Using Graph-Based Models in a Persuasive Social Recommendation System

Javier Palanca
Departamento de Sistemas Informaticos y Computacion
Universitat Politecnica de Valencia. Camino de Vera S/N
46022 Valencia (Spain)
jpalanca@dsic.upv.es

Stella Heras
Departamento de Sistemas Informaticos y Computacion
Universitat Politecnica de Valencia. Camino de Vera S/N
46022 Valencia (Spain)
sheras@dsic.upv.es

Javier Jorge
Departamento de Sistemas Informaticos y Computacion
Universitat Politecnica de Valencia. Camino de Vera S/N
46022 Valencia (Spain)
jjorge@dsic.upv.es

Vicente Julian
Departamento de Sistemas Informaticos y Computacion
Universitat Politecnica de Valencia. Camino de Vera S/N
46022 Valencia (Spain)
vinglada@dsic.upv.es

ABSTRACT

Nowadays, social networks have an enormous impact in the society generating a lot of useful information to be employed in new social applications. In this paper, we show how we have used a graph-based model to extract and model data in order to develop a Social Recommendation System which recommends recipes in a social network.

Keywords

recommender systems, data integration, social networks

1. INTRODUCTION

Over the last few years, social networks have changed activities performed by users on the Internet. Concretely, mere search and navigation activities over stored information have been evolved into a direct interaction with other users. Therefore, users have gone from being mere consumers of information to real producers. Due to the increasing number of users and information generated, coupled with the heterogeneity of the users themselves, their unpredictable behavior, and the dynamism of the network structure, users have to cope with a high degree of uncertainty when choosing who to interact to or what information to consume [1]. In order to decrease this uncertainty we will require tools that help users in their decision-making processes within the network. A possible solution is the use of recommendation systems [2] [3] which are capable of performing effective recommendations about what action users can take or what information they may consume. Some examples of this processes typically occur when a user has to select a seller at eBay, or when she has to select a book in Amazon or a hotel in TripAdvisor. However, due to its heterogeneity (users with unpredictable behavior) and dynamism (members joining and leaving the network at their will, consequently evolving the network's structure), the users of these networks continuously have to deal with uncertainty in any type of interaction. These issues make hard a direct employment of traditional recommender solutions.

Another important problem is the use of different information that must be treated in an uniformed way in order to improve the recommendation processes in social networks. When we need to use data from different sources, the data must be often significantly integrated before being used for other purposes. This requires making projections between the vocabularies used as patterns in different data sources and merging data (instances) of different sources. There exists mechanisms in RDF Schema and OWL to express such relationships between terms of vocabularies so that you can perform transformations between schemes easily. There are also work on how to generate these alignments in a more or less automatic way [4], [5]. The main challenge in data fusion is the resolution of conflicts when different values for the same property of an object is obtained. There exists works in this line in the database community [6] and the efforts on reconciliation of identities in the Web community is growing [7]. In this sense, we propose the employment of graph databases as a solution for the integration and fusion of domain-specific information and social information. Graph databases focus on the structure of the data, storing information as a network. Nowadays, data evolves to be more interlinked and connected and graph-based models allow to represent billions of nodes and relationships. One of the main reasons for choosing a graph database versus relational databases and NOSQL stores is their improved performance when dealing with connected data [8]. Following this approach, a Persuasive Social Recommendation System for recipe recommendation in a social network (called

SAC'15 April 13-17, 2015, Salamanca, Spain.
Copyright 2015 ACM 978-1-4503-3196-8/15/04...$15.00.
http://dx.doi.org/xx.xxxx/xxxxxxx.xxxxxxx

receteame.com) is presented in this paper. The proposed system allows the recommendation of recipes taking into account aspects like persuasion, similarity, friendship, trust, reputation and, similarity valuations between users' food tastes. The intelligent system proposed in this work allows information fusion in the sense that it integrates different types of information (domain specific information of several recipes and social information about users) and different agreement technologies (such as, argumentation, trust and reputation). Also, the information treated by the system is stored in graph-based models and transformed in arguable recommendations for the recommendation process.

The rest of the paper is structured as follow: section 2 presents related work; section 3 shows how the social data has been modeled using a graph-based approach; section 4 presents the proposed Recommendation System; and, finally, conclusions are explained in section 5.

