Smartmuseum

Personalized Context-aware Access to Digital Cultural Heritage

Tuukka Ruotsalo[1], Eetu Mäkelä[1], Tomi Kauppinen[1], Eero Hyvönen[1], Krister Haav[3], Ville Rantala[1], Matias Frosterus[1], Nima Dokoohaki[2], and Mihhail Matskin[2]

[1] Semantic Computing Research Group (SeCo), Helsinki University of Technology and University of Helsinki
first.last@tkk.fi

[2] Department of Electronics, Computer and Software Systems (ECS), School of Information and Communications Technology (ICT), Royal Institute of Technology (KTH), Stockholm
{nimad,misha}@kth.se

[3] Apprise Ltd., Tallinn, Estonia
krister@apprise.eu

Abstract. This paper presents a semantic recommender method and a system for a personalized access to digital cultural heritage through context-aware user profiling. Given annotation knowledge-bases, explicit background knowledge in the form of ontologies, a user model capturing the user's behavior and context, the system produces recommendations. Ontology-based user profiling can be used to reduce cold-start, sparsity and over-specialization problems. In addition, we present a recommendation retrieval method that is based on the vector space model and uses indices that enable fast and scalable implementation of the system.

1 Introduction

Digital cultural heritage collections contain mutually interrelated data that are provided for users through digital libraries. Semantic web technologies have enabled semantically interlinking of such collections [25, 14] and made possible to not only search individual collections, but also to browse, visualize and recommend objects across heterogeneous collections.

Recently, ubiquitous systems have gained popularity and we are witnessing an increasing number of users accessing web-based digital services using mobile devices. This opens up new possibilities for digital libraries that preserve descriptions of physical collections, such as museum databases, archaeological archives and tourist information catalogues. In addition to digital access to catalogued content, ubiquitous systems can guide the users to find objects in physical environments.

Recommender systems are able to assist the user to find contents that are likely of interest for the user [1, 27]. Unlike search systems, recommender systems enable information access without an explicit query given by the user. This is particularly important in ubiquitous user scenarios, where the usability of the mobile devices often limits the user's willingness to perform complicated search tasks. Especially, personalized recommender systems that compare the user's profile to reference characteristics, and predict the relevance that a user would give to an object they had not yet seen, are beneficial and reduce the need of interaction with the device. However, minimizing the user interaction imposes the problems of automatically matching the relevant content according to the user's profile and context.

In ubiquitous scenarios it is possible to limit the recommendations according to user's context information. In addition, user's behavior can be tracked to adapt the profile that can further limit the possible objects offered for the user. However, recommender systems suffer from the following problems [27, 5]:

1. *Cold-start.* Many of the users visiting museums are first-time visitors and their user profiles have limited data about the particular museum being visited. Therefore, users cannot immediately benefit from the recommender system unless it can generalize over similar contexts and content. In addition, the users must be able to manually construct and edit the profile.
2. *Sparsity.* The descriptions of the cultural heritage objects originate from heterogeneous collections and can be described with different structures, vocabularies and levels of granularity. The limitation in interlinking can lead to poor recommendations. Therefore, the recommendation methods should be able to benefit from semantic background information that provides richer interlinking and reduces sparsity.
3. *Over specialization.* The user profiles are constructed based on the features of the objects in the collections. This can lead to over specialization where users are recommended the objects that are too similar to the ones the user has already seen in the past. Therefore, the recommendation method must be able to generalize over similar but not necessarily equal features.

In this paper we demonstrate how background knowledge, semantic reasoning, query expansion and context aware user profiling can be used to overcome the above problems. The rest of the paper is organized as follows. In section 2 we will present an overview of the Smartmuseum recommender system. Section 3 describes the content creation and storage. Section 4 presents the user and the context models, and the method to construct the models for user profiling. Section 5 defines the recommendation method. Finally, we discuss the contributions, related work and future work.

2 Smartmuseum Recommender System

Smartmuseum is a system for enhancing on-site personalized access to digital cultural heritage through adaptive user profiling. Here we present the knowledge-based recommender system of Smartmuseum that uses context-aware user modeling and semantic data processing to provide recommendations. The functioning of the Smartmuseum recommender system can be illustrated through four main scenarios:

1. *Outside user scenario:* the user is provided information about interesting visiting sites or museums.
2. *Inside user scenario:* the user enters a visiting site or museum and is provided information about objects that could be in interest of the user.

3. *User administrative scenario:* the user enters user profile information either manually or tags objects to express interests on-site.
4. *Curator administrative scenario:* the system curator enters a new object into the system.

