Ontology-based Recommendation for Points of Interest Retrieved from Multiple Data Sources

Ozer Ozdikis, Fatih Orhan, Furkan Danismaz
Anel Arge
Hacettepe Teknokent, 2. Arge Binasi
Kat:1, No:11, Beytepe, Ankara / TURKEY
+90-312-299 26 30
{ozer.ozdikis, fatih.orhan, furkan.danismaz}@anelarge.com

ABSTRACT
Introduction of powerful mobile devices and increasing availability of online services make it possible to develop a wide range of mobile applications. Making recommendations to the users on their mobile devices based on their location is a well-known application area of location based services. In this work we introduce an ontology based approach to find reasonable recommendations for sites (Points of Interest) like restaurants, hotels, and touristic places. We extend an existing OWL ontology in order to keep semantic relationships between different site types. Our application populates this ontology with site instances collected from several data sources, namely Google Maps, GeoNames, DBpedia and a local database. During this integration process, solutions for ontology mapping, site categorization, and duplicate site detection are developed. The ontology is then used to make recommendations on a mobile augmented reality application based on user's inputs on his device.

Categories and Subject Descriptors
D.3.1 [**Programming Languages**]: Formal Definitions and Theory – *Semantics*.

I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods – *Representations*.

General Terms
Algorithms, Design, Languages

Keywords
Ontology Reuse, Mapping and Merging, Location Based Services, Information Integration, Augmented Reality Applications

1. INTRODUCTION
With the introduction of stronger mobile devices in the last decade, available services to be provided gained a large diversity.

SWIM 2011, June 12, 2011, Athens, Greece.

Most of the mobile phones in the market today are sold with an integrated camera and they support internet connection through Wi-Fi or 3G interfaces. Location Based Services (LBS) is one of the application areas for such mobile devices. LBS aims to deliver relevant and timely services to the users by using their location information [7]. Another application area is mobile Augmented Reality (AR) applications [10]. Devices capable of combining many sensors (e.g. camera, GPS, accelerometer, magnetometer) with large displays and powerful processing units can merge the real world environment with virtual images. Combining overlaid annotations on top of the real world view can help the user see the points of interest in his surroundings.

Location Based Services and Augmented Reality applications can be combined in a single solution to produce a context-aware pervasive recommendation system. Such a solution provides the user with filtered relevant information in his surrounding and with an efficient human-device interaction. A simple scenario could be bringing several pharmacies together with the hospitals, if the user is looking for a nearby hospital. The device would show the user all such relevant Points of Interest (POI) as tagged items on the screen. A POI is a geographic entity with name, type, latitude and longitude information [1]. In the rest of the paper, the terms "POI" and "site" are used interchangeably.

In this work, we aim to gather site information from multiple data sources, classify them into previously defined categories, remove duplications and define the relevance relationships between these categories. Sites are collected into an OWL ontology which serves as a knowledge base to support the definition of semantic relationships and reasoning facilities. Based on these relationships, when the user selects a site on the screen of his device, our system provides him a group of closest and most relevant other sites as recommendations. Therefore, the context of the user will be the location gathered from the GPS sensor of the device and the selected site which indicates the user's interest.

There are two major problems studied in this work. The first one is about designing the ontology, i.e. identifying the required concepts and their relationships. The ontology structure must enable creation of site instances and specify their categories. For example, a specific restaurant must be created as an instance of "Restaurant" class in the ontology. Moreover, it must be possible to define the relationships and give relevance values among categories. For example, defining the relevance of "Hospital" with the "Pharmacy" must be supported by the ontology.