2. RELATED WORK

Traditional recommender systems base their recommendations on quantitative measures of similarity between the user's preferences and the current items to recommend (i.e. content-based recommenders [9]), between the user's profile and the profile of other users with similar preferences (i.e. collaborative filtering recommenders [10]) and on combinations of both (i.e. hybrid recommenders [11]). However, [12] has stated the inability of current recommender systems to use the large amount of qualitative data available online to empower recommendations. Usually, recommender systems do not provide an explanation about the reasoning process that has been followed to come up with specific recommendations. Recommendations tend to come directly from the recommendation algorithm that runs the website and not from the acquaintances that a user has in his social network. However, this does not follow current trends on the Web, where discovering is becoming social and recommendations could be expected to come directly from acquaintances in a decentralised way. Moreover, people trust recommendations more when the engine can provide reasons for them [13]. Thus, what is understood as a good recommendation is changing from the one that minimises some error evaluation to the one that really makes people happier.

In addition, online recommender systems suffer from problems inherent to their use in complex social networks, where the number of users and/or items to recommend can be very high. In the case of collaborative filtering, for instance, the process for comparing two users with the aim of extracting their similarity requires that they have qualified the same objects, which can be unreaslistic in large social networks. Another major weakness of online recommender systems is their trustworthiness. In an open network with a large number of users is impossible to ensure that all views expressed are true opinions of users and there is no tampering with the resulting recommendations. In order to overcome these problems, it is necessary to embed a social layer in current recommender approaches, taking into account aspects such as the generation of arguments that support recommendations, reputation and trust. Therefore, there are a number of open challenges for the development of a new generation of recommender systems [12], such as *exposing underlying assumptions* behind recommendations, *approaching trust and trustworthiness* from the perspective of backing recommendations and *providing rationally compelling arguments* for recommendations. Our work involves a contribution in these areas, presenting a persuasive social recommendation system for recipe recommendation in a social network.

Currently, social networks are the substrate where "knowledge" is placed. Social networks generate a lot of information that can be very useful for recommendation processes. The future is to be able to capture, analyze and understand the relationships and information flow which are being produced on social networks. In order to do this, we propose to use Graph-Based Databases. These databases address a new research trend which consists on taking advantage of the complex and dynamic relationships that exist among data in order to generate new knowledge. This type of databases facilitates the understanding and analysis of the vast amount of data conveyed on social networks. Graph-Based Databases is not a completely new concept, and we can find some attempts to develop proprietary technologies for this kind of databases. However, recent developments have presented not proprietary solutions, as Neo4J[1], which will allow the widespread use of this technology. Moreover, over the last few years we can find in the literature some recommendation systems which are based on this kind of databases [8] with very promising results.

3. SOCIAL RECOMMENDER IN THE RECIPE DOMAIN

In this section we are going to present how information is retieved and fusioned in order to have a fast and highly available database that allows us to perform the recommendation process in an efficent and effective way. For our recommendation process we need to retrieve two types of data: social data from the users, and a big amount of recipes, with nutritional information about their health issues, diseases for which they are encouraged and discouraged, nutrients of each ingredient and their relationships with health and dietary labels. In what follows, we present how we have retrieved both social information and recipes information and how we represent this big data in a graph database.

3.1 Graph-Based models

Due to the users heterogeneity and dynamism, we have to manage an enormous quantity of data that is constantly generated in an efficient way and, thus, the traditional databases fail because of their most important drawback: the data consistency (ACID[2] philosophy). In addition, in an environment as interrelated as a social network, it becomes necessary to store the semantics that is underneath the interaction of the users. In other words, we are more interested in the relationships than in the content. For this reason, the potential need to include new information that may be related with hundreds of users or viceversa could imply an excessive computational cost.

In this scene is where the NOSQL databases, which are guided by BASE[3] principles, show their potential. Concretely, in our domain the entities can be interrelated among them, with a high growth and dynamism. Therefore, among a wide range of possibilities within the NOSQL databases, we will work with graph databases. This type of databases

[1] www.neo4j.org

[2] Atomicity, Consistency, Isolation and Durability

[3] Basically Available, Soft state, Eventual consistency

are highly scalable and close to the natural structure of the data which we will work with.

3.1.1 Social Information

In what follows, we explain the process that will be carried out by the user who wants to receive recommendations about recipes with the help of the application that we have developed. This task can be perform using the site *receteame.com*[4], which gives us an interface to interact with the user.

