Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 1055–1062
ISBN 978-83-60810-51-4

Semantic Web Recommendation Application

Szymon Łazaruk, Jakub Dzikowski, Monika Kaczmarek and Witold Abramowicz
Department of Information Systems
Faculty of Informatics and Electronic Economy
Poznan University of Economics
Poland, Poznan 61-875
Email: [s.lazaruk,j.dzikowski,m.kaczmarek,w.abramowicz]@kie.ue.poznan.pl

Abstract—**This paper focuses on a semantically-enhanced Social Web Recommendation application, called Taste It! Try It! It is a mobile restaurants' review and recommendation application based on a Linked Data source and integrated with a social network. The application is consuming Linked Data (while creating the reviews), producing semantic annotations (about the reviewed entities), and then, querying the gathered data in order to offer personalized recommendations.**

I. Introduction

THE Semantic Web is to enable machines to understand the meaning of information on the WWW via extending the network of hyperlinked human-readable web pages by inserting machine-readable metadata, i.e., semantic annotations, thus, enabling automated reasoning [1]. A semantic annotation is machine processable, if it is explicit, formal, and unambiguous and this goal is usually reached by using ontologies [2].

Semantic review systems are those whose performance is based on some knowledge base defined as, e.g., ontology [3] [4]. The application of ontology within the review system: semantically extends descriptions of user opinions; allows to complete the incomplete information through inferences; semantically extends descriptions of user contextual factors; allows for the dynamic contextualization of user preferences and opinions in specific domains; guarantees the interoperability of system resources and the homogeneity of the representation of information; improves communication processes between agents and between agents and users [4].

The Web has evolved into Web of Data [5] by using a set of best practices for publishing and connecting structured data on the Web known as Linked Data. The Linking Open Data

project[1] aims at building a "data commons by making various open data sources available on the Web as RDF and by setting RDF links between data items from different data sources". Examples of datasets encompass well-known DBpedia, Geonames or Freebase. The content of the Linked Data cloud is diverse in nature, comprising inter alia [5]: data about geographic locations, people, companies, books, scientific publications, movies, music, television and radio programmes, genes, proteins, drugs and clinical trials, online communities, statistical data, census results, and reviews (e.g., Revyu system [6]). Since 2007, the Linking Open Data cloud has expanded considerably. However, apart of some initiatives showing how to build applications using it, there is still plenty of space for more end-user applications that operate on Linked Data.

The above trends constitute a motivation for the development of a tool supporting the creation of semantic content using mobile devices in a user-friendly manner operating on Linked Data. The goal of the proposed application is to make annotations sufficiently easy to create for end-users' acceptance, while at the same time provide added value through the ease of finding suitable data, and the ease of integrating the data and reason on them. Thus, in the proposed application, we show, how faceted based annotations (annotations taking advantage of an a-priori domain knowledge and labelling) along with the social network based user friendly application can support users in creating semantic annotations of reviews that may be further on used within recommendations systems.

The work conducted encompassed both the research and practical related aspects. On the one hand, the aim was to contribute to a general understanding of the problem and on the other hand, the aim was to develop a system that could not only be used as a proof for testing, but also could constitute a fully fledged tool to be used by users. Thus, the System Development Method (SDM) was utilized [7]. According to Burstein – SDM "allows the exploration of the interplay between theory and practice, advancing the practice, while also offering new insights into theoretical concepts". The approach followed consisted then out of three main steps. First, the concept building phase took place, which resulted in the theoretical concepts presented in the next sections. The next step was the process of system building, which was encompassing development of a system based on the theoretical concepts established. The system development was guided by a methodology of incentivizing users to create semantic content developed within the INSEMTIVES project[2]. Some core concepts of created application has been presented during the previous AITM conference [8]. This paper is to describe the final version of the application and to discuss the obtained results.

In order to meet the defined goal, the paper is structured as follows. First the related work in the relevant research area is shortly discussed. Then, the vision of the tool is presented. Next, we focus our attention on the functionalities supported by the system including semantic content creation and semantic search. The paper concludes with final remarks.

[1] http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

[2] http:\\insemtives.eu

II. Related Work

Recommender Systems (RS) are information search tools that have been proposed to cope with the information-overload problem, i.e, the typical state of a consumer, having too much information to make a decision [3], [9].

Recommender Systems can be either [10]: *rating-based* (content-based or social/collaborative-based) – users explicitly express their preferences by giving binary or multi-scale scores to items that they have already experienced, or *feature-based* (case-based, utility-based, knowledge-based and critiquing-based) – evaluating the match between a user's need and the set of options available [11].