The second issue is how to populate the ontology with sites. This depends heavily on the data sources. In this work, we use Google

Maps[1], GeoNames[2], geo-tagged entries of DBpedia[3], and a database provided by a third party content provider as our data sources. There are cases to be handled while the data is collected from these data sources. One of them is the classification of sites. Different data sources have different category structures. Therefore mappings between categories in the data source and categories in the ontology have to be defined. There are also situations where a one-to-one mapping may not be defined easily. For example, a category in the data source can be so generic that it may correspond to several categories in the ontology. Moreover, it is possible that a site is assigned to more than one category in its data source, or it may not even have any category assignment at all. This is observed especially in the sites extracted from Google Maps. For such situations where a straightforward category mapping cannot place a site into the category structure of the ontology, a solution has to be designed to find out (or guess) the correct category in the ontology. In addition to the classification, the other problem to be solved while populating the ontology is duplicated site instances. There may be multiple entries for the same site in different data sources. Such duplications should definitely be avoided to end up with a useful application. Otherwise several items for the same site would be displayed on the screen, and this would definitely reduce user satisfaction.

The rest of this paper is organized as follows. We first present some related work in Section 2. Section 3 introduces the structure of our target ontology, while Section 4 describes how this ontology is populated with sites collected from different data sources. In Section 5, the augmented reality application we developed based on the resulting ontology is briefly demonstrated. Finally the concluding remarks and our future plans are presented in Section 6.

2. RELATED WORK

An ontology based recommendation system for mobile users is introduced in [12]. In this work, a new ontology is designed which keeps information about several types of shops (e.g. food, movie-theater) and the profiles of the users. The shop data in the ontology is populated by the users themselves. Recommendation is based on the user profiles and features of the shops. [3] and [4] are other examples of studies to integrate LBS and Semantic Web. In [3] a semantic web based tour guide system is developed on hotel domain. An ontology is defined to describe the properties of hotels, which is populated by crawling the web sites of desired hotels. An LBS middleware which semantically describes POI information is presented in [4]. The middleware includes an OWL-based description language named POIDL which is used by content providers to define POI instances. A taxonomy ontology is also designed to define the categories of the POI entities in a hierarchy.

An approach for semantic enrichment of places is introduced in [1]. The aim is to populate place ontologies in different domains. In other words, a system is developed to populate separate ontologies for restaurants, museums, pubs, travels and shows with specific POI instances. The approach is simply based on using web resources to extract information about POI's. Given a POI name and its city, a web search is run using the Yahoo API. Among the returning top web pages, most distinguishing terms are identified using an information extraction technique, namely TF-IDF. These terms are then used to identify the category of the POI. The accuracy depends heavily on the structure of the target ontology, results of the web search engine and the performance of the term extraction algorithm. The percentage of correctly categorized restaurants ranges from 41% to 94%, where this rate is between 0% and 13% for museums. This approach uses lexical and semantic resources in English.

The POI's we are interested in could be considered as concepts in the tourism domain. A comprehensive list of existing tourism ontologies and ontology tools in use are presented in [9] and [8]. The Qall-Me ontology, which was developed as part of a EU funded project (FP6 IST-033860), covers a large number of concepts and relationships in the tourism domain [8]. It is written in OWL-DL and contains almost all the site categories and properties we need in our work.

Using Linked Data for mobile augmented reality applications and problems regarding the integration of different data sources were discussed in 2010 in a W3C Workshop [10]. The need for the standardization of POI data, using semantic web resources like GeoNames and DBpedia, and establishing connections between different types of POI's taken from different data sources are mentioned in [10]. Issues to be solved are stated as avoiding information overload and defining concept mappings. Moreover which data sources to trust, how to solve performance issues and usage of SPARQL[4] queries are also noted as discussion points.

W3C formed a POI Working Group in October 2010, which aims to develop a specification to represent POI data in a specific format [11]. The objective is to provide a standard POI data format so that information providers can define and exchange their data. Moreover, it is stated that the primary focus of the group is the usage of POI data in AR applications.

We believe our work in this paper is very parallel with these activities of W3C, considering that we aim to gather POI data from open data sources on the web in order to develop a useful AR application. Collecting site information from two popular linked data sources (namely DBpedia and GeoNames) and Google Maps in a single category hierarchy with duplicate removal may contribute to these efforts. Establishing relationships between site categories and providing SPARQL query features are consistent with the discussions made in W3C Workshops. Moreover, our approach does not depend on a specific language. For example, the POI data that we used in our application is mostly in Turkish.

