World Wide Web (2011) 14:377–405
DOI 10.1007/s11280-011-0116-6

A social approach to context-aware retrieval

Stefano Mizzaro · Luca Vassena

Received: 14 July 2010 / Revised: 10 January 2011 /
Accepted: 24 January 2011 / Published online: 3 February 2011

Abstract We present a general purpose solution to Web content and services perusal by means of mobile devices, named Social Context-Aware Browser. This is a novel approach for information access based on users' context, that exploits social and collaborative models to overtake the limits of the existing solutions. Instead of relying on a pool of experts and on a rigid categorization, as it is usually done in the context-aware field, our solution allows the crowd of users to model, control, and manage the contextual knowledge through collaboration and participation. To have a dynamic and user-tailored context representation, and to enhance the process of retrieval based on users' actual situation, the community of users is encouraged to define the contexts of interest, to share, use, and discuss them, and to associate context to content and resources (Web pages, services, applications, etc.). This paper provides an overall presentation of our solution, describing the idea, the implementation, and the evaluation through a benchmark based methodology.

Keywords mobile · context-aware retrieval · middleware · web services · social

1 Introduction

Context-aware computing is a computational paradigm that has faced a rapid growth in the last few years, especially in the field of mobile devices. A key role in this new approach is played by the notion of context, that is roughly described as the situation the user is in. The information provided by the context can be exploited to improve the capabilities of mobile devices, adapting them to the user's needs. We

S. Mizzaro · L. Vassena (✉)
Dipartimento di Matematica e Informatica, Università degli Studi di Udine,
via delle Scienze 206, Udine, Italy
e-mail: luca.vassena@uniud.it

S. Mizzaro
e-mail: mizzaro@uniud.it

can imagine, for example, a user seeking information on the Web, by means of her mobile device, while moving around a town. In a normal situation, the user has to manually interact with search engines, making explicit her information need into a query and filtering out the not relevant retrieved resources. Exploiting the context, on the contrary, the user's query can be automatically refined and the retrieved documents can be filtered. Going further, contextual data can allow to predict the user needs and to proactively seek and retrieve information, thereby providing the right information in the right place at the right time and reducing the complexity of the user-device interaction. This scenario is even more striking when taking into account that, nowadays, the Web is not used as an information storage only. On the contrary, often the mobile user is searching for some service (e.g., buying something, finding a route on a map) and these services are being provided in many case by combining in a mash-up more basic services.

We propose an approach to proactive context-aware retrieval of Web contents and access to Web services on mobile devices, where contextual data are exploited to capture the dynamic nature of the user needs, of the information available, and of the relevance of this information, typical of a mobile user moving in the real world. Following this approach, we develop the *Social Context-Aware Browser* (SCAB): it is an extension of a previously presented framework, the Context-Aware Browser [4], and shares with it two basic objectives. First, it aims at discovering *"the query behind the context"*: to retrieve what the user needs, even if she did not issue any query [13]. Second, it is not a domain dependent application, but a generic way of interaction and information access, able to adapt to every domain. Differently from the Context-Aware Browser, however, the SCAB is based on models for context-awareness that aim at overtaking the limits that strictly bound existing approaches, Context-Aware Browser included. In particular, the SCAB is based on the social dynamics at the basis of Web 2.0, and exploits social computation to increase retrieval effectiveness.

This paper is structured as follows. We first briefly survey related work (Section 2), presenting the context-aware retrieval field, describing the previous framework our solution is based upon, and introducing the main ideas behind the social Web. We then propose our solution (Section 3), showing the motivations behind it and the overall conceptual model. In Section 4 we present the implementation approach. We then propose our evaluation methodology based on a benchmark (Section 5): we present the procedure followed, the evaluation goals, and the results. Finally we draw some conclusions and we sketch future work (Section 6).

2 Related work

Related work involves information retrieval based on context, the Context-Aware Browser, mobile services, and Web 2.0.