Recommendation systems attempt to predict items a user may be interested in, given some information about user's preferences and past behaviour, i.e., a user profile [3], [12]. Most existing recommender systems take advantage either of [9]:

- collaborative filtering techniques, i.e., analyzing past actions and behaviour of all users in order to identify interesting associations between them or between the objects, which can be used to make recommendations to a single person (memory-based collaborative filtering (e.g., [13]) and model-based collaborative filtering (e.g., [14]));
- content-based methods, i.e., recommending objects by analyzing the associations between user's past choices and descriptions of new objects [15], [16];
- knowledge-based i.e., system suggest products based on inferences about user's needs and preferences [11], or
- hybrid filtering methods combining previous ones.

Each of these techniques has know shortcomings [9], such as cold-start problem for collaborative and content-based systems or knowledge engineering bottleneck in knowledge-based approaches. They are addressed by a hybrid recommender system that combines multiple techniques to achieve some synergy between them. An in-depth survey of hybrid recommender systems has been presented in [9].

A typical recommendation mechanism analyzes the user context (a user profile, if available), and presents to the user one or more descriptions of objects that may be of their interest. Recommendation mechanisms may be used in pull (recommendations are explicitly requested) or push mode (recommendations are made when a user did not ask for them). In either way, the recommendation should be personalized [20]. Following [16], different levels of personalization can be distinguished starting from coarse grained ones (e.g., relying on the country of residence) to fine-grained (e.g., based on the recent search history). The process of personalization is accurate, if the system possesses accurate information on a user as well as the object/topic the user is interested in, and the information is machine-understandable.

Recommender systems normally use software instead of users for the information filtering tasks [17]. This approach, however, has some disadvantages. The communications process, either between agents and users or agents only, is complicated because of the heterogeneity of information representation, which in turn leads to incapability of its reuse in other processes and applications. Thus, Semantic Web technologies are used within the recommender systems.

A particularly interesting example of ontology-based system has been introduced by Cantador and Castells in [18] and extended in [19]. In this work, a multi-layer semantic social network model has been proposed, based on hypothesis that since user interests are not made of a single piece, any approach that deals with them as such would have inevitable limitations. Thus, the system has been defined from different perspectives, splitting user profiles according to meaningful groups/layers of preferences shared among users, so that the similarities between users are to be established based on sub-profiles rather than the global ones. This approach is also continued in the *Taste It! Try It!* application.

Semantic Web technologies have been introduced almost a decade ago, and yet, their real-life impact has been considerably limited for first few years. The situation has changed dramatically by an initiative called Linked Data project. Based on the simple semantic technologies, like RDF and URIs, used along with Linked Data principles[3], a number[4] of datasets have been made available in a machine-understandable manner, eg., Wikipedia's resources are available on the Web of Data in the form of DBpedia. Linked Data sets are used in more and more real-world application. Examples include [20]:

- Faviki[5] – social bookmarking tool, utilizing semantic tags stemming from Wikipedia (via DBpedia) so that all concepts are ambiguously identified;
- DBpedia mobile[6] – mobile, location-based application presenting information from DBpedia on a map.
- Revyu[7] – a generic reviewing site based on the Linked Data principles and the Semantic Web technology stack.

Linked Data lowers the entry barrier for data providers by focusing on publishing structured data rather than, on the ontological level or inferencing, hence fosters a wide spread adoption. However, there exists some challenges that need to be tackled by developers of real-world linked open data applications, not least of which include resource discovery, consolidation and integration across a distributed environment.

The *Taste It! Try It!* application benefits from the already developed semantic technologies and tools, and offers an added value through their integration and usage in order to, on the one hand, contribute to the Linked Data by producing semantic annotations, and on the other, to offer personalized advanced discovery and clustering possibilities. For the needs of the *Taste It! Try It!* application, a distinct disambiguation solution has been designed, adjusted to the specific needs of a mobile device. All of these features together, make the *Taste It! Try It!* application a distinct solution.