3. DESIGNING THE ONTOLOGY FOR RECOMMENDATION

The Qall-Me ontology defines a comprehensive list of OWL classes, including the classes for site types organized in a hierarchy. A site is created as an instance of the corresponding site class. For example, while populating the ontology, a specific hotel must be created as an instance of class *Hotel*. However, the Qall-Me ontology does not provide a method to define relationships between these site classes, i.e. categories. Moreover, it must be possible to define the strength of these relationships. In other words, strong relationship between a Hospital and a Pharmacy must be distinguished from a relatively weaker relationship between a Club and a Museum. In order to express these relationships, the Qall-Me ontology has been extended with a new

[1] Google Maps - http://code.google.com/apis/maps/index.html

[2] GeoNames - http://www.geonames.org

[3] DBpedia - http://dbpedia.org

[4] SPARQL - www.w3.org/TR/rdf-sparql-query/

class named *SiteCategory* and three object properties for the relationships, namely *stronglyRelated*, *related* and *weaklyRelated*. Instances of *SiteCategory* class are created for each site type (e.g. *SiteCategory_Hotel*, *SiteCategory_Restaurant*) and desired relationships are defined between these instances. Decision about which relationship type to establish between two site categories is made by the domain expert. A snapshot of the extended Qall-Me ontology together with the relationships defined for the category of Café is given in Figure 1.

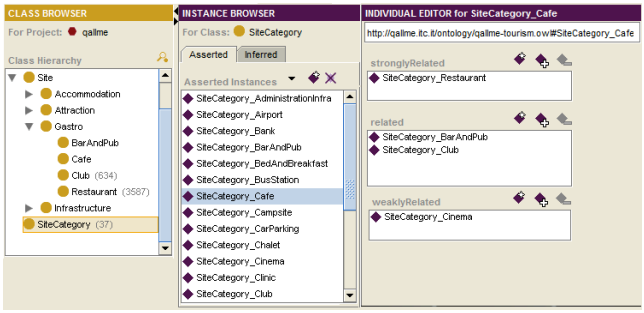

Figure 1 - Extension of Qall-Me Ontology with SiteCategory

As a result of this extension, while generating the list of sites related to a given site instance s_i, following operations are done in the given order:

- the class of s_i is found (e.g. *Café*),
- *SiteCategory* instance corresponding to the class of s_i is identified (e.g. *SiteCategory_Cafe*),
- Using the relationship properties of the identified *SiteCategory*, a list of "most related" *SiteCategory* instances are retrieved (e.g. *SiteCategory_Club*, *SiteCategory_Cinema,* etc…),
- *Site* classes corresponding to the *SiteCategory* instances are found (e.g. *Restaurant*, *Cinema,* etc…),
- Instances of these *Site* classes are returned as the related sites for s_i.

Type of the relationship between site categories is important in the third step, i.e. while finding the "most related" *SiteCategories* for a given *SiteCategory*. In this work, the strengths of these relationships are discretized in three levels, with some corresponding numerical weights. As mentioned before, these relationship types are *stronglyRelated*, *related* and *weaklyRelated*, and they have 1.0, 0.8 and 0.5 as numerical weights respectively. Relationships and their values are used to build a relationship graph for a given category. Figure 2 is an example relationship graph generated for *Café* as the starting node. In this graph, nodes are *SiteCategories,* edges between them are weighted relationships, and the numbers on nodes represent the strength of the relationship between that node and the starting node. Node values are calculated recursively. For example, since *Club* is *related* to *Café*, its value is 0.8. Since *Club* and *MetroStation* are *weaklyRelated*, the relationship value on Club (0.8) is multiplied by the weight for *weaklyRelated* property (0.5), resulting in 0.4 as the relationship strength between *Café* and *MetroStation*. While calculating the relationship value for a node, the maximum possible value is assigned to it if there are multiple paths between that node and the starting node.