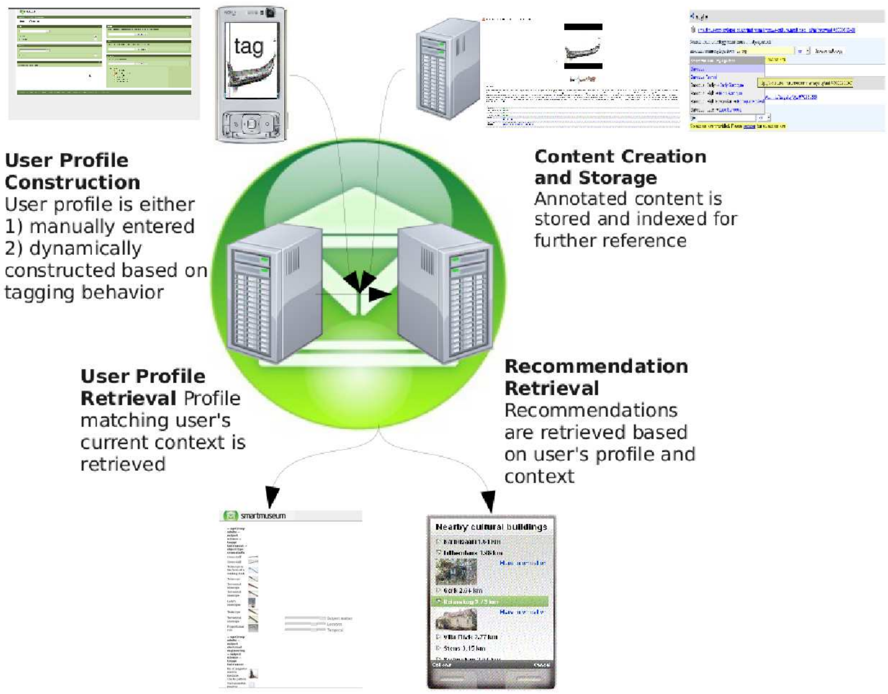

Fig. 1: System-level features of the Smartmuseum recommender system.

The outside scenario is dependent on the location of the user determined by the GPS of a mobile device or a map interface of a web browser. The location information forms the context of the user. This context is combined with the user's personal profile to recommend sites of interest that are close to the user's current location. When the user finds a suitable visiting site and wants to know more on what to do at the specific site, the user can click on the user interface to indicate the entrance to the visiting site. This switches the mode of the system to the inside scenario.

In the inside scenario, the user's context is attached with the information of being inside a certain visiting site or a museum. The user profile is then matched to objects limited to the ones available on the site.

In the user administrative scenario, the user is able to tag the recommended objects using an "I like" tag or an "I dislike" tag. The metadata of the objects marked with these tags are further used to construct and adapt the user's profile. The user is also able to manually enter profile information using a web interface. In the curator administrative scenario, the digital library or museum curator is able to enter new objects to the system. The objects are then stored and indexed using semantic reasoning techniques for further reference.

These main level scenarios can be divided into four system-level features illustrated in Figure 1. A User can enter the user profile manually or it can be constructed automatically based on the user's tagging behavior. In the recommendation phase, the user profile is first retrieved based on user's current context. Recommendations are retrieved for the user in either an inside or an outside scenario. To support the administrative scenario, the system curator is able to enter new objects to the system for further reference. The objects are stored and indexed in a triple-store that enables query expansion and efficient retrieval.

3 Content Creation and Storage

3.1 Ontologies, Metadata Schemas and Annotations

The metadata describing objects are called annotations. Annotations are stored and indexed to enable efficient retrieval in the recommendation phase. The actual objects can be regular HTML pages or images and stored decentralized. The annotations are stored in the Smartmuseum annotation base and can further be used to give reference to the HTML pages or images. Annotations are described in RDF(S) language [6].

In this study, the system was tested with two different dataset on cultural heritage domain: Finnish cultural heritage data from CultureSampo portal[4] and Smartmuseum data from Heritage Malta[5] and Institute and Museum of the History of Science in Florence[6]. Both datasets consist of lightweight ontologies and semantically annotated data corresponding to a Dublin Core compatible metadata schemas. CultureSampo dataset is indexed with the KOKO ontology[7] developed in FinnONTO project [15]. Smartmuseum data is indexed with Getty Vocabularies[8]. The ontologies used are light weight ontologies that are transformed to RDF(S) from thesauri, where concepts are organized in subsumption hierarchies. Geographical instances are structured in meronymical hierarchies that represent geographical inclusion. Geographical instances are geo-coded using the W3C Geo Vocabulary [9].

The aggregated knowledge base of CultureSampo dataset contains 133,735 objects and the Smartmuseum data consist of 1000 objects. The knowledge base consists of over 1.3 million RDF triples. The content is enriched using reasoning, resulting in some 12 million triples.