Information retrieval (IR) is the science of searching for unstructured documents, or for information within these documents, from large collections, with the aim of satisfying an information need. IR is an old discipline, but in the last ten years it has gained new importance because of Web: in fact, the constant increase of the number of Web pages and available documents calls for tools for easily retrieving information. Modern IR systems, or Web search engines, like Google, are these tools. Documents and information needs are the main elements of the IR model:

the information needs are expressed through a query, that is matched with an index of the documents, and the most relevant ones are retrieved. A similar approach is Information Filtering (IF): in this case, the user does not actively search for information (pull approach), but information is automatically delivered to users (push approach). IF is content based, if the content of documents is analyzed, or social, if the relationships among users in a community are exploited to filter resources and to deliver only the relevant information.

Context-aware Retrieval (CAR) is an extension of classical IR and IF that incorporates the contextual information into the retrieval process, with the aim of delivering to the users information that is relevant within their current context [8]. CAR systems deal with the acquisition of context, its understanding, and the application of behaviour based on the recognized context [19]. Thus the CAR model includes, among the classical IR model elements, the user's context, that is both used in the query formulation process and associated with the documents that are candidates for retrieval. For example, knowing that it is dinner time and that the user is in a town she has never visited, the system can automatically provide the list of the restaurants in the town that best match her taste. In the same way, if it is raining the system can show just restaurants in the nearbies. Examples of mobile CAR frameworks are the following: AmbieSense [15], Physical Mobile Interaction [1], Ubiquitous Web [11], and MoBe [13].

More generally, in [2] a model-driven approach for adaptive context-aware Web applications is presented. This solution fully covers the design and the development of context-aware Web applications through automatic code generation, based on a consolidated CASE tool for Web application modeling, where the stress is on the importance of user-independent, context-triggered adaptivity actions. The importance of a context-aware access to information is also witnessed by an increasing research on low level protocols to optimize the content delivery in mobile wireless networks, as done in [3]: a context-aware transmission process is proposed, where contents are fragmented into several frames and only the frame of interest for the client are downloaded.

A more general system is the Context-Aware Browser [4], a general-purpose solution to Web content perusal by means of context-aware mobile devices. The main idea behind it is to empower a generic mobile device with a browser able to automatically and dynamically retrieve and load Web pages, services, and applications according to the current context. The aim is the so called "physical browsing": browsing the digital world based on the situations in the real world. The Context-Aware Browser acquires information related to the user and the surrounding environment by means of sensors installed on the device or through external Web services. This information, combined with the user's personal history, and preferences, is exploited to infer the user's current context (and its likelihood). In the subsequent retrieval process, a query is automatically built and sent to an external search engine, in order to find the most suitable Web pages for the sensed context and present them to the user, or to invoke the most suitable Web services. This approach has to take into account several features; whence, we can say that the Context-Aware Browser is best described by the sum of the following parts: a web browser; a context-aware application for mobile devices to automatically retrieve and constantly update the contextual information gathered from the surrounding environment and remote services; an adapted search engine to search both for "traditional" Web pages, applications, and services on the

Web, and for specifically tailored applications; an application able to automatically load contents and manage remote services contextually invoked.

In the last years, attention has been given to the exploitation of service-oriented architectures to overcome the limitations in context-aware technologies. Service-oriented computing promotes the idea of assembling application components into a network of services that can be loosely coupled to create flexible, dynamic business processes and agile applications that span organizations and computing platforms [17]; thus they are particularly suitable for mobile devices. In addition, the orchestration of Web services can support and simplify the exchange of context information in large scale environments, thus enabling Web services systems to utilize various types of context information to adapt their behaviors and operations to dynamic changes [24]. Two examples of context frameworks that facilitate the development and deployment of context-aware adaptable Web services are Service-Globe [9] and Contextserv [20].

The last piece of the puzzle is Web 2.0 [16]. Web 2.0 and social software represent all web-based services with "an architecture of participation", featuring a high interaction level among users and allowing users to generate, share, and take care of the content. In the plenty of tools provided by Web 2.0, social bookmarking, folksonomies, and social filtering are of particular interest here. Social bookmarking is a method for organizing, searching, and managing documents of interest among users. In a social bookmarking system, users save links to documents of interest in order to remember or share them with the community. Social bookmarking is strictly related to the practice of annotating and categorizing content in a collaborative way, by means of informal tags. With the diffusion of Web 2.0 services, social tagging has gained importance, thanks to the easy and informal approach that allows also non-expert users to classify and find information. Although most people use tagging to organize their own content collection, even resources tagged for personal use can benefit other users. For example, if many users find something `funny`, there is a reasonable likelihood that someone else would also find it to be so [5].