[3]http://www.w3.org/DesignIssues/LinkedData.html
[4]295 datasets up to 2011 - source: http://lod-cloud.net/
[5]http://faviki.com
[6]http://wiki.dbpedia.org/DBpediaMobile
[7]http://revyu.com/

III. Vision of the Tool

Taste It! Try It! has been designed as both a mobile and WWW, Web 2.0 application supporting the creation of semantic annotations describing various places and locations. It is targeted at end-users, among which two

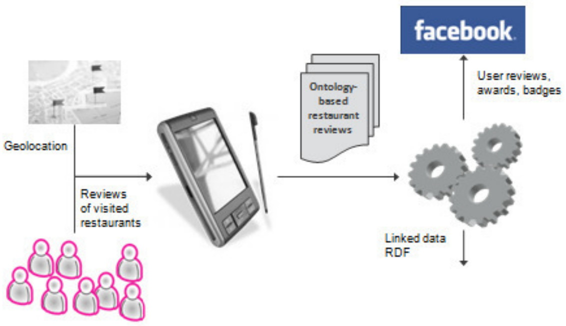

Fig. 1: *Taste It! Try It!*– conceptual model

groups may be distinguished:

- data producers (contributors) – users providing reviews of places, i.e., people creating semantically annotated reviews,
- data consumers (beneficiaries) – users interested in the content produced by the application, i.e., people looking for opinions about various places.

Therefore, on the one hand, the *Taste It! Try It!* should enable data producers to contribute to a semantic content creation process using their mobile devices[8] (as well as a WWW interface, if desired by a user), and on the other, should provide data consumers with personalized, semantic, context-aware recommendation process (i.e., offer a personalized semantic search mechanism).

The storyboard supported by the system is as follows. A user goes to a restaurant. While being at the restaurant, the user decides to share his opinion about the restaurant and its quality of service factors with other members of the community. He uses *Taste It! Try It!* to express this opinion. The application starts by capturing the position of the place (using the GPS system in a mobile device). This enables associating the semantically annotated review that is created afterwards with a specific point in space. Then, the user creates a review by providing values to selected features suggested by the application. Additionally, the user may create a free-text comment regarding the object being reviewed. The review is then uploaded to a *Taste It! Try It!* server and in the background, the semantic representation is created. Based on the quantity and quality of created annotations, the user may be awarded with a special title e.g., *Polish-cuisine expert*, *International-food expert*. This title is visible to his friends at the community portal, in our example the Facebook portal, with which the application is integrated. Moreover, a user may check the ranking among his friends on Facebook. If a specified number of users mark the same spot and assign the same category of place to it (e.g., restaurant XYZ), a new location appears on the map.

In addition, based on the user behaviour and data made available by the Facebook portal, the user profile is created, which is then used in the personalization process. The created annotations are then further on used by a semantic-based recommender system while searching for restaurants fulfilling certain criteria, e.g., vegetarian, low budget, and high quality, in the neighbourhood of a user. As the semantically annotated reviews are linked to Linked Open data sets [5], some more sophisticated reasoning over the data is possible and extends the possibilities offered by the semantic-based recommendation system.

Thus, the application is to fulfil the following goals:

- to provide semantically-enabled reviews sufficiently easy to create for end-user acceptance – the process of attaching the machine understandable semantics should be user-friendly;
- to keep a user entertained - integrating the proposed application with a social portal such as Facebook and adding the possibility of gaining badges, are some of the incentives that are utilised to make the system more attractive to users;
- to offer a personalized, semantic, context-aware recommendation process (both push and pull).

In order to encourage users to create the semantic annotations,

the application is taking advantage of the incentives mechanism defined within the INSEMTIVES project as well as is using some of the tools provided by the INSEMTIVES platform. The incentives mechanisms built-in into the *Taste It! Try It!* application, may be roughly divided into the following dimensions:

- usability of the tool - e.g., ease of creation of the semantic data, background processing of semantics (although the *Taste It! Try It!* application takes advantage of the semantic datasets, the complex semantic nature of the underlying information is hidden to users as much as possible. The users do not interact directly with Semantic Web languages or technologies e.g., *SPARQL* endpoint);
- social aspect - keeping a user entertained (integration with the Facebook portal, awarding users with badges and points for different tasks);
- gaining additional benefits - by using an application a user is obtaining an access to a semantic and context-aware personalized recommendation process (by taking advantage of the semantically enhanced multi-layered clustering approach).

The application has been designed according to the client-server architecture model. The clients however, are twofold: mobile and Facebook, thus, supporting different user-application collaboration models.

IV. Mechanisms and the System Flow

The system flow described within this subsection encompasses the registration process, the usage and navigation within the mobile client, creation of semantically annotated reviews, visualization of submitted reviews as well as a user profile on the Facebook portal, and finally, personalized searching for restaurants of interest.