[4] http://www.kulttuurisampo.fi/
[5] http://www.heritagemalta.org/
[6] http://www.imss.fi.it/
[7] http://www.yso.fi/onto/koko
[8] http://www.getty.edu/research/conducting_research/vocabularies/
[9] http://www.w3.org/2003/01/geo/

3.2 Annotation Storage and Indexing

Smartmuseum requires fast retrieval of recommendations and dynamic query expansion depending on the user's profile and context. Indexing is inevitable to enable such functionalities. To support these requirements, we built four indices:

1. *Annotation Index* stores the original annotations and is built by storing all triples for each of the objects. The purpose of the index is to support fast retrieval when building a Triple-space index. Annotation index is also used to retrieve the information of the object to be shown in the user interface. We use a matrix of objects times triples. This enables further access either by querying using triples, or querying using object URIs.
2. *Ontology Index* is an index that stores the subsumption and part-of hierarchies. These are used for on-line query expansion and building the Triple-space index. For each resource in the domain, we store the superclasses of the resource in subsumption and part-of hierarchies. We use a matrix of resources times resources. For example, for resource Helsinki, we store all it's ancestors in the part-of tree: Uusimaa, Finland, Scandinavia, Northern-Europe, Europe and the Earth.
3. *Triple-space Index* is an index that stores the annotation with respect to transitive subsumption and part-of relations, i.e. triples originating from the annotations based on reasoning. This index makes use of the annotation index and ontology index. We use a matrix of objects times triples. For example, for triple $<$*sm:Object1, sm:manufacturedIn, place:Helsinki*$>$ we store all combinations reachable by reasoning, i.e. $<$*sm:Object1, sm:relatedTo, place:Helsinki*$>$, $<$*sm:Object1, rdf:Property, place:Helsinki*$>$, $<$*sm:Object1, sm:manufacturedIn, place:Finland*$>$, ... , $<$*rdf:Resource, rdf:Resource, rdf:Resource*$>$.
4. *Spatial Index* is an index that stores positioning information about objects. In our data set, positioning information is represented in two different ways: either 1) directly using WGS84 as a reference datum for values of properties *geo:lat* and *geo:long* from the W3C Geo Vocabulary, or 2) indirectly by a reference to an location described in ontologies which then refers to WGS84 values either directly or through it's ancestor concepts. For each annotation, we store a reference to a WGS84 point using latitude lat and longitude lon. This indexing is also done for annotations that refer to coordinates indirectly through instances in the geo-ontology. This ensures fast retrieval of the objects based on the WGS84 coordinate information. In addition, we store coordinate location of every location in the ontology.

Annotation index is designed to support the other indexing tasks and Ontology index to support on-line query expansion. Therefore, an indexed annotation can now be represented with the Spatial index and the Triple-space index. An annotation of an object is a set of triples and coordinates attached to an object. More formally, a object D consist of a set of triples $t \in T$ and set of coordinates c. Triples belong to the Cartesian product of the resources R in the domain, $T \in R \times R \times R$. Each coordinate c has latitude $lat \in [-180, 180]$ and longitude $long \in [-180, 180]$. Therefore, an object D is a pair $D =< t, c >$, where t forms a vector space of triples. The vector space model (VSM) [22] is further used for the retrieval of the objects.

All of the indices are implemented using Apache Lucene[10]. We store all the indexes in separate fields, but in a single final index. This enables evaluating all retrieval criteria in the same query evaluation task, which is essential in real applications. This is due, that result sets for the Triple space index can be very large and in worst case all of the annotations stored in the annotation base. Therefore limiting the result set using spatial criteria must be done in the same query evaluation task than the triple space matching. This is expected to cut down the search space significantly in real life retrieval cases.

4 User Profiling

A user profile is a collection of data about interests associated to a specific user. We support ontology-based user profiles and our model for representing user profiles consists of a set of RDF triples. This is important to enable matching the user profiles to the annotations. The profile triples are formulated using resource identifiers from ontologies. The rationale behind this is that triples can be matched to annotations using ontology-based reasoning and query expansion. Positioning information is mapped to geographical location identifiers of the ontology using the Spatial index.

The model used to represent a user profile is context aware. This means that the context where the tagging is performed can be attached to the actual triple that describes the user's interest. Smartmuseum currently supports only location context, but the context model is designed to use triples and therefore any context information can be deployed. Each triple in the user profile can be attached to a context. The context is represented using a set of triples.