The set of these tags forms a so-called folksonomy. A folksonomy (a portmanteau of folk and taxonomy), allows users to easily and informally describe documents and content, and represents a powerful combination that has gained popularity as it allows a management of knowledge more natural and simpler than traditional hierarchical system: the use of freely choosen categorizations and the collaborative aspect allow also non-expert users to classify and find information. Folksonomies are criticized because the lack of terminological control could lead to unreliable and inconsistent results [5]. Since the tags are freely chosen rather than taken from a given vocabulary, the following problems could arise: synonymy (multiple tags for the same concept), homonymy (same tag used with different meaning), polysemy (same tag with multiple related meanings), and heterogeneity in interpretations and definitions of terms. Despite these limitations, tags and folksonomies are widely used: so called tag clouds are appearing at fast pace in many website and tagged bookmarks are being shared by communities.

In the plenty of Web 2.0 systems, some researchers have already begun to explore the convergence of Web 2.0, IR, and context-aware computing. For example, [22, 23] propose a new just-in time information retrieval method using context information in a Web 2.0 environment, and [7] describe a context-aware personalized tagging system.

3 Social model for CAR

This section analyzes the limits of the current approaches to context management and suggests how to overtake these limits through a social approach.

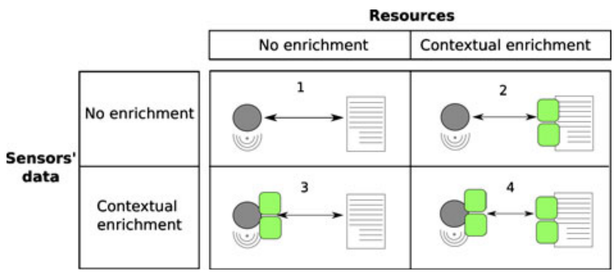
3.1 Motivations for a social approach

The difference between the Context-Aware Browser and the SCAB proposed in this paper is similar to the difference between content-based and social filtering: to work mainly on the content, or to exploit the community behaviour to determine which resources are relevant in a certain situation.

The SCAB workflow is similar to the Contex-Aware Browser one: (i) it acquires information related to the user and the surrounding environment, by means of sensors installed on the device; (ii) it enriches the representation of context exploiting context-aware models; and (iii) it retrieves the resources, or it invokes the services, most relevant for the given context. In particular, by means of information representing contexts, sensors data are matched with the resources. This matching can be realized in four ways: (1) directly matching sensors data and resources, (2) enriching the resources representation with contextual information, (3) enriching sensors data representation with contextual information, or (4) enriching, at the same time, both sensors and resources. More in detail (see also Figure 1):

1. Resources are retrieved directly on the basis of their content and of the information provided by sensors. For example, inside a museum, the SCAB perceives a wireless network named *Museum X* and retrieves the home page of Museum X.
2. As a first improvement, resources can be enriched with contextual information, in order to enhance their retrieval in the right situation. For example, a Web page describing a historical fact can be enriched with location information (GPS coordinates), to make simpler its automatical retrieval when users are in the related place (with a GPS enabled device).
3. In the same way, sensors data can be enriched with contextual information, in order to abstract on the raw data and to enrich the context representation. For example, information about user activities can be related to a particular set of sensors data: when the user is taking the dog out for a walk, the combination of certain sensors data (GPS position, speed, trajectory, temperature, bluetooth proximity, etc.) can be enhanced with high level contextual information describing that activity, for example the fact that the user is having fun training her dog.

Figure 1 Classification of the contextual enrichment cases of both sensors and resources.