First of all, the user accedes to the site through the main page, where the user could register or log in with the user's Facebook account. When the user registers using Facebook, the process is carried out by means of OAuth 2.0 protocol[5], that is widely used on social networks and applications that work with them. This method facilitates the access to HTTP services in a restricted way. These limitations are imposed by the user, who provides permissions to the application in the register phase. In this case, the application requests some permissions like obtaining the mail account, birthday, or friends list, among others. This is a sensitive step of the process, due to the reluctance of users when they are requested for granting permissions related to their personal information. For this reason, we have payed special attention to how users react to our request, and we have obtained succesful results, getting those permissions in over an 80% of the cases (thus getting access to the inbox messages, shared links and tagged photos).

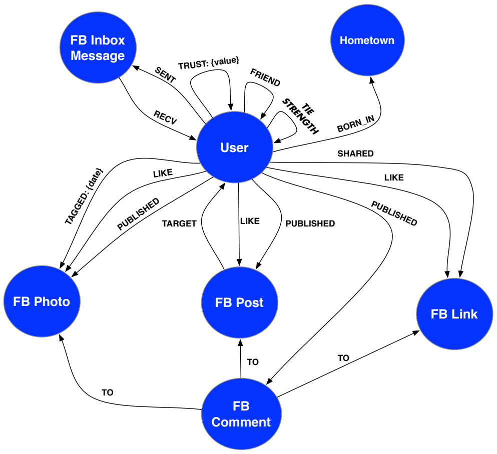


Figure 1: Graph of the user

Once the registration process has finished by means of the Facebook account, the application will get a token that identifies the relationship between the user and the application. With this token, the application can obtain the user's node that is stored in the Facebook's databases. This node enables the application to obtain the information about the user's activities in the social network, such as ”likes", comments or tags. Only the information related with the interactions in the social network will be stored in the application's database. The contents that the user has in the user's account, such as messages or posts, are not stored.

Figure 1 shows the information about the user that we have obtained in this process, which enables the algorithm BFF[15], by means of the relations established among the users (with comments, friendship, likes, posts and tags), to provide a *Tie Strength* value. This value establishes the degree of friendship between two users on the social network.

3.1.2 Recipe Information

Making recipe recommendations requires to have a big database of valid recipes that may result interesting to users. The bigger the database is, the better and more accurate recommendations can be delivered to users. Recommending recipes also presents a special characteristic; although people may be interested in world wide recipe discovering, they usually tend to be interested in local cuisine that uses products and ingredients that they can get easily. This is also known as *local food movement*, and tries to reduce the food miles, which gives rise to a more sustainable meal. This is also important with cooking techniques, because users tend to be interested in cooking what they can afford. For instance, it makes no sense to propose a recipe that requires a *tandoor oven* (a cylindrical clay or metal oven used in Southern, Central and Western Asia, as well as in the Caucasus) to users from Western Europe, since they have not easy access to these kind of cooking elements or techniques.

These considerations have guided us to build a recipe database focused in a unique country (Spain) with individual eating habits and a very rich cooking tradition. Spanish gastronomy is varied and rich, making use of lots of fresh ingredients like vegetables, fruit, fresh fish and shellfish, and ingredients that are country-specific while famous, like olive oil or iberic ham. These ingredients are accessible everywhere in Spain but are not as common outside the country.

There are three ways of building a good recipe database that accomplishes these characteristics: to hire nutritionists and chefs to write it, which is too expensive, to let users to upload their own recipes, which is too slow, and to crawl the web looking for the recipes. The latter was the method followed in this work.

In order to locate semantically correct recipes we made use of microformats [14]. Microformats (μF) is a semantic markup language that extends HTML with semantic tags that allow web pages to be processed automatically and extract data that contains semantic information intended for end-users. Two of the most known microformats for recipes are *h-recipe* and Schema.org Recipe. Both are very similar and use classes or `itemprops` to tag content. Some tageable information for recipes is *preparation time, recipe instructions, description*, or *ingredients*. Recipe microformats have also nutritional information that can be useful (when present) to classify recipes for users with special requirements.

Figure 2 shows the information that we have retrieved and stored in the graph database. Since relationships are very important in this data representation, graph databases result a very good solution for our problem, as we will see later with the information fusion problem.