More formally, a user profile P consist of a set of profile entries pi. Profile entries consist of a triple $t \in T$ and a context triple $ct \in T$ that belong to a Cartesian product of the resources R in the domain, $T \in R \times R \times R$. Each triple t is attached with a context in which it was added. For each triple in the profile entry pi in the user profile, there exists a weight $w \in [-1, 1]$. A User can create a user profile either manually using a web user-interface or dynamically based on the tags that the user explicitly adds for the visited objects.

4.1 Manual User Profile Construction

A user profile can be created manually using a web user-interface shown in Figure 2. The user is able to add concepts that the user likes or dislikes. The selections can be made from three facets: general concepts, persons and places. Each concept can be selected either by using an auto-completion search or faceted browsing based on the subsumption and part-of hierarchies of the ontologies. The rationale is to support initial profile creation using web interface and further adapt the profile to the user's interests based on the user's behavior. The concepts that are manually inserted to the profile are expanded as triples. Each concept is placed as an object of the triple and *rdf:Property* is used as the predicate and *rdf:Resource* as the subject. This is because the user interface is required to be simple and only initial profile is required to be inserted using the

[10] http://lucene.apache.org/

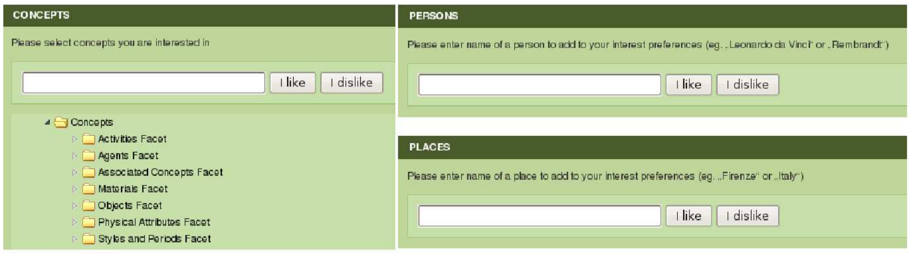

Fig. 2: Web user-interface for manual profile editing. Users can add explicit concepts that they like or dislike. The selection can be made using faceted browsing based on the subsumption hierarchies of the ontologies (left). In addition, the selections can be made from three auto-completion widgets: general concepts, persons or places (person and places widgets shown right).

web user-interface. The annotations are binary predicates, where the subject is always the identifier of the object and the object is the actual value describing the annotated object. Therefore, placing the concept only as the object of the triple and using the most general property is suitable and guarantees the match to all required resources in the recommendation retrieval phase. The triples in the manual profile construction phase are added with an empty context.

4.2 Dynamic Profile Construction

User profile can also be constructed dynamically based on the tagging behavior of a user. When the user sees an interesting object inside a museum, the user is able to tag the object with an ”I Like” tag or an ”I dislike” tag. The user profile is then updated accordingly. Each triple in the user's profile is attached to a context in which the tagging of the object, to which the triple was attached to, was performed. For example, consider a case where user has tagged an ”I like” tag for a *painting* that is annotated to have a *renaissance* as a *style period* , and the tagging has taken place in Italy. In our example, this would result to a following triples to be inserted as a user profile entry: t $=<$*sm:painting,sm:stylePeriod, koko:renaissance*$>$, ct $=<$*rdf:Resource, sm:userLocation, place:Italy*$>$, $w = 1$.

The problem with direct estimation like this is that user has indicated a preference of Italian paintings, however, the contexts in which the observations are done can be very sparse. Therefore, we use Laplace (i.e. add one) smoothing [18] to shave a share of the probability mass to contexts for which no observations are available. In this way, we can observe some probability for every triple even if it has not been tagged in the specified context.

5 Recommendation Retrieval

Recommendation retrieval is performed in five phases. First, the user profile is retrieved based on the user's context. This results into a weighted set of profile triples. Second,