Then resources can be retrieved on the basis of the activity of the user: when she is out with the dog, Web pages about dogs training are retrieved.
4. In the last and more general scenario, both resources and sensors data are enriched with contextual information. As an example, we can imagine a user in a museum: when she is near an artwork, a detailed description is presented by her device. In a crowded situation, on the contrary, a detailed description is not useful as users can have difficulties in seeing the paintings, thus a high resolution picture can be more interesting. In this example, the two resources (the description and the picture) are related to different contextual information (to be near the artwork and to be far from it because of the crowded place) and, at the same time, sensors data are enhanced with high level contextual information in order to discriminate a normal and a crowded situation.

Therefore, contextual information is continuously associated (added, removed, and modified) to both resources and low level sensors data. This requires the creation and management of a large amount of information related to contexts, to model all the possible contexts of interest. Thus, the problem is to understand who is the provider of this contextual information and how it has to be defined.

In current approaches to context-awareness, the information about contexts is usually provided by a small group of experts (application developers or specific domain experts). This is due to the difficulties in representing contexts. In fact, a precise representation of the user's context requires a full definition of all the information that composes that context, and the only way to support the required precision is exploiting techniques like categorizations, taxonomies, and ontologies. Moreover, existing approaches that manage several dimensions of context (e.g., location, time, user's activities, needs, resources in the nearbies, light, noise, movement, etc.) require a large amount of information to represent all these contextual dimensions. These are the reasons why current approaches show a trade off between the generality of applications and the depth of context representations: applications that fully manage several contextual dimensions are confined to limited fields (e.g., Smart Homes), while general applications work only on a narrow notion of context (e.g., in location-based applications the context is represented just by location-time). In addition, contexts are defined a priori, and there is no way to dynamically extend the contextual values adopted or to enhance their representation at run-time: the operations of modeling contexts and using context-aware applications are rigidly separated.

Current approaches are not suitable for the SCAB, because we aim at a general context representation, where several contextual dimensions are exploited, and where the information dynamically changes to adapt to the user's current situation. Thus a dynamic nature and a large amount of information to be categorized and modeled (to represent both contexts and contexts-resources associations) are required.

Starting from these considerations, we propose a novel model for context-awareness, to support the SCAB, aiming at overtaking the just defined limits by exploiting the social dynamics underlying the Web 2.0. The underlying working assumption is that the collaborative effort of a community can allow a comprehensive definition, management, and use of context. In particular, the SCAB avoids a priori contexts definitions made by experts, and allows people not to be just passive users. Rather, users can freely interact with resources and contextual information: through collaborative annotations, they can explicitly define the context their are in, they can define that a resource is relevant (or not adapt) to their current context, they can associate resources to particular contexts, and finally they can browse resources relevant for their current context. The aims are to have a dynamic and user-tailored context representation, and to enhance the process of retrieval based on users' actual situation.

3.2 Conceptual model

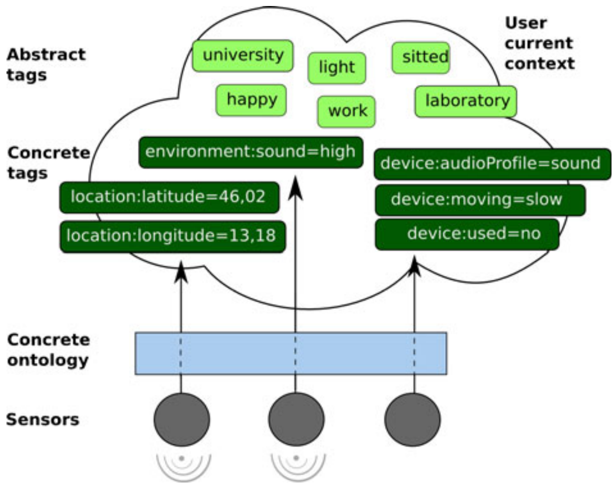
The SCAB conceptual model includes a tag based context representation and a set of operations to manage contextual information and resources.