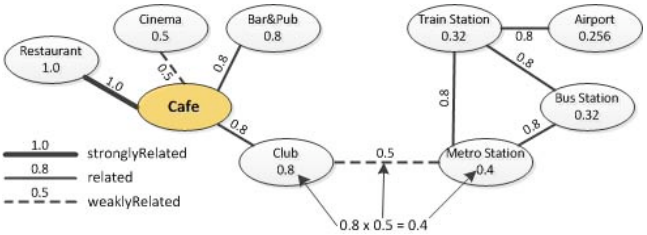

Figure 2 - Example Relationship Graph

The *stronglyRelated* property is slightly different than *related* and *weaklyRelated* properties. Since its weight is 1.0, all *SiteCategories* which are strongly related to each other can be considered as equivalent, always having the same relationship strength. Therefore it is defined as a transitive property.

It is worth mentioning that another possibility to define these relationships is using SWRL[5] rules with the support of a reasoning engine like Pellet[6]. Using SWRL, rules are defined in the conceptual level, which are applied on the individuals of the ontology. For example, to state that instances of class *Restaurant* are *stronglyRelated* to the instances of *Café*, an SWRL rule like

$$restaurant(s_i) \wedge cafe(s_j) \rightarrow stronglyrelated(s_i, s_j)$$

could be defined. As a result, by running SPARQL queries, it is possible to retrieve a list of related sites for any given site. One major drawback of defining relationships with SWRL is the performance. In our experiments with a few hundred *Restaurants* and *Cafés*, the response of a query based on SWRL rules takes almost a minute. On the other hand, the query for the relationships based on our method using *SiteCategories* runs in a few seconds even with a much larger amount of data.

In addition to the relationships between site categories, to be able to make reasonable recommendations, it is necessary for sites to have a visibility range. For example, the visibility range of a café can be 1 kilometer, while a world-famous landmark like Hagia Sophia or an airport can be visible from 5 kilometers. Without any filtering using the visibility range, very related but impractical recommendations can be made, since some sites may be too far from the user. Therefore another extension for the Qall-Me ontology is the definition of *visibilityRange* as a datatype property for the *Sites*. If the distance between the user and a *Site* is higher than the visibility range of the *Site*, it is not shown to the user.

Another minor extension we made to Qall-Me ontology is the addition of new site categories (e.g. *Museum*, *School* and *University)* and a datatype property named *datasource* which keeps an identifier for the data sources of *Sites*.

4. POPULATING THE ONTOLOGY

The extended Qall-Me Ontology is populated with the instances of *Sites* taken from several data sources. The only prerequisite for the *Sites* to be imported into the ontology is that they have a name (preferably in Turkish) and geographic coordinates (latitude and longitude). In this work, we focus on the sites in Ankara/Turkey.

[5] SWRL - http://www.w3.org/Submission/SWRL

[6] Pellet Reasoner - http://clarkparsia.com/pellet/

4.1 Data Sources

The first data source to introduce is a local database which is provided by a third party content provider. This database includes 3463 restaurants, 650 movie theaters, 4068 pharmacies, 634 clubs and 124 Cafes in Turkey. In addition to name and geographic coordinates, the content includes details such as address and contact data. These sites are grouped in relevant tables in the database. There are already *Site* classes in the ontology which can be mapped to each of these five tables. Therefore entries in tables are created as instances of corresponding *Site* classes in the ontology.