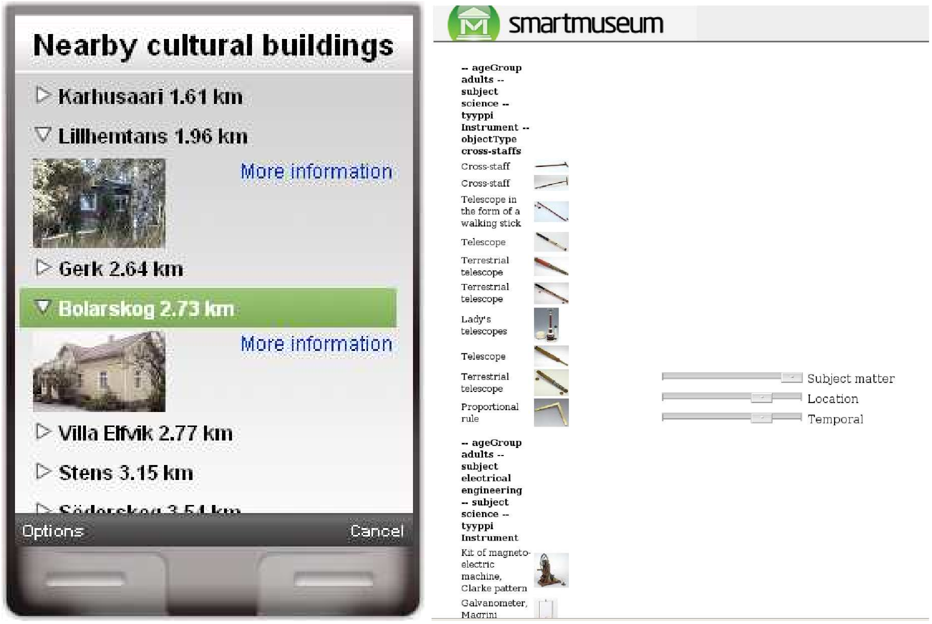

Fig. 3: The mobile user interface (left) illustrates the outside scenario, where user is recommended cultural sites close the the user's current location. The web browser user interface (right) represents the inside scenario, where the user is recommended objects inside the museum that match the user's profile.

the user's current context is created based on the location of the user. This gives spatial restrictions for objects to be recommended and includes mapping the user's location using spatial index to an ontological representation that is used in the user profile. Performing these two phases, we are able to formulate a user profile that corresponds to the user's current context and specifies spatial restrictions that can be used in the recommendation retrieval phase. Third, the query-time context information that are offered for user as slider controls in the user interface (see web browser user interface in Figure 3) are used to perform on-line query expansion and weighting. Fourth, the user profile is used to retrieve the relevant objects from the annotation base. Finally, the best results are clustered to provide different viewpoints to the recommendation results. Each cluster is explained in terms of the matching triples that the objects in the cluster contain.

5.1 User Profile Retrieval

Retrieving the user profile based on the user's context is done by mapping the user's location, determined by GPS or the user pointing a map in an user interface, to ontological concepts using the spatial index. This results into an ontological resource representing the position information. The weighted user profile is then retrieved based on the context. The weights for the triples can be observed from the tagging behavior of the user. We use the likelihood of a context generating a certain triple. It can be observed from the relative frequencies of the profile entries. Formally, $P(t|ct) = \frac{count(t|ct)}{count(ct)}$. For example, if a user profile contains tags for triples about Italian paintings in the context of Helsinki, say 10 times, and triples in Helsinki in total 20 times, the $P(<\textit{sm:Painting, sm:manufacturedIn, place:Italy}> \mid <\textit{rdf:Resource, sm:userLocation, place:Helsinki}>)$ would be $10/20 = 0.5$. In addition, we have the negative or positive votes for the triple. We calculate the average of the votes of the triple in the given context and multiply it with the probability of the triple in the context.

5.2 User Context Creation

Constructing spatial constraints corresponding the user's current context heads to limit the matching of the possible objects to be recommended to those that are close to the user's current location. We create constraints to retrieve nearby content by expanding the location information of the user. The received coordinate point is expanded to cover a circular area within the radius r. A sole bounding box is created where each of the edges of the box have a distance r from the point p, where the user is located i.e. the distance to the edges of the bounding box from the given p is r. The distance r can be adjusted in the user interface. We set the possible interval to be from 100 meters to 50 kilometers.

5.3 Query Expansion and Weighting

On-line query expansion and weighting means to expand the triples t retrieved in a first phase by using ontological reasoning. We provide slider controls, shown in Figure 3 for the user to control the following dimensions: subject matter, location and temporal. Here, the subject matter slider reflects to triples predicated by *dc:subject* or any of it's sub-properties, the location slider reflects to triples predicated by the spatial coverage predicate or any of it's sub-properties and the temporal slider reflects to triples predicated by temporal coverage predicate or any of it's sub-properties. For example by moving the subject matter -slider to a lower position, we perform query expansion to retrieve also content that is more general than defined by the triples in the profile. This means not only matching more specific cases of the triple (i.e. in case of Finland, match also Helsinki), but also more general cases (i.e. in case of Finland, match also Scandinavia). This query expansion up in the hierarchy is done by calculating the Wu-Palmer measure [28] for each resource of the triple and by accepting all triple combinations of resources that have a Wu-Palmer value below the value given by the slider. In addition of the query expansion, the weight of the triple in case is weighted according to the value given by the slider. Here, the ontology index is used to make fast on-line calculation of the Wu-Palmer measure.