A. *Registration to* Taste It! Try It!

The first step towards using the *Taste It! Try It!* is to visit the Facebook *Taste It! Try It!* site and grant required

[8]The application is developed to work with the Android system

permissions that are necessary for the application to work (see fig. 2).

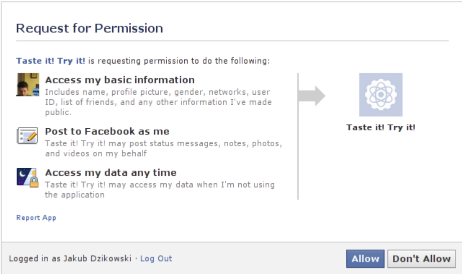

Fig. 2: Facebook permissions' pop-up

Once the access is granted, the Facebook client retrieves from Facebook the basic information about the user and generates a verification code, which is going to be used for the registration of the Android client. This code is a pseudo-random one-day-valid unique sequence of 6 alphanumeric chars, which can uniquely identify the user.

B. Mobile Client navigation

While using the Mobile client, a user can choose any of four menu elements: create new review, manage reviews, check profile and search (see figure 3).

Fig. 3: *Taste It! Try It!* Mobile client menu screen

In order to ensure usability, typical Android UI elements (e.g., buttons, system menus, dialogs and view pagers) have been used. As a result, the use of the *Taste It! Try It!* application is more intuitive and user friendly.

C. Preparing and sending a review

Reviews can be created either using the Mobile client (figure 4), which is the typical and preferred way, or using the Facebook client. All reviews created using the Mobile client can be saved at any time, and the edition process can be resumed whenever the user wishes, using the review-management screen.

The review edition screen is divided into 3 tabs:

- **Main tab** containing expected by most users, basic and obligatory information such as: name of the place being reviewed; type of location: restaurant, pub, fastfood, cafe; GPS location – which is to be provided using mobile device's GPS module; and star ratings that allow the user to express his *Overall*, *Service*, *Atmosphere*, *Food* impressions in the quantitative manner, by assigning from 1 to 5 stars in each category.
- **Details tab** allowing the user to assess a wide range of qualitative features of the place, which are grouped in intuitive categories such as: Dining options, Entertainment or Good for. In this tab, the user is also able to select the cuisine type and best dishes/drinks served. Values of those fields are suggested from DBpedia.
- **More tab** containing some additional star ratings and features together with a free-text comment field.

Fig. 4: *Taste It! Try It!* Mobile client review creation screens

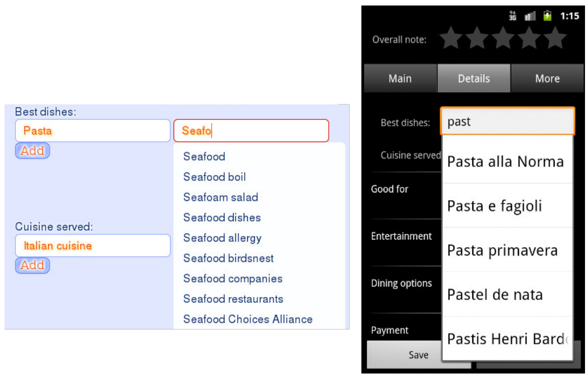

Fig. 5: Autocompletion mechanism

While developing the application, our main motivation was to hide from users the complexity of semantics being the backbone of the application. Users do not interact directly with Semantic Web languages or technologies e.g., *SPARQL* endpoint. The semantic annotations that are created are template-based annotations (faceted-based). Users while filling in some aspects of review (e.g., category of restaurant, type of cuisine, food and drinks served), are pointing to the concepts from DBpedia taking advantage of an auto-completion mechanism suggesting possible tags to be used (see figure 5).

As we are following the faceted-based approach to the review creation, we can benefit from the additional knowledge in order to disambiguate and limit the potential tags (concepts from DBpedia) to be presented to users as an option to choose from. Therefore, for the needs of the

Taste It! Try It! application, a disambiguation solution has been designed as follows – once a user starts to type in the first characters of the tag for the selected feature, the disambiguation takes place in the following steps:

- an appropriate SPARQL query is created limited to the branch of interest (e.g., in case a user is providing a tag to the *best dishes* feature, only the concepts related to Food and drinks are suggested),
- the obtained result is filtered using the preferred language of the user as well as the first characters typed in and as a result the list of concepts with their labels is retrieved from DBpedia,
- the proposed list of suggestions is sorted and presented to users.

The SPARQL queries used to gather the relevant concepts for auto-completion, depending on the type of field being annotated, needed to be created in a semi-automated manner – using appropriate scripts to generate queries based on the manual analysis of the structure of DBpedia.