DBpedia, which extracts information from the Wikipedia pages, provides this data both in XML datasets and through a SPARQL endpoint. In this work we used DBpedia3.6 datasets generated from the Wikipedia dumps in November 2010. Since DBpedia keeps different types of information in different files, this process requires combining attributes in multiple files to form a useful dataset. For example, all geo-related entries are provided in a file with their unique id and geographic coordinates. If the description information is needed for these entries, another file for the descriptions has to be processed. The identification of these entries in different files is done using their unique id's. Using the datasets for coordinates and countries, more than 1700 geo-related entries are found in Turkey. However, there is no strict regulation for the categories of these entries. Some of them conform to the category hierarchy of DBpedia defined in an OWL ontology, while some of them may have unstructured labels. By mapping the categories in DBpedia ontology to the categories of the Qall-Me ontology, 8 museums, 25 landmarks, 55 airports, 24 train stations, 51 stadiums, 19 universities, 10 schools and 12 cultural heritages in Turkey are imported to the Qall-Me ontology.

GeoNames is a geographical database, containing more than 10 million geographical entities with their names, coordinates, categories and additional information like alternate names or population. GeoNames provides this information both in country-specific files and through web services. In this work, we used the dataset for Turkey from February 2011 which consists of almost 60000 geographical entities. There are 645 feature codes in GeoNames which are used as category identifiers. Each entity in the dataset has exactly one feature code. By mapping the relevant feature codes to the corresponding *SiteCategories* in the Qall-Me ontology, more than 1500 sites are created in the ontology. For example, 430 cultural heritages, 649 hotels, 367 train stations and 75 airports in Turkey are added to the ontology as the result of this process.

Google Maps provides an API which returns site instances for a given query string in a specific region. In this work we divided the region of Ankara in 50x50 meters of grids and sent keyword based queries to Google Maps API for each grid. These keywords consist of the most descriptive terms for our site categories, e.g. "restaurant", "hotel", "museum" and their Turkish synonyms. As the result of this process, around 16000 sites are collected. After the analysis we made on this data, 785 different categories are observed which are used to categorize these sites in the Google Maps environment. Moreover, these categories are entered by the community and especially old sites do not conform to a well-defined category structure. It is observed that 40% of the sites are assigned to a single category, 8% are assigned to multiple categories and the remaining 52% have no category information at all. Therefore, before these sites are created in the Qall-Me ontology, their correct *SiteCategory* have to be identified. In order to use the categories in Google Maps, a comprehensive subset of the 785 categories are manually mapped to the categories in the *SiteCategories* in the Qall-Me ontology. As a result, more than 6000 sites are uniquely assigned to a *SiteCategory*, and created as a *Site* instance of relevant type in the ontology. These sites include 2700 pharmacies, 700 restaurants, 768 schools, 783 banks, 382 shopping centers, 365 taxi stations and 65 hospitals. For the remaining 10000 sites, a classification method which is explained in the next section is applied.

4.2 Site Categorization

Correct categorization of the site instances is the key to make quality recommendations in our solution. Site categorization problem depends on the compatibility of the category structures in different data sources. In other words, data sources have different site categories; therefore it is not always possible to define a one-to-one mapping into the categories in Qall-Me ontology. Moreover it is not always guaranteed that a category is always provided for all sites by a data source. As a result a method is implemented for assigning a site to a *SiteCategory* in the Qall-Me ontology.

For such classification problems, Naïve Bayes is a widely used method based on simple probabilistic models [6]. In our classifier, the terms in site names are the only features to calculate posterior probabilities of categories. Sites that are already classified into a category make up the training set. By comparing the name of an unclassified site and names of the preclassified sites, the category with the highest probability is found and assigned to the unclassified site. The posterior probability of the site s_i belonging to a category c_k is found using the following Bayes equation.

$$P(c_k \mid s_i) = \frac{P(c_k) \cdot P(s_i \mid c_k)}{P(s_i)}$$

In this equation, $P(c_k)$ is the probability of the category c_k independent of s_i. It is calculated as the ratio of sites in category c_k to the total number of sites in the environment. $P(s_i)$ is the probability of the site. However this term is ignored since it is the same for all categories. $P(s_i|c_k)$ represents the probability of terms in the name of s_i to appear in c_k. For example, assume "restaurant" is a term which appears frequently in names of *Restaurants*. If a site whose name includes the term "restaurant" is to be classified, the category *Restaurant* must have a higher chance. The calculation of $P(s_i|c_k)$ is based on this idea [5]. If the name of the site s_i is composed of m terms represented as t_1, t_2,…t_m, then $P(s_i|c_k)$ is calculated using the following equation.