The non-repeatable information is stored in the **Recipe**

[4] http://www.receteame.com; http://buscador.receteame.com
[5] http://oauth.net/2/

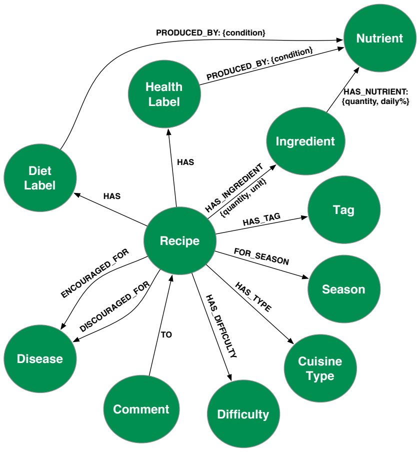


Figure 2: Recipes model graph

node. There we can found the recipe title, description, image, instructions, preparation time, servings, etc. On the other hand, all the information that may be useful to be represented as relationships has been modeled in separated nodes connected by relationships. Some of this information is the list of ingredients, health and diet labels, tags, difficulty, etc. Thanks to this graph representation it is easy (and quick) to find recipes that share a set of ingredients, or that are recommended for a specific allergy (looking at the encouraged and discouraged diseases relationships or the health labels). The nutritional information is also represented in this graph, connecting all the nutrients contained in an ingredient with the health labels and diet labels which are related to. This nutrient information has been extracted from the United States Department of Agriculture, who has published a National Nutrient Database for Standard Reference[6].

This crawling algorithm, based on microformats, has created a database of more than 10000 recipes that grows everyday. The growth rate is limited by the poor number of websites that use microformats (and even more limited if we focus only in recipes from a specific region). Microformats are very extended for calendar events (h-event) and people information (h-card), but are still not very much extended for the rest of prepared models, like h-recipe, h-product, h-item, etc, or the schema.org standards like Restaurant, Review, Product or Health.

3.2 Data Integration

Once explained how data have been obtained and organized following a graph-based model, the next step consists on the integration of both social and recipe information. Figure 3 shows how the integration has been defined. The use of the graph-based model allows a natural fusion of the obtained information, just adding the needed relationships among existing nodes. In the aforementioned figure there are represented the relationships that integrate this amount of data. Most number of relationships are between the *User* node and the *Recipe* node. These relationships between nodes mainly represent events that have occurred in the system. The relationship **VIEWED** represents that a user has viewed a recipe at a `date`. The same applies for the **RATED** relationship, which means that a recipe has been rated by a user with the rating `rate`. We can also see that there is another relationship between Recipe and User but, this time, in the opposite direction. This relationship represents the event that a recipe was recommended to a user at a `date`. Finally there are relationships that do not represent events but semantic information, like a disease that a user suffers, a diet label a user is interested in (i.e. `HIGH_PROTEIN`), or the user that has published a recipe.

One of the main advantages of using this approach is that the produced overload in the model due to the increase of the number of nodes is considered as negligible comparing with more traditional approaches. Therefore, scalability is not a problem.

One important feature of the proposed model is that there exists a clear unbalancement between the nodes needed to represent the social data and the recipe data. After the extraction process, the number of recipes were around 10.000 and the number of nodes needed to store the social information of users were around hundred thousands. Moreover, the dynamism of this kind of information will produce in the future an exponential evolution. Nevertheless, the proposed application based on this approach (which is presented in next section and is currently available in receteame.com) perfectly supports these orders of magnitude.

Another important issue is the information retrieval process for recommendation purposes. In this sense, the declarative language supported by this kind of models and the optimized storage facilitate a fast retrieval of the nodes and their relationships in order to follow the steps needed in the recommendation process. To show the expressivity of the query language, empowered by this graph-based technology, let us to propose an example where we want to get a recommendation based on the relationships between two users. Suppose that we want to make a recommendation to a user that has a disease (i.e. an allergy). Also, suppose that people who was born in the same town share eating habits. With all this information we are going to search for two users that share a disease, who were born in the same hometown and who have a tie strength (a friendship value) calculated between them, and get a recipe that was rated by this second user. Then, in the same query, we apply a filter to get only users with a high tie strength (more than 4.0 in this example). We also filter all the recipes that were already rated by the target user of the recommendation, in order to not recommend a recipe that the user already knows. And finally, we also filter all the recipes that are discouraged for any of the diseases that the target user has (not only the disease shared with user2), because we do not want to recommend a recipe that may be dangerous for our user.

This query example is shown in Listing 1. Having this information represented in a unique graph empowers us with the ability to create recommendations with relatively simple queries, as shown in this example. The next step in our approach will be to improve the recommendation process by introducing in the algorithm the social information with more components, like trust, reputation, content-based recommendation, etc., which we expect to be relatively simple thanks to the data representation presented in this work.