The reviews provided by users are stored on the server and then, after performed disambiguation, serialized into the RDF format, so that they can be later on published as linked dataset. The mentioned disambiguation of the object being reviewed is performed in the following steps:

- Taking into account the data provided by a user, i.e., the geographical coordinates of the location, the SPARQL query is formulated in order to retrieve all already described places of interest (restaurants and all subcategories) within specific radius from the coordinates provided. As a result a set of objects is being retrieved.
- If the set is not null, the Levenshtein measure is applied to the name of the concepts being retrieved and the name provided in the review in order to identify similar objects (e.g., in case of a typo in a name of the place).
- Based on the above, the ranking list is created. In case only one object reaches the defined similarity threshold, the disambiguation is automatically performed, in all other cases – the system is assuming it is a new concept as long as a user being presented with the ranked list of restaurants will not decide otherwise.

Created semantic annotations that are to be made available outside the *Taste It! Try It!* application, are anonymous (no information about the author of the review is revealed outside the application) and the subjective evaluation of the venue is expressed in an aggregated form. A new venue is being added to the database as soon as it reaches the limit of 3 reviews being assigned. Each new review added about the already existing venue within the knowledge base, may result in updating the information on the restaurant stored there. The created RDF triples are uploaded to the INSEMTIVES platform via the SPARQL interface and stored in the local RDF repository.

D. Contribution visualization and user rewarding

As already mentioned, users are to be rewarded for their contribution, which among other mechanisms, should motivate them to undertake various activities. In addition, for each review submitted to the server, users are awarded

Fig. 6: *Taste It! Try It!* badges example

with a certain amount of points that later allow to create rankings of users of *Taste It! Try It!* application. The badges (see fig. 6) in turn show the status of a user, his/her hobby as well as achievements.

Fig. 7: *Taste It! Try It!*– user profile visualisation and friends ranking

Badges and points are all displayed on the profile and wall of the user on the Facebook portal. Facebook users can view other users' scoring and badges to compare their performance. Thus, through the application of badges and points integrated with such portal as Facebook, the following motivation levers, defined within in *Incentive Models and PD Guidelines* [21], are applied: reputation, competition, conformity to a group, competition, usefulness, altruism, reciprocity and self-esteem.

E. Search

Two key uses of the semantic annotations of restaurant reviews in the *Taste It! Try It!* application are: personalized discovery of restaurants (pull) and personalized recommendations (push). Here we focus on the first one.

In order to support the required personalization of the mentioned processes, it is necessary to get to know the preferences of users of the application. Thus, the creation of user profiles based on the data the user creates (i.e., reviews) and the data from the Social Web (i.e., the

Facebook portal) becomes necessary. In this way, we facilitate a personalized experience without an intrusive user profiling process.

The first step is the creation of ontology-enhanced users' profiles encompassing also their preferences. Then, the clustering of the domain concept space is performed. Once the semantic clusters have been computed, it is possible to identify similarities between individuals and then use this knowledge in collaborative environments, such as *Taste It! Try It!*.

Within our work, we follow the approach presented in [18], and divide users into clusters of cohesive interest (i.e., multiple layers). Thus, for instance, users may share preferences and benefit from each other's experience in specific conceptual areas (e.g., type of food), even if they totally disagree on other aspects (e.g., taste in the décor and atmosphere). Therefore, depending on the context, only selected layers of a user profile may be considered during the recommendation process, allowing to offer more accurate and context-sensitive results [18].

The following layers have been identified based on the data gathered by the *Taste It! Try It!* application:

- restaurants layer – reflecting the similarity in the type of visited restaurants and their locations (it requires computing semantic similarity between restaurants, their type, location – as explained later),
- taste layer – reflecting similarity in the type of food (dishes) and cuisine of the visited restaurants (it requires computing semantic distance between dishes from DBpedia),
- reviews layer – reflecting the similarity in the assigned ratings to the features of visited restaurants (quantitative criteria) as well as similarity in features that the reviewer is interested in (qualitative criteria, e.g., always commenting on music or wi-fi),
- activities layer – reflecting similarities in the performed activities (number of reviews in various locations, awarded badges), as well as social information gathered about a user.