$$P(s_i \mid c_k) = P(t_1, t_2, ...t_m \mid c_k) = \prod_{j=1}^{m} \frac{n(s, t_j, c_k)}{n(s, c_k)}$$

In this equation, $n(s,t_j,c_k)$ represents the number of sites in c_k whose name includes the term t_j, where $n(s,c_k)$ is the total number of sites in c_k. Therefore it uses the ratio of the number of sites in c_k with term t_j to the total number of sites in c_k. On the other hand, it is possible that a site has a very unique name, which has no common terms with a pre-classified site to give a hint about its category. In such cases, the site is not classified to a category. Instead it is expected from the user to decide it manually. Moreover, we define a few keywords that are used to ignore some sites. For example the term "Inc." indicates a company name, which is not relevant in our application. Sites that include one of these terms are simply ignored.

Before using this classification algorithm on the sites retrieved from Google Maps, its accuracy is measured with sample data. To do this analysis, ontology is populated with the sites from GeoNames, DBpedia, local database and 90% of the sites from Google Maps which are uniquely classified as explained in the previous section. These sites constitute the training data set. The remaining 10% of the uniquely classified sites from Google Maps are used as the test data. The precision and recall values for five sample categories are given in Table 1. Except for the low recall for the Shopping Infrastructures, the accuracy values are considered to be satisfactory. One reason for the low recall values is the unclassified sites. As explained before, if a site in the test data has no common term with a previously classified site, it is not assigned to a category. This results in low recall values. Moreover, it is observed that some of the wrong classifications are due to the strong similarities between categories. For example a restaurant name may include the term "Café", which may cause it to be assigned to the category *Café*. Similarly, a cinema may be named after the shopping center it is located in, which may cause it to be assigned to the category *ShoppingInfrastructure*. Therefore, some of the categorizations which are considered to be wrong are actually acceptable categorizations. Therefore, the accuracy results of this categorization method are considered to be good enough to categorize sites taken from Google Maps.

Table 1 - Accuracy of the classifier

	Test Data Size	Precision	Recall
Restaurant	400	0.977	0.962
Pharmacy	600	0.993	0.995
School	80	1.0	0.950
Taxi	40	0.975	0.975
Shopping Infrastructure	40	1.0	0.525

This method is applied on the remaining 10000 sites from Google Maps which are assigned to multiple categories or which do not have a category. As a result, around 8000 sites are categorized and created as instances in the ontology. Remaining 2000 sites are left uncategorized, since they have either no common term with a previously classified site or they include a term that we defined in the "to-be-ignored" list.

4.3 Handling Site Duplicates

While reading similar data from multiple data sources, it is possible to receive duplicate entities. For example, in our case, a restaurant can be taken from the database and at the same time retrieved from the Google Maps. Their locations may not be identical. Moreover, there may even be spelling differences between their names. Our method to detect and remove duplicates is based on removing trivial terms from the site names and comparing the remaining terms (i.e. distinguishing terms) using string distance metrics. Before processing the site names, a preprocessing is applied on terms for word-stemming, and removal of stop words and punctuations. Trivial terms (like "restaurant", "pizzeria") are found by calculating the frequency of each term in all site categories. If the frequency of a term exceeds a threshold (in our case, if it appears in more than 3% of the sites), it is accepted as a trivial term. After these terms are removed from the site names, the remaining terms are expected to be the distinguishing terms. While looking for a duplicate of a site s_i, all neighboring sites of same type in the range of 100 meters are found. The distinguishing terms of s_i and the distinguishing terms of its neighbors are compared using the Damerau–Levenshtein distance [2]. If the string distance is below a certain threshold, the terms are assumed to be the same. In our application, $term_x$ and $term_y$ are accepted as the same terms if one insertion, deletion, substitution of a character or transposition of two adjacent characters is enough to produce $term_x$ from $term_y$. The similarity of these terms is used to decide on the similarity of sites. If all distinguishing terms of s_i are found in the distinguishing terms of s_j (or vice versa), s_i and s_j are supposed to refer to the same entity. An example for the application of this algorithm is illustrated in Figure 3. In this example, site A (with name "Schnitzel Rest." taken from Google Maps) is matched with sites B and C. Despite the spelling mistake in the name of site C (sHCnitzel), it is successfully matched with site A because of the string distance method we applied. As the result of this process, 275 duplications are found among 7000 restaurants in the ontology. This method is applied on all site categories. For example, 1270 of 8000 pharmacies and 36 of 1600 schools are identified as duplicates. While removing the duplicates, the reliability of the data sources is taken into account. According to our observations, our local database contains the most reliable site data. After our local database, GeoNames and DBpedia are found to be more reliable than Google Maps.