[6] http://ndb.nal.usda.gov/

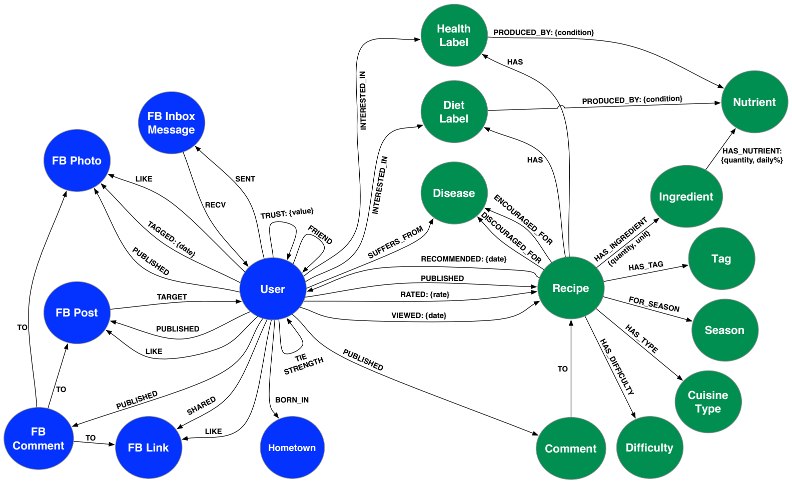


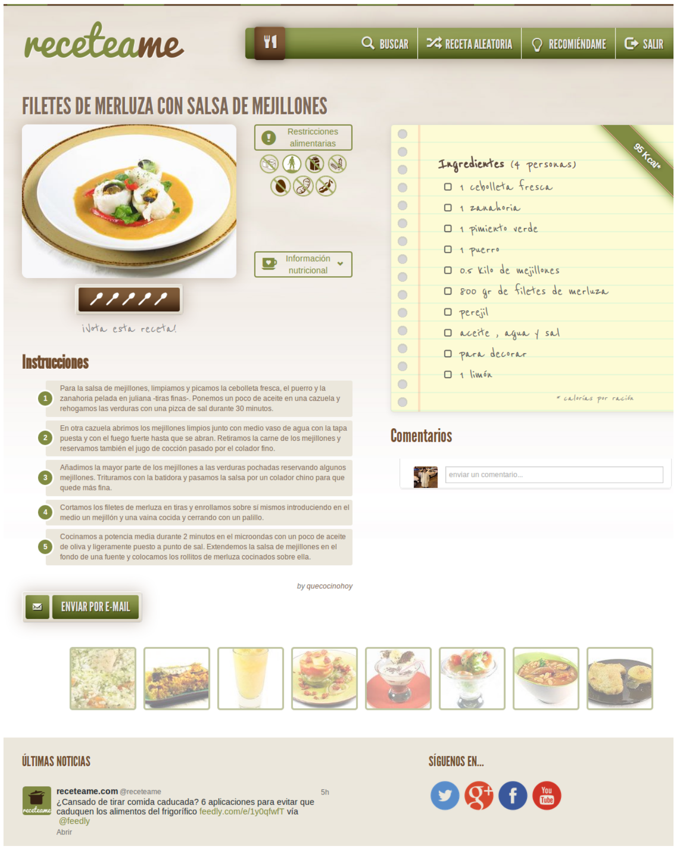
Figure 3: Integration graph model

Listing 1: Example Query

```
MATCH (user1:User)-[tie:TIE_STRENGTH]-(user2:User),
      (user1)-[:HAS_DISEASE]->(diseases:Disease)
      (user1)-[:HAS_DISEASE]->(:Disease)<-[:HAS_DISEASE]-(user2),
      (user1)-[:BORN_IN]->(:Hometown)<-[:BORN_IN]-(user2),
      (user2)-[rated:RATED]->(recipe:Recipe)
WHERE tie > 4.0 AND
      NOT (user1)-[:RATED]->(recipe) AND
      NOT (recipe)-[:DISCOURAGED_FOR]->(diseases)
RETURN recipe
ORDER BY rated.rate DESC LIMIT 1
```

4. APPLICATION

receteame.com[7] is a persuasive social recommendation system to make personalized recommendations about recipes to its users (figure 4 shows a screenshot of the website). The system retrieves recipes from the Internet (as commented in previous section) and automatically calculates their nutritional information and dietary restrictions. The site runs an intelligent algorithm (based on argumentation techniques and social network analysis) to learn the tastes and needs of each user and recommend fully customized recipes. *receteame.com* is able to learn the user preferences from two main sources of information: from the votes that users give to each recipe, and, from the activity of the user and friends in Facebook. Therefore, the system uses this information to make recommendations and propose new recipes that the user may like.