Intuitively, this means that by lowering the value given by the slider, users indicate the dimension less relevant for their current information need. In this way, users can allow more radical query expansion and at the same time give lower value for the dimension specified by the slider. On the other hand, users can set a certain dimension more important, and triples under this dimension will be used in retrieval strictly based on user's profile and with higher weight.

5.4 Recommendation Retrieval

Recommendation retrieval is performed by using the query constructed from user profile and context. Based on the earlier phases we have a set of profile triples t that each have weight w. Each triple may be expanded using query expansion to multiple triples, that each have the weight of the original triples. The weight of each resulting triple is then multiplied by the value given by the slider. As a result we have a set of triples $t \in T$, each triple having a weight w. In addition, we have a spatial constraint, that defines the lower and upper bound for latitude and longitude. Finally, we can define the retrieval as a two step matching procedure that utilizes the spatial constraints and a scoring function used to calculate the cosine similarity [23] in vector space model [22] that we have generalized for the triple space:

1. Prune the search space based on the spatial index, such that the spatial constraints hold.
2. For the remaining matrix, calculate cosine similarity between vectors of index triples it in triple-space index and triples t in the profile using the Apache Lucene tf-idf scoring function: $score(t, it) = \sum tripleMatch(t, it)$, where

$$tripleMatch(t, it) = \begin{cases} (tf(t) \cdot idf(t)^2 \cdot w) & \text{when } t \equiv it \\ 0 & \text{otherwise} \end{cases},$$

   where tf is a triple frequency of a triple t given by $tf = freq(t)^{1/2}$, $idf(t)$ is an inverse triple frequency given by $idf(t) = 1 + log(\frac{N}{N_t+1})$, where N is the number of all objects and N_t is the number of objects, where t appears and w is the weight determined for the triple t.

5.5 Clustering

The recommendation results are returned as a ranked list by the retrieval method. While the ranking of the objects is important, to avoid over specialization, users may also want to receive recommendations from the different viewpoints specified in their user profiles. Therefore, we cluster the top recommendations and show the user 10 highest ranked objects from each cluster.

The clustering is based on the matching triples collected for each of the top 300 recommended objects given by the retrieval method. We use the FastICA algorithm to perform independent component analysis [13][11]. Independent component analysis is a computational method for separating a multivariate signal into subcomponents supposing the mutual statistical independence. We construct a concept combinations times objects matrix. The following combinations of the matching triples are used as concept combinations: subject, object, subject and object, predicate and object, and the full

[11] An Java implementation of FastICA (http://sourceforge.net/projects/fastica/) is used.

triple. Then, the principal component analysis is run for the matrix to reduce dimensions. Eigenfilter with filtering percentage of 98 is used in PCA. This means that after PCA sorts the eigenvalues, the first highest eigenvalues, whose sum is higher than 98 percent of all of the eigenvalues are used in the actual FastICA algorithm.

For FastICA, we set the number of desired clusters to 10. This is the maximum number of clusters returned. However, the algorithm determines less clusters if no further meaningful separation can be made. We used the hyperbolic tangent (a=1) as a contrast function. Clusters were obtained based on the highest absolute value of each object from the returned component vectors. Finally, the clusters were labeled by including the labels of the five most common concepts occurring in the cluster excluding concepts that occur in all of the clusters.

5.6 User Interfaces

The Smartmuseum recommender system is implemented with two separate user interfaces: web browser based interface and mobile phone interface shown in Figure 3. The mobile user widget reads the user's location context from the mobile device and retrieves matching content. In the example showed in the figure, the user's location is Espoo in Finland, and the user is known to be outdoor based on user's own indication. The web user interface demonstrates a situation where the user has indicated to be inside a museum. The web interface shows the results corresponding to the user's profile. The results are clustered into two clusters. The first cluster contains objects meant for adult visitors, have a subject science and are cross-staffs. The second cluster contains instruments that have subject electrical engineering and are also meant for adult visitors. Using the web interface, the user is able to adjust the recommendations on three dimensions using sliders as explained before: subject matter, location and time. Adjusting the sliders different matching triples may be determined and therefore also different clusters may be formulated. In the current system sensors are not included to identify if the user is entering the museum or if the user is looking some particular artwork. Therefore user is required to indicate the rating of the object or the entrance to a museum by clicking a link in the user interface.

The current version of the system has been implemented for a web browser and with partial functionality for Symbian S60 mobile devices using the S60 Web Runtime (WRT)[12] The mobile system acts as a "push" service and automatically updates recommendation data by polling the server based on a configurable time interval. The recommendation engine is implemented in Java and uses Apache Lucene[13] for indexing and cosine similarity calculation. All of the data is modeled in RDF(S)[14].