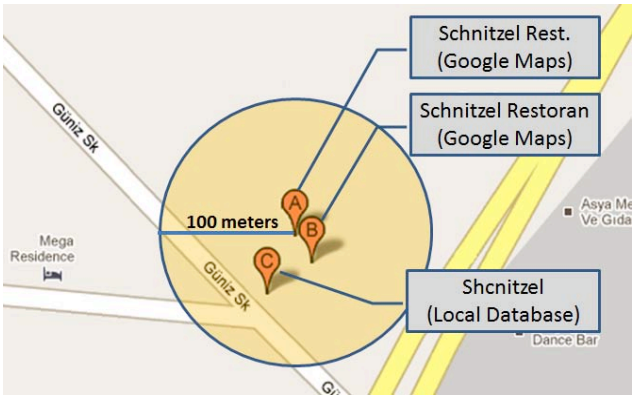

Figure 3 - Site Matching

5. APPLICATION

After the extension of the Qall-Me ontology and creation of the site instances, the ontology contains more than 21000 entities mostly in Ankara from very different domains, like Hotels, Cafés, Airports, Schools, Hospitals and Museums. Visibility ranges of sites are set during their creation. For example visibility range of all restaurants is set to be 500 meters, while visibility range of all airports is 5000. Moreover relationships between *SiteCategories* are established similar to the example for *SiteCategory_Café* in Figure 1.

These sites and their relationships in the ontology are used in an augmented reality application to make recommendations to the user. The application we developed reads the sites from the ontology, and based on their coordinates, it places icons on camera view of the device. It is successfully deployed and run on iPhone-iOS and Android operating systems. A screenshot from the application is given in Figure 4.

Figure 4 - Example Result on AR Application

The example in Figure 4 shows the recommendations generated according to user's position, selection and sites in his surroundings. In this example scenario, the user selects a Café, and our application recommends him 4 Cafés, 3 Restaurants, a Bar and a Cinema. This result is consistent with our example graph in Figure 2.

6. CONCLUSION AND FUTURE WORK

In this work, we aim to provide a complete solution to integrate different site types from different data sources with different data structures. This data is aggregated in an OWL ontology, which is extended from an existing tourism ontology, namely Qall-Me. The extension includes the definition of relationships between site categories. During the information integration process, a Naïve Bayes classifier is used to categorize unclassified sites. A method based on string edit distance is developed for identifying duplicate site definitions. Finally a comprehensive ontology with 21000 sites is generated. These site instances, together with their coordinates, visibility ranges, categories and relationships are then used in an AR application to make recommendations to the user based on his location and choices. We believe the results of our work may contribute to the efforts of W3C in the standardization of POI data.