In order to assess the similarity between two concepts in DBpedia, a path between two nodes on a semantic graph needs to be found. Although DBpedia is very successful, there still seems to be a lack of semantic similarity measures available on the DBpedia data, which makes it difficult to use. By taking advantage of the SPARQL endpoint, various queries may be asked to DBpedia, achieving high precision of results. In general, a similarity between two resources in DBpedia can be detected, if in the RDF graph:

- concepts are directly linked using some property,
- they are subject of two RDF triples having the same property and the same object,
- they are objects of two RDF triples having the same property and the same subject.

In DBpedia, the hierarchical structure of categories is modelled using two properties: dcterms:subject (relates a resource to its categories) and skos:broader (used to relate a category to its parent category). Hence, the similarity between dishes' cuisines may be also discovered in case they have some ancestor categories in common (within the hierarchy). This allows one to catch implicit relations and hidden information (i.e., the information that cannot be directly discovered looking only at the nearest neighbours in the RDF graph).

Within our research we apply semantic-based VSM (Vector Space Model) able to deal with RDF graphs in case of semantic-enabled features of an restaurant. As already mentioned, our approach has been based on the work of [22] and [23]. In case of semantic features, each RDF graph is represented as 3-dimensional tensor, where each slice of the tensor refers to a given property from the ontology. A component of a tensor is not null, if there is a property that relates a subject (on the rows of the tensor) to an object (on the columns). Thus, given a specific property, each object (dish, cuisine type or restaurant) is represented as a vector, whose elements refer to the subject frequency-inverse object frequency (term-frequency document frequency where the term is the subject and the document is an object TF-IDF). Thus, for each resource, for a given dimension, the similarity degree is the correlation between the two vectors and may be quantified by the cosine of the angle between them.

In case the property is dcterms:subject (relating a resource to a category) and skos:broader (relating category to other categories), the transitive closure of the categories a resource is related to is computed. Once the whole set is identified (the entire hierarchy of interest as already explained), it is assigned as the value of the dcterms:subject (or skos:broader), depending on whether the resource is a category or not.

As we are interested in computing similarities between resources of the same or similar type, thus, rows represent the domain of the property (all subjects) and the columns - range (all objects). Thus, for a given property, the components of each row represent the information on how a given object (e.g., ingredient) contributes to the corresponding subject (e.g., dish).

Then, given a property p, a resource r is represented by a vector containing all the nodes related to r via p. Thus, a representation of a resource r, according to a property p, is a d-dimensional vector given by:

$$\overrightarrow{r_{i,p}} = (\omega_{1,i,p}, \omega_{2,i,p}, \ldots, \omega_{d,i,p}) \qquad (1)$$

where:

- d is the total number of index terms (i.e., objects),
- $\omega_{n,i,p}$ is a non-negative and non-binary value representing the weight associated with an object-subject pair.

The weights are computed as:

$$\omega_{n,i,p} = f_{n,i,p} \cdot \log \frac{R}{a_{n,p}} \qquad (2)$$

where $f_{n,i,p}$ represents the frequency of the RDF-node n, as the object of an RDF triple having p as a property and the node i as a subject. This term can be at most 1 (as two identical triples cannot be defined within an RDF graph). Thus, in case there is a triple that links a node i to a node n via the property p, the $f_{n,i,p} = 1$, otherwise $f_{n,i,p} = 0$ and the $\omega_{n,i,p} = 0$.

Fig. 8: *Taste It! Try It!*– searching for a restaurant – choosing location, cuisine and dishes

R is the total number of resources (e.g., dishes) in the collection, and $a_{n,p}$ is the number of resources (e.g., dishes) that are linked to the object n, through the property p. Thus, each resource is represented as a $d \times P$ matrix, where P is the total number of properties characterizing a given resource.

Therefore, as in the classical VSM, in order to assess the degree of similarity between $\overrightarrow{r_{i,p}}$ and $\overrightarrow{r_{j,p}}$ the cosine of the angle between two vectors is computed.

Thus, following this approach, we may implement the multi-layered approach by asking which resources are most similar to a given resource taking into account a specific property p, or the entire knowledge base.

The following general discovery scenarios are currently supported within the *Taste It! Try It!* application:

- Searching for a restaurant with some quantitative criteria (non-semantic, e.g., number of stars assigned (not less than...)).
- Searching for a restaurant with some qualitative (non-semantic) criteria added, e.g., wi-fi zone, live sport events transmissions etc.
- Searching restaurants near some location – a map and coordinates (see fig 8).
- Searching for a restaurant with some criteria requiring reasoning (semantic ones from DBpedia) – type of cuisine and type of dishes (see fig 8).

While returning the search results, the additional personalization based on the layers and clusters computation, described within this section, may be applied.