3.2.1 Context and resources representation

We represent the user context as a tag cloud, where each tag (keyword or string of text) represents a single contextual information. This representation allows an easy and informal modelling of context, giving the opportunity also to non-expert users to classify and find context-related information. This tag cloud is composed of two kinds of contextual information: the low level information coming from sensors and the high level information introduced by users; thus we distinguish two categories of tags: concrete and abstract tags. The user context tag cloud is composed by a set of these tags; for example the cloud expressed in Figure 2 can refer to a working afternoon. Also, the sensors on the mobile device return tags: this allows a consistent representation and users can directly manage the low level contextual information.

In particular, concrete tags represent the information obtained by a set of sensors. Sensors can be either physical ones, like a thermometer or a wireless antenna, that gather data from the surrounding physical environment, or logical, like a calendar, that obtain data via software. Physical and logical sensors can be either on the mobile device, or on an environmental server that communicates with the device. Abstract tags, on the contrary, represent the high level contextual information that are freely associated by the users to the concrete contexts, in order to detail their context description. Some examples are: `home`, `shopping`, `dog`, `walk`, etc.

Figure 2 User's current context.



While the high level contextual information (abstract tags) is freely managed by the users, without conditions, concrete tags are automatically generated starting from sensors values. In addition, concrete tags should be formally defined using a common structure. Considering for example location, a distance expressed in meters is different from a distance expressed in miles, thus it is important to exactly categorize low level information. To facilitate interoperability and high-level context inference, to avoid mismatch between contextual information, and, in general, to guarantee a systematic management of low level contextual information, we designed a simple ontology to represent the sensor-based context information by means of tags. This approach is feasible because the concrete tags are in a limited number when compared to all their combinations, and to the abstract tags. We rely on already existing solutions, by exploiting the ideas proposed in [10]: our ontology consists of a vocabulary, that presents the terms for describing context information and a schema which represents the structure and the properties for all the ontology concepts.

As done for the Context-Aware Browser (see [4] and the papers cited therein for more details) we rely on a rule-based system, implemented in Jess (http://www.jessrules.com/), to map the raw sensor data onto the concrete context vocabulary. In particular, this is a twofold process: a first set of rules converts the continuous sensor data into discrete form (for example, the light intensity is quantized into four levels: "null, low, medium, high"), and a second set of rules maps the contextual features into the ontology representing sensor-based context information (for example, the light intensity can be represented with the tags `Environment:Light:Measure:Lux:22`, `Environment:Light:Level:Medium`).

Each resource is represented by its URL and all the tags associated to it.

3.2.2 Entities and operations

Users, contexts, and resources are the main elements of our model (Figure 3), that refines a previous proposal [14]. A user, in the real world, is engaged in some activities, she has some needs, and she perceives her surrounding environment through her senses and the sensors on her mobile device. Contexts are representations of the users current situations. Resources are any kind of content that could satisfy the user's needs. In the SCAB, the tags (small squares in Figure 3) are socially defined

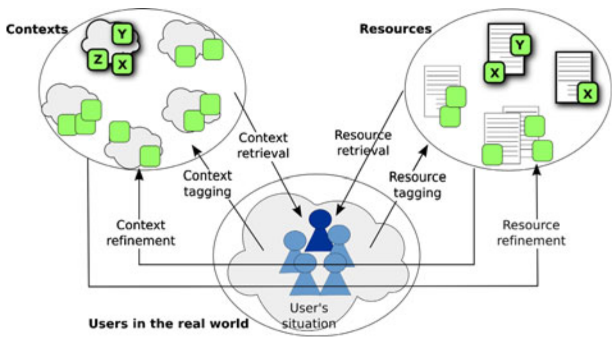


Figure 3 A conceptual model for social CAR.

by the users themselves through tagging operations. Six are the main operations in the SCAB model:

Context-tagging: users can explicitly use tags to represent contextual information, i.e., they can tag the contexts they are in. For example, the user can enhance the values provided by sensors (concrete tags) on the mobile device by manually adding her own tags, as `out, dog, walk, leash, park, play, ball`. Doing so, these abstract tags are directly linked to the concrete ones and when other users have the same (or similar) set of sensors values, these tags become part of their context representation.