[12] http://www.forum.nokia.com/Technology_Topics/Web_Technologies/Web_Runtime/. The application is functional in S60 mobile devices that have the positioning support.
[13] http://lucene.apache.org/java/docs/
[14] http://www.w3.org/RDF/

6 Discussion

6.1 Related Work

Recommender systems have been used in a number of different applications such as web browsing [8], recommending books, music [17], movies [19] and news [16].

The current recommendation methods can be divided into a demographic technique [20, 26], collaborative filtering [16, 21], case based filtering [7] and content based filtering [24, 20]. Knowledge-based recommender systems [9, 11] are a type of content-based filtering.

Many existing cultural heritage portals support knowledge-based recommendation [25, 14] and mobile recommender systems have been implemented in the cultural heritage domain [2, 4]. The CHIP demonstrator [27] provides a personalized access to museum collections using ontology-based annotations and knowledge-based recommendation. A similar approach has been applied to provide recommendation support in e-tourism [10]. DBPedia Mobile enables map-based visualization and semantic search on DBPedia dataset [3]. Recently, many researchers have actively tried to leverage a Semantic Web in support of context-aware and personalized recommendation. For example, MyCampus is a Semantic Web environment for context-aware mobile services [12]. It provides interfaces to perform privacy aware queries and to access information from external web services.

6.2 Contributions

This paper presented knowledge-based recommender system Smartmuseum. The contributions of our approach are threefold. First, we present a generic user profile model that is not based on pre-defined schemata, but is a simple set of weighted triples. Second, user profiling used is context-aware and takes advantage of contextual information, such as user's current location, to further improve personalization. Third, the knowledge representation and retrieval methods of Smartmuseum are based on metadata schemas and large domain ontologies. This allows generalization for heterogeneous distributed collections described with different vocabularies and different levels of granularity. We describe a recommender method that is able to benefit from generic context-aware user profile, operates on smart indices and enables on-line performance of context-dependent query expansion. The recommendation method is demonstrated in a prototype application that operates on a knowledge-base of over 130.000 objects and has both mobile and web user interfaces.

Content- and knowledge based recommender systems often suffer from over-specialization [1]. This means that the user is being recommended only the objects that are similar to the ones she rated highly in the past. However, if the objects are too similar to something the user has already seen, such as a different news article describing the same event, objects should not be recommended. In Smartmuseum, query expansion and clustering can be used to reduce over-specialization.

In case of content- and knowledge based recommender systems, the user has to rate a sufficient number of objects before a content-based recommender system can really understand user's preferences and present reliable recommendations. This is often referred as a cold-start problem. We tackle the problem by offering a web interface, where users can manually edit and insert new profile information. In this way, the user is also able to avoid non-relevant information to be included to user's profile. In addition, the background knowledge is used to reduce the sparsity of the data.

Although formal evaluation of the system has not yet been conducted, the demonstrator system that tackles the problems presented above has been built and intuitively gives relevant recommendations. However, further evaluation of the methods and usability of the user interfaces are required.

Smartmuseum only uses location as a context and the performance of the context-based profiles are not studied in detail. Also the profile entries are independent and therefore modeling conditional dependencies of multiple context parameters is not supported. The system may also suffer from restricted positioning because it relies only on GPS.

In addition, Smartmuseum recommender system has only limited content analysis capabilities and is most useful in domains where content information can be extracted automatically or like in our case where it has been provided manually by museum curators.

6.3 Future Work

Our intention is to extend the work to the following directions. First, we will deploy automatic detection of user's location inside a museum based on external sensors. We will also provide RFID based identification for individual objects inside the museums. Second, we will investigate how collaborative filtering techniques could be incorporated with our knowledge-based recommendation method. Such hybrid approach would benefit from annotated content and ontology-based reasoning, but would be able to capture the knowledge originating from collaborative behavior of the users. Third, our intention is to extended the system to use other external information sources that can provide context information.

Acknowledgements

SMARTMUSEUM (Cultural Heritage Knowledge Exchange Platform)[15] is a Research and Development project sponsored under the Europeans Commission's 7th Framework (FP7-216923).

References

1. Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1):103–145, 2005.