In order to describe the relationships between *SiteCategories*, we define three properties in the ontology, namely *stronglyRelated*, *related* and *weaklyRelated*. We believe it would be very useful if the next versions of OWL specification supported association of numeric values to the OWL properties. If this were possible, just one property with numeric weights would be enough to define the relationship between *SiteCategories*. In other words, properties would be instantiated with a numeric weight which may be considered as a reliability value for the relationship. For example *Café* would be *relatedTo Club* with weight 0.8, or *relatedTo Cinema* with weight 0.5.

As the future work, we are planning to improve the recommendation method by featuring it with detailed site information, user profile and user context. Detailed site information includes additional features for sites, like the opening-closing hours of a museum, type of food served by a restaurant and price range for the rooms of a hotel. Although some of these attributes may be acquired from some data sources (for example Google Maps has a category for pizza restaurants), most of this data is supposed to be maintained manually. Qall-Me ontology provides the required fields to define these attributes. These attributes will enable the creation of user profiles. During the recommendation process, the context of the user (e.g. time of the day and user status) will be taken into account as well.

7. ACKNOWLEDGMENTS

Our thanks to Dr. Hakan CAGLAR, Dr. Halit OGUZTUZUN, and Dr. Pinar SENKUL for their reviews and feedbacks.

8. REFERENCES

[1] Alves, A., Antunes, B., Pereira, F.C., and Bento, C. 2009. Semantic Enrichment of Places: Ontology Learning From Web. *International Journal of Knowledge-based and Intelligent Engineering Systems*, volume:13, no:1, pages:19-30

[2] Damerau, F. J. 1964. A Technique for Computer Detection and Correction of Spelling Errors. *Communications of the ACM*, volume:7, no:3, pages:171–176

[3] Kim, J.W., Kim, J.Y., Hwang, H.S., and Kim, C.S. 2005. Location-Sensitive Tour Guide Services Using the Semantic Web. In *Proceedings of the 9th International Conference on Knowledge-based Intelligent Information and Engineering Systems (KES 2005)*, Lecture Notes in Artificial Intelligence, volume:3682, pages:908–914

[4] Kim, J.W., Kim, J.Y., and Kim, C.S. 2006. Semantic LBS: Ontological Approach for Enhancing Interoperability in Location Based Services. In *Proceedings of On the Move to Meaningful Internet Systems (OTM'06) Workshops*, Lecture Notes in Computer Science, volume:4277, pages:792–801

[5] Lewis, D. D. 1998. Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. In *Proceedings of the Tenth European Conference on Machine Learning*. pages:4-15, Germany

[6] Mitchell, T., 1997. *Machine Learning*, McGraw Hill

[7] Mohapatra, D., and Suma, S. B. 2005. Survey of Location Based Wireless Services. In *Proceedings of the IEEE international conference on personal wireless communication (ICPWC)*, pages:358–362

[8] Ou, S., Pekar, V., Orasan, C., Spurk, C., and Negri, M. 2008. Development and Alignment of a Domain-Specific Ontology for Question Answering. In *Proceedings of the Sixth International Language Resources and Evaluation LREC'08*, (Marrakech, Morocco, May 2008)

[9] Prantner, K., Ding, Y., Luger, M., Yan, Z., and Herzog, C. 2007. Tourism Ontology and Semantic Management System: State-of-the-arts Analysis. *IADIS International Conference WWW/Internet 2007*, (Vila Real, Portugal, October 2007)

[10] Reynolds, V., Hausenblas, M., Polleres, A., Hauswirth, M., and Hegde, V. 2010. Exploiting Linked Open Data for Mobile Augmented Reality. In *W3C Workshop: Augmented Reality on the Web (June 2010)*, www.w3.org/2010/06/w3car/exploiting_lod_for_ar.pdf

[11] W3C POI Working Group Charter, http://www.w3.org/2010/POI/charter, last access time: March 2011

[12] Yu, Y., Kim, J., Shin, K., and Jo, G.S. 2009. Recommendation System Using Location-Based Ontology on Wireless Internet: An Example of Collective Intelligence by Using 'Mashup' Applications. *Expert Systems with Applications*, volume:36, issue:9, pages:11675-11681