Thus, the following personalisation-enhanced search scenarios are supported:

- Searching for a restaurant I may like, i.e., recommended by people with a similar profile.
- Searching for a restaurant that my friends from the Facebook recommended (criteria – author of the review).
- Searching for a restaurant that one specific persons (that I trust) likes.
- Hang-out (recommend a restaurant for n-number of *Taste It! Try It!* users).

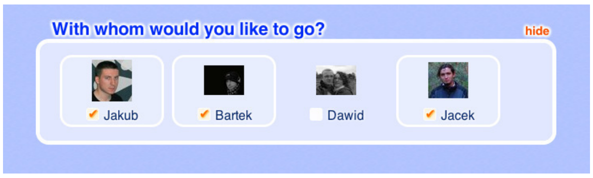

Fig. 9: *Taste It! Try It!*– searching for a restaurant – choosing friends to go with

Thus, in fact while searching, the personalized recommendation exploits both the knowledge base – information gathered by the *Taste It! Try It!* application and DBpedia (content-based approach), and the similarities between users (collaborative-filtering approach).

When it comes to specifying the semantic criteria, a user searches for it by typing characters in the corresponding text field, as indicated on fig 8. The application returns an auto-complete list of suggested concepts retrieved from DBpedia. Once the list has been populated, the user can select one (or more) of the suggested concepts.

V. Conclusion

The *Taste It! Try It!* application, presented in this paper, is a semantic content creation tool for a mobile device. It is to support users in the process of creation of semantically annotated reviews of various venues. It uses DBpedia as a source of data and is integrated with the Facebook portal.

As we are following the faceted-based approach to the review creation, we benefit from the additional information within the disambiguation process. For the needs of the *Taste It! Try It!* application, a distinct disambiguation solution has been designed, adjusted to the specific needs of a mobile device. Within this paper, we have shown how the application is consuming the Linked Data (while creating the reviews), and how additional semantic annotations (about the reviewed entities) are created. As they are to be made available outside the *Taste It! Try It!* application, they are anonymous and the subjective evaluation of the venue is expressed in an aggregated form. The tool also addresses the issue connected with the consensus reaching in case of contradictory information about a given venue provided by users.

In order to attract users and ensure their engagement in the semantic content creation process, a few incentives mechanisms have been applied. The creation of a user friendly and intuitive interface is one of them. In addition, the appropriate social incentives mechanisms taking advantage of Web 2.0 paradigm are to guarantee the appropriate quantity and quality (level of details) of the created semantic annotations. This will in turn allow to offer personalised and more accurate search possibilities of the application, thus, constituting additional incentive for users to use the application.

We have performed the evaluation of the proposed solution with 180 participants. We have gathered 2274 reviews on about 900 unique restaurants. The reviewed venue are located mainly in the Wielkpolska region of Poland. The gathered data helped us to design the feature allowing to reach a consensus in case of contradictory information. Results of the conducted experiments have shown that the design of application can have an important effect on the motivation of users to contribute content. We studied the effect of assignment of badges and the effect of the presentation of information on the willingness of users to add reviews.

Another important finding is the poor quality of information provided by DBpedia (mainly the lack of consistency regarding the structure to which concepts of the same type are being assigned). The coverage of DBpedia was deemed as unsatisfactory by most of the users. It does not necessarily result from the fact that the required concept is not present in the data gathered by DBpedia, but the concept could have been assigned to a not-intuitive place in the structure, which makes it difficult to be discovered.

After performed evaluation and testing, we argue that the developed application is showing that it is possible to create a user friendly application for a mobile device producing semantically annotated reviews and make this process as far as possible invisible for users; and that the applied incentives mechanisms can ensure appropriate quality and quantity of information included in the reviews and encourage users to more active behaviour.

REFERENCES

[1] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, vol. 284, no. 5, pp. 34–43, May 2001. [Online]. Available: http://www.scientificamerican.com/article.cfm?id=the-semantic-web

[2] M. Uschold and M. Grüninger, "Ontologies: principles, methods, and applications," *Knowledge Engineering Review*, vol. 11, no. 2, pp. 93–155, 1996. [Online]. Available: citeseer.ist.psu.edu/uschold96ontologie.html

[3] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. on Knowl. and Data Eng.*, vol. 17, pp. 734–749, June 2005. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2005.99

[4] E. Peis, J. M. Morales-Del-Castillo, and J. A. Delgado-López. (2008, May) Semantic recommender systems - analysis of the state of the topic. [Online]. Available: http://www.hipertext.net/english/pag1031.htm

[5] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data - the story so far," *International Journal on Semantic Web and Information Systems*, vol. 5, no. 3, pp. 1–22, 2009.