[15] www.smartmuseum.eu

2. Bashar Al Takrouri, Karen Detken, Carlos Martinez, Mari Klara Oja, Steve Stein, Luo Zhu, and Andreas Schrader. Mobile holstentour: contextualized multimedia museum guide. In *MoMM '08: Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*, pages 460–463, Linz, Austria, 2008. ACM.
3. Christian Becker and Christian Bizer. DBpedia mobile – a location-enabled linked data browser. In *Proceedings of Linked Data on the Web Workshop (LDOW2008)*, Beijing, China, 2008.
4. Victoria Bellotti, Bo Begole, Ed H. Chi, Nicolas Ducheneaut, Ji Fang, Ellen Isaacs, Tracy King, Mark W. Newman, Kurt Partridge, Bob Price, Paul Rasmussen, Michael Roberts, Diane J. Schiano, and Alan Walendowski. Activity-based serendipitous recommendations with the Magitti mobile leisure guide. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1157–1166, Florence, Italy, 2008. ACM.
5. Jonathan Bower and Silvia Filippini-Fantoni. Personalization and the web from a museum perspective. In David Bearman and Jennifer Trant, editors, *Proceedings of Museums and the Web*, Washington DC / Arlington VA, USA, 2004. Archives and Museum Informatics.
6. D. Brickley and R. V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema W3C Recommendation 10 February 2004. Recommendation, World Wide Web Consortium, February 2004.
7. Derek Bridge, Mehmet H. Göker, Lorraine McGinty, and Barry Smyth. Case-based recommender systems. *Knowl. Eng. Rev.*, 20(3):315–320, 2005.
8. Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.
9. Robin D. Burke, Kristian J. Hammond, and Benjamin C. Young. The FindMe approach to assisted browsing. *IEEE Expert: Intelligent Systems and Their Applications*, 12(4):32–40, 1997.
10. Angelo Corallo, Gianluca Lorenzo, and Gianluca Solazzo. A semantic recommender engine enabling an eTourism scenario. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 1092–1101. Springer, 2006.
11. Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Zanker. An integrated environment for the development of knowledge-based recommender applications. *Int. J. Electron. Commerce*, 11(2):11–34, 2006.
12. Fabien L. Gandon and Norman M. Sadeh. Semantic web technologies to reconcile privacy and context awareness. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(3):241 – 260, 2004.
13. Aapo Hyvärinen and Erkki Oja. A fast fixed-point algorithm for independent component analysis. *Neural Comput.*, 9(7):1483–1492, 1997.
14. Eero Hyvönen, Eetu Mäkelä, Mirva Salminen, Arttu Valo, Kim Viljanen, Samppa Saarela, Miikka Junnila, and Suvi Kettula. Museumfinland–finnish museums on the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(2-3):224 – 241, 2005. Selcted Papers from the International Semantic Web Conference, 2004 - ISWC, 2004.
15. Eero Hyvönen, Kim Viljanen, Jouni Tuominen, and Katri Seppälä. Building a national semantic web ontology and ontology service infrastructure—the FinnONTO approach. In *Proceedings of the European Semantic Web Conference ESWC 2008*, Tenerife, Spain, June 1-5 2008. Springer.
16. Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. GroupLens: applying collaborative filtering to Usenet news. *Commun. ACM*, 40(3):77–87, 1997.
17. Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.

18. Christopher D. Manning and Hinrich Schuetze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1 edition, June 1999.
19. Bradley N. Miller, Istvan Albert, Shyong K. Lam, Joseph A. Konstan, and John Riedl. Movielens unplugged: experiences with an occasionally connected recommender system. In *IUI '03: Proceedings of the 8th international conference on Intelligent user interfaces*, pages 263–266, Miami, Florida, USA, 2003. ACM.
20. Michael J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artif. Intell. Rev.*, 13(5-6):393–408, 1999.
21. Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens: an open architecture for collaborative filtering of netnews. In *CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, Chapel Hill, North Carolina, United States, 1994. ACM.
22. G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.
23. Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523, 1988.
24. Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Analysis of recommendation algorithms for e-commerce. In *EC '00: Proceedings of the 2nd ACM conference on Electronic commerce*, pages 158–167, Minneapolis, Minnesota, United States, 2000. ACM.
25. Guus Schreiber, Alia Amin, Lora Aroyo, Mark van Assem, Viktor de Boer, Lynda Hardman, Michiel Hildebrand, Borys Omelayenko, Jacco van Ossenbruggen, Anna Tordai, Jan Wielemaker, and Bob J. Wielinga. Semantic annotation and search of cultural-heritage collections: The MultimediaN e-culture demonstrator. *J. Web Sem.*, 6(4):243–249, 2008.
26. Manolis Vozalis and Konstantinos G. Margaritis. On the enhancement of collaborative filtering by demographic data. *Web Intelli. and Agent Sys.*, 4(2):117–138, 2006.
27. Yiwen Wang, Natalia Stash, Lora Aroyo, Peter Gorgels, Lloyd Rutledge, and Guus Schreiber. Recommendations based on semantically enriched museum collections. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):283 – 290, 2008. Semantic Web Challenge 2006/2007.
28. Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138, Las Cruces, New Mexico, 1994. Association for Computational Linguistics.