Resource-tagging: users can explicitly annotate a resource with contextual information to allow her, and the community of users, to automatically retrieve it when they are in the suggested context. For example a user can associate a web-radio with the abstract tags `out, dog, walk`, to represent the fact that she usually listens to the web-radio when walking with her dog.

Context-retrieval: starting from the concrete tags provided by sensors, the abstract tags defined by the community that best describe the user's actual situation are retrieved by the SCAB. For example, starting from the tags representing GPS coordinates, the current user's context could be automatically enhanced with the tags `walk, sunny, park`. This happens as some user has previously associated these contextual tags with the same or similar GPS coordinate.

Resource-retrieval: on the basis of both users' current context and the contextual information associated to resources, the most relevant resources are retrieved by the SCAB. For example, when the user is out with her dog, first the SCAB enhances her context representation through a *context-retrieval* operation (e.g., retrieving the tags `out, dog, walk`); then, the resources relevant to this context, like the web-radio previously tagged, are retrieved.

Context-refinement: information about contexts is automatically refined by the SCAB on the basis of the interaction between users and resources. For example, if a user is using resources annotated with the contextual tag `work`, probably she is working, and the representation of her context can be enhanced with this information.

Resource-refinement: information related to resources is automatically refined by the system on the basis of the interaction of users within their contexts. For example, if a lot of users work with a Web application in the same context `work`, probably this resource is related to that context and it is automatically annotated with it.

By "tagging" we mean both a positive tagging, where users add tags, and a negative tagging, where users remove tags from a collection. For example, a user

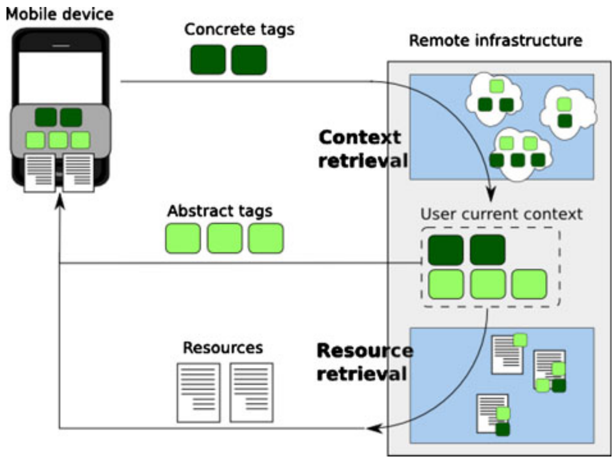


Figure 4 Context and resource retrieval operations.

can observe her context tag cloud and remove the abstract tags that do not suit her situation or she can remove contextual tags associated to resources.

Although the knowledge related to the whole community is exploited to infer and refine the current context of single users, it is important to differentiate the personal from the community level, giving more importance to the first one. For example, if a user annotates a situation as `play`, she is considered to be in `play` context, even if most people annotate the same situation as `work`. On the contrary, if a user is in a situation for the first time (e.g., a never visited location), her context is refined just with the information from the community. Considering the previous example, as most people annotate the situation with `work`, the user is considered to be in `work` context.

The overall SCAB system is composed by two main elements (Figure 4): a remote infrastructure aimed at storing all the information representing contexts, the resources, and their associations, and an application on the users' mobile device, that allows users to interact with contextual information and resources. While the *context-tagging* and *resource-tagging* operations are performed by users, the *context-retrieval* and *resource-retrieval* operation are performed automatically by the application on the mobile device: in particular it continuously receives sensors data, enhances the context representation by retrieving abstract tags, and, on the basis of the context tag cloud, it retrieves the most relevant resources. Finally the *context-refinement* and *resource-refinement* operations are automatically performed when users interact with resources. Considering the temporal aspect, a tag remains in the user's context representation until sensors data change so much that the considered tag is no more relevant (similar for resources). For example, considering location, the tag with the name of the town is the user's context until she is in that town.

