existing one (http://mslab.csie.ntu.edu.tw/~tim/recipe.zip), containing about 220k recipes, randomly aggregated into about 100k menus, where the PREFer system presents comparable average precision with respect to recent approaches. Main experiments in the scope of the Smart BREAK project are being focused on the second objective. They are being performed on a population of about two hundreds students, equally distributed among males and females, with an age included between 18 and 24. Within the population of students, we identified users with pathologies directly related with nutrition (e.g., diabetes, different grades of obesity or various kinds of intolerances) or having bad nutrition behaviour, by submitting Food Frequency Questionnaires. The compliance of users' choices with reference prescriptions, in order to quantify how much the system is able to improve their behaviour, is quantified through the average aggregation similarity between users' choices and reference prescriptions, starting from Eq. (4). Experiments will be carried on until November 2015. Monthly, statistics are generated that, with respect to users' profiles, show the percentage of requests and menu choices that are compliant with or closer to reference prescriptions. Experiments carried on during the first months showed a satisfying increment of closeness between past preferences and reference prescriptions (around 24 % on average, but reaching about 43 % if we consider only users with preferences that are far from the advisable ones, that is, average closeness that is lower than 0.5). These first results are very encouraging and an online community will be created to enabling exchange of food experiences between students who are participating to the experiment.

6 Conclusion and Future Work

In this paper, we presented *PREFer*, a menu generation system that uses a recipe dataset and annotations to recommend menus according to user's preferences. Compared to recent food recommendation efforts, the *PREFer* system takes into account also reference prescriptions schemes, aiming at improving nutritional habits of users. The system has been developed within a regional project, related to the main topics of the 2015 World Exposition (EXPO2015, Milan, Italy), where the University of Brescia aims at promoting healthy behavioural habits in nutrition. The approach will be further extended to refine the recommendation of recipes: (a) by enhancing variety of food in menu preparation; (b) in cases where the violation of health and cultural constraints is due to specific ingredients, by introducing the possibility of suggesting similar recipes, where only the prohibited ingredients are substituted. A semi-automatic functionality for supporting medical doctors in the generation of prescription types will be developed as well. Finally, experimentation is being performed on the approach, but further experiments will be carried on till the end of the Smart BREAK project in order to check the effectiveness of the proposed approach in improving nutritional habits and lifestyles.

References

1. Bobadilla, J., Ortega, F., Hernando, A., Gutièrrez, A.: Recommender systems survey. Knowl.-Based Syst. **46**, 109–132 (2013)
2. Lin, C.-J., Kuo, T.-T., Lin, S.-D.: A content-based matrix factorization model for recipe recommendation. In: Tseng, V.S., Ho, T.B., Zhou, Z.-H., Chen, A.L.P., Kao, H.-Y. (eds.) PAKDD 2014, Part II. LNCS, vol. 8444, pp. 560–571. Springer, Heidelberg (2014)
3. Abidi, S., Chen, H.: Adaptable personalized care planning via a semantic web framework. In: 20th International Conference of the European Federation for Medical Informatics (2006)

4. Maneerat, N., Varakulsiripunth, R., Fudholi, D.: Ontology-based nutrition planning assistance system for health control. Asean Eng. J. **1**(2), 28–41 (2013)
5. Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A.: User profiles for personalized information access. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 54–89. Springer, Heidelberg (2007)
6. Suominen, O., Hyvonen, E., Viljanen, K., Hukka, E.: HealthFinland - a national semantic publishing network and portal for health information. In: Web Semantics: Science, Services and Agents on the World Wide Web 2009, pp. 287–297 (2009)
7. Dominguez, D., Grasso, F., Miller, T., Serafin, R.: PIPS. An integrated environment for health care delivery and healthy lifestyle support. In: 4th Workshop on Agent applied in Healhcare ECAI 2006 (2006)
8. Wang, Y., Liu, Z.: Personalized health information retrieval system. In: AMIA Annual Symposium Proceedings, p. 1149 (2005)
9. Teng, C., Lin, Y., Adamic, L.A.: Recipe recommendation using ingredient networks. In: Proceedings of the 4th Annual ACM Web Science Conference, pp. 298–307 (2011)
10. Freyne, J., Berkovsky, S.: Recommending food: reasoning on recipes and ingredients. In: De Bra, P., Kobsa, A., Chin, D. (eds.) UMAP 2010. LNCS, vol. 6075, pp. 381–386. Springer, Heidelberg (2010)
11. Sobecki, J., Babiak, E., Słanina, M.: Application of hybrid recommendation in web-based cooking assistant. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) KES 2006. LNCS (LNAI), vol. 4253, pp. 797–804. Springer, Heidelberg (2006)
12. Kim, J., Chung, K.: Ontology-based healthcare context information model to implement ubiquitous environment. Multimedia Tools Appl. **71**, 873–888 (2014)
13. Al Nazer, A., Helmy, T., Al Mulhem, M.: User's profile ontology-based semantic framework for personalized food and nutrition recommendation. Procedia Comput. Sci. **32**, 101–108 (2014)
14. Said, A., Bellogín, A.: You are what you eat! tracking health through recipe interactions. In: Proceedings of the 6th Workshop on Recommender Systems and the Social Web (RSWeb) (2014)
15. Kusmierczyk, T., Trattner, C., Nørvåg, K.: Temporality in online food recipe consumption and production. In: Proceedings of the ACM 2015 International Conference on World Wide Web Companion (WWW 2015) (2015)
16. Geleijnse, G., Natchtigall, P., van Kaam, P., Wijgergangs, L.: A personalized recipe advice system to promote healthful choices. In: IUI, pp. 437–438. ACM (2011)
17. Kamieth, F., Braun, A., Schlehuber, C.: Adaptive implicit interaction for healthy nutrition and food intake supervision. In: Jacko, J.A. (ed.) Human-Computer Interaction, Part III, HCII 2011. LNCS, vol. 6763, pp. 205–212. Springer, Heidelberg (2011)
18. Hsiao, J., Chang, H.: SmartDiet: a personal diet consultant for healthy meal planning. In: Proceedings of IEEE 23rd International Symposium on Computer-Based Medical Systems (CBMS) (2010)
19. Garcia, P., Mena, E., Bermùdez, J.: Some common pitfalls in the design of ontology-driven information systems. In: International Conference on Knowledge Engineering and Ontology Development (KEOD 2009), pp. 468–471 (2009)
20. Bianchini, D., De Antonellis, V., Melchiori, M.: A linked data perspective for effective exploration of web APIs repositories. In: Daniel, F., Dolog, P., Li, Q. (eds.) ICWE 2013. LNCS, vol. 7977, pp. 506–509. Springer, Heidelberg (2013)
21. Rankinen, T., Bouchard, C.: Genetics of food intake and eating behavior phenotypes in humans. Annu. Rev. Nutr. **26**, 413–434 (2006)