FILETES DE MERLUZA CON SALSA DE MEJILLONES

Ingredientes (4 personas)

□ 1 cebolleta fresca
□ 1 zanahoria
□ 1 pimiento verde
□ 1 puerro
□ 0.5 kilo de mejillones
□ 800 gr de filetes de merluza
□ perejil
□ aceite , agua y sal
□ para decorar
□ 1 limón

* calorías por ración

Instrucciones

1 Para la salsa de mejillones, limpiamos y picamos la cebolleta fresca, el puerro y la zanahoria pelada en juliana -tiras finas-. Ponemos un poco de aceite en una cazuela y rehogamos las verduras con una pizca de sal durante 30 minutos.

2 En otra cazuela abrimos los mejillones limpios junto con medio vaso de agua con la tapa puesta y con el fuego fuerte hasta que se abran. Retiramos la carne de los mejillones y reservamos también el jugo de cocción pasado por el colador fino.

3 Añadimos la mayor parte de los mejillones a las verduras pochadas reservando algunos mejillones. Trituramos con la batidora y pasamos la salsa por un colador chino para que quede más fina.

4 Cortamos los filetes de merluza en tiras y enrollamos sobre sí mismos introduciendo en el medio un mejillón y una vaina cocida y cerrando con un palillo.

5 Cocinamos a potencia media durante 2 minutos en el microondas con un poco de aceite de oliva y ligeramente puesto a punto de sal. Extendemos la salsa de mejillones en el fondo de una fuente y colocamos los rollitos de merluza cocinados sobre ella.

by quecocinohoy

Comentarios

enviar un comentario...

receteame.com @receteame 5h
¿Cansado de tirar comida caducada? 6 aplicaciones para evitar que caduquen los alimentos del frigorífico feedly.com/e/1y0qfwfT vía @feedly
Abrir

Figure 4: Screenshot of receteame.com

Registered users can obtain a recommendation that matches their user's taste and dietary restrictions using the persuasive social recommendation algorithm that implements the main recommendation functionality of the website. The algorithm receives a recipe recommendation query for a specific user, which includes parameters describing the user profile (preferences and tastes, dietary restrictions, personal data, etc.) and the context of the query (e.g. if the user is looking for a main course of for a particular ingredient, the number of dinner guests, etc.). With this information, the algorithm performs two main searches to select a potential set of recipes to recommend to the user. On the one hand, the algorithm follows a content-based recommendation approach to generate a list of recipes that match the query. However, note that the accuracy of recommendations generated by this process completely relies on the amount and accuracy of previous votes that the user made to other recipes with similar characteristics. Therefore, it is highly influenced by the *cold start problem* (i.e. the performance of content-based recommenders is poor with new users that have not yet rated a sufficient amount of recipes) and the drawbacks of applying traditional recommendation approaches on large social networks (i.e. issues related with computational costs of getting an accurate recommendation, and loss of the big amount of related social information available in the network). To overcome these problems, the algorithm performs an alternative search that follows a social recommendation approach.

On the other hand, the algorithm selects a set of users of the system and spreads the query to obtain recommendations from these users. This set contains the set of friends of the target user, and if necessary, a randomly selected set of users to avoid the cold start problem when the target user is new on the system and still does not have an adequate number of friends. Each user that has received the query selects a set of recipes that match the original query from his own set of known recipes (those voted by this user). Then, for each user, this part of the algorithm generates an ordered list of recipes to recommend according to two criteria: the preferences of the user that is being asked for recommendations, for instance, taking into account the votes of the user; and the preferences of the target user, for instance, taking into account the votes of the target user to a recipe (if any).