4 Detailed model

In this section we show how the six operations in the model can be implemented in a system, from a detailed and quantitative point of view. To avoid problems related to the quality of context and resources, we rely on a social evaluation/reputation mechanism: each element in the model (users, contexts, resources) has a score that increases or decreases according to the community behaviour. The score of each user is used to weigh the operations she performs and it will increase/decrease based on the goodness of the tags inserted by the user associated both to contexts and resource. The scores of contexts and resource tags define their quality and relevance. If a resource tagged with a given tag is never used in a context tagged with the same tag, the related score decreases and more relevant resources will stand out.

Although we provide some justifications, the low level details of our proposal are often arbitrary and alternatives are possible (and we hint at some of them in the following): our aim is not to concentrate on the details, but to provide a general view on the whole system logic and to show that at least one approach is feasible. In Section 5 we will discuss the effectiveness of our specific solution.

4.1 Indexes

In our model concrete tags are exploited to retrieve the most relevant abstract tags, and all the tags are exploited to retrieve the most relevant resources. The connections between concrete tags, abstract tags, and resources are built on the basis of the community social behaviour. To store the links from concrete to abstract tags and the links from all the tags to the resources, we define two indexes (we adopt this terminology since the whole process is mainly a retrieval process): in the first one, called *contexts index*, concrete tags index abstract tags, while in the second one, called *resources index*, the set of all tags (both concrete and abstract) indexes the resources (Figure 5). Both the indexes are managed by remote servers and not stored on the mobile device. Since the approach is similar for both indexes, we are going to show just the first one.

The context index is a matrix, where each column corresponds to a concrete tag, and each row corresponds to an abstract tag. The matrix is not fixed, but the number of columns/rows can evolve. Each cell contains three values:

– U_{ij}: user that created the association from concrete tag c_j to abstract tag a_i;
– S_{ij}: a score in $[0, 1]$ and starting value ϵ that defines how much relevant the abstract tag a_i is for the concrete tag c_j;
– σ_{ij}: a steadiness value, greater than 0 and starting value ϵ, that defines how steady is the association between the abstract tag a_i and the concrete tag c_j.

The process to compute score and steadiness values is described in Section 4.2.

Since not all the abstract tags can be related to all concrete tags, the index is a very sparse matrix, and because of the very high number of concrete and abstract tags, the

	c_1	c_2	...
a_1			
a_2		$(U_{22}, S_{22}, \sigma_{22})$	
$\vdots$			

	t_1	t_2	...
r_1			
r_2		$(U_{22}, S_{22}, \sigma_{22})$	
$\vdots$			

Figure 5 Context and resource indexes.

index can be quite large. State of the art compression techniques, also in the CAR field [15], can be adopted: their discussion is out of the scope this work.

4.2 Context and resource scores and steadiness

The indexes in Figure 5 are not static: the values related to the association between concrete and abstract tags and resources are continuously updated, on the basis of the interaction of users with resources in context.

After every *context-tagging* operation between an abstract tag a_i and a concrete tag c_j, the values in the contexts index are updated according to the following formulae (for the resource index and the *resource-tagging* operation a similar approach is used):

$$\sigma_{ij}(t_{k+1}) = \sigma_{ij}(t_k) + S_{U_c}(t_k) \times \beta, \tag{1}$$

$$S_{ij}(t_{k+1}) = \max\left(\frac{\sigma_{ij}(t_k) \times S_{ij}(t_k) \pm S_{U_c}(t_k) \times \beta}{\sigma_{ij}(t_{k+1})}, 0\right), \tag{2}$$

where t_k represents a discrete time instant, t_{k+1} the subsequent time instant, and β is a parameter used to weight the user score. S_{U_c} is a score in [0, 1] measuring user's goodness in associating abstract tags to concrete tags; it is defined in Section 4.3.

The steadiness values (Formula (1)) increase over time and their increment depends on the score of the user performing the *context-tagging* operation: good users will make an association to steady faster than bad users, as good users should have more effect on the system than bad ones. The new score value of an association (Formula (2)) is computed by starting from the old score value of the association weighted by its old steadiness, and by summing (or subtracting) the user score. The resulting value is divided by the steadiness at the subsequent time instant, in order to have a score value in [0, 1]. The higher the steadiness of an association is, the more stable the association is, and then the smaller effect each subsequent update operation