[6] T. Heath and E. Motta, "Revyu: Linking reviews and ratings into the web of data," *Web Semant.*, vol. 6, pp. 266–273, November 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1464505.1464600

[7] F. Burstein, *Research methods for students and professionals: Information management and systems*, 2nd ed. Wagga Wagga, Australia: Centre for Information Studies, Charles Sturt University, 2002, vol. 2, ch. Systems development in information systems research, pp. 147–158.

[8] W. Abramowicz, J. Dzikowski, A. Filipowska, M. Kaczmarek, and S. Lazaruk, "The semantic web's application for preparation of reviews – architecture and incentives for the needs of semantic content creation," in *Advanced Information Technologies for Management – AITM2011*, 2011.

[9] R. Burke, "Hybrid web recommender systems," in *The Adaptive Web: Methods and Strategies of Web Personalization*, Berlin, 2007, ch. 12, pp. 377–408.

[10] P. Pu, L. Chen, and R. Hu, "Evaluating recommender systems from the user's perspective: survey of the state of the art," *User Modeling and User-Adapted Interaction*, pp. 1–39, 2012, 10.1007/s11257-011-9115-7. [Online]. Available: http://dx.doi.org/10.1007/s11257-011-9115-7

[11] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, pp. 331–370, November 2002.

[12] H. Garcia-Molina, G. Koutrika, and A. Parameswaran, "Information seeking: convergence of search, recommendations, and advertising," *Commun. ACM*, vol. 54, no. 11, pp. 121–130, Nov. 2011. [Online]. Available: http://doi.acm.org/10.1145/2018396.2018423

[13] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, ser. UAI'98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 43–52. [Online]. Available: http://dl.acm.org/citation.cfm?id=2074094.2074100

[14] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Trans. Inf. Syst.*, vol. 22, pp. 89–115, January 2004. [Online]. Available: http://doi.acm.org/10.1145/963770.963774

[15] W. Yang, Z. Wang, and M. You, "An improved collaborative filtering method for recommendations' generation," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 5, oct. 2004, pp. 4135 – 4139 vol.5.

[16] M. Pazzani, D. Billsus, S. Michalski, and J. Wnek, "Learning and revising user profiles: The identification of interesting web sites," in *Machine Learning*, 1997, pp. 313–331.

[17] E. Peis, J. M. M. del Castillo, and J. A. Delgado-Lpez, "Semantic recommender systems. analysis of the state of the topic." *Hipertext.net*, vol. 6, p. (online), 2008. [Online]. Available: http://www.hipertext.net/english/pag1031.htm

[18] I. Cantador and P. Castells, "Multilayered semantic social network modeling by ontology-based user profiles clustering: Application to collaborative filtering," in *EKAW*, 2006, pp. 334–349.

[19] I. Cantador, P. Castells, and A. Bellogn, "An enhanced semantic layer for hybrid recommender systems: Application to news recommendation." *Int. J. Semantic Web Inf. Syst.*, vol. 7, no. 1, pp. 44–78, 2011. [Online]. Available: http://dblp.uni-trier.de/db/journals/ijswis/ijswis7.html#CantadorCB11

[20] M. Hausenblas, "Exploiting linked data to build web applications," *IEEE Internet Computing*, vol. 13, pp. 68–73, 2009.

[21] M. Zamarian, R. Cuel, O. Tokarchuk, M. Rohde, F. Yetim, J. Meurer, and M. Sein, "Deliverable 1.3.2 incentive models and pd guidelines (final version)," http://www.insemtives.eu/deliverables/INSEMTIVES_D1.3.2_Incentive_models_and_PD_guidelines.pdf, Insemtives, Tech. Rep., 2011.

[22] Z. Eidoon, N. Yazdani, and F. Oroumchian, "A vector based method of ontology matching," in *Proceedings of the Third International Conference on Semantics, Knowledge and Grid*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 378–381. [Online]. Available: http://dl.acm.org/citation.cfm?id=1338447.1338912

[23] R. Mirizzi, T. Di Noia, A. Ragone, V. C. Ostuni, and E. Di Sciascio, "Movie recommendation with dbpedia," in *3rd Italian Information Retrieval Workshop (IIR 2012)*. CEUR-WS, 2012. [Online]. Available: http://www-ictserv.poliba.it/sisinflab/publications/2012/MDROD12