With the full set of recommended recipes from other users, the algorithm makes an overall ranking of recipes employing four social criteria parameters: 1) the trust on the user who had recommended a recipe from the point of view of the target user and his friends: this parameter is calculated by using a direct trust evaluation between these two users, and, if any, aggregating the trust evaluations of the friends of the target user that are also friends of the user that made the recommendation; 2) the reputation of the user who had recommended a recipe: this is a global parameter calculated by computing the average trust regarding all recommendations made by one user to his friends; 3) the strength of the friendship between the target user and the user that had recommended the recipe: this parameter is calculated by using several predictive friendship variables [15] and depends on the activity of the target user on the social network where the algorithm operates (e.g. Facebook); and 4) the similarity between the target user and the recommender user in terms of their preferences. The result of this process is a unique and ordered list of recipes to recommend to the target user. Finally, the algorithm mixes the recommendations that has obtained from both searches, assigning weights to bound content-based and social recommendations, and selects the best recommendation to propose. This process also includes an internal agreement procedure based on argumentation techniques, which allows the algorithm to promote those recommendations that come from users that are able to provide better justifications for their recommendations [16].

5. CONCLUSIONS

[7] http://www.receteame.com; http://buscador.receteame.com

This work presents a persuasive social recommendation system for the recommendation of recipes. The system is embedded in the web application *receteame.com.* For the implementation of this system, we have used a graph database which is highly scalable and close to the natural structure of the recipe data and users' social data that we work with. This approach is highly flexible and scalable, and the expected overload in the model due to the high dynamism of recipes and social information on the web (which produces a quick increase on the number of nodes) is considered as negligible comparing with more traditional approaches. In addition, the declarative language supported by our model and the optimized storage facilitate a fast retrieval of the nodes and their relationships in the recommendation process.

As future work, we are working on making the recommendation process faster and increasingly more accurate and on adding new functionalities to the website. We will also gradually introduce a more comprehensive dietary restrictions to include new diseases and intolerances. Furthermore, *receteame.com* may be able to recommend full menus in the future, such as a weekly menu for a family that takes into account the preferences of all members, or a menu for a dinner with friends where one can cook something with confidence that it will like to all guests.

Acknowledgment

This work was partially supported by MINECO/FEDER TIN2012-36586-C03-01 of the Spanish government.

6. REFERENCES

[1] Desel, J., Pernici, B., Weske, M. Mining Social Networks: Uncovering Interaction Patterns in Business Processes. *Business Process Management*, Berlin, vol. 3080, pp.244-260 (2004)

[2] Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering **17**(6) (2005) 734–749

[3] X. Zhou, Y. Xu, Y. Li, A. Josang, and C. Cox, "The state-of-the-art in personalized recommender systems for social networking," *Artificial Intelligence Review*, vol. 37, no. 2, pp. 119–132, 2012.

[4] Ehrig M., "Ontology Alignment: Bridging the Semantic Gap," *Springer*, 2007.

[5] Euzenat, J. and Shvaiko P. , "Ontology matching," *Springer, Heidelberg (DE)*, 2007.

[6] Bleiholder, J., Naumann, F. , "Data Fusion," *ACM Computing Surveys*, 41(1):1-41, 2008.

[7] Halpin, H., Thomson, H. , "Special Issue on Identify, Reference and the Web," *International Journal on Semantic Web and Information Systems*, 4(2):1-72, 2008.

[8] I. Robinson, J.Webber, and E. Eifrem, *Graph Databases.* O'Reilly, 2013.

[9] M. Pazzani and D. Billsus, *Content-Based Recommendation Systems*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2007, vol. 4321, pp. 325–341.

[10] J. Schafer, D. Frankowski, J. Herlocker, and S. Sen, *Collaborative Filtering Recommender Systems*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2007, vol. 4321, pp. 291–324.

[11] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[12] C. Chesñevar, A. Maguitman, and M. González, *Empowering Recommendation Technologies Through Argumentation.* Springer, 2009, pp. 403–422.

[13] G. Linden, J. Hong, M. Stonebraker, and M. Guzdial:, "Recommendation Algorithms, Online Privacy and More," *Communications of the ACM*, vol. 52, no. 5, 2009.

[14] Khare, Rohit and Çelik, Tantek, "Microformats: a pragmatic path to the semantic web" in *15th international conference on World Wide Web*, ACM, 2006, pp. 865–866.

[15] Fogués, Ricard L and Such, Jose M and et al, "BFF: A tool for eliciting tie strength and user communities in social networking services", in *Information Systems Frontiers*, Springer, 2013, pp. 1–13.

[16] S. Heras, V. Botti, and V. Julián. Argument-based agreements in agent societies. *Neurocomputing*, doi:10.1016/j.neucom.2011.02.022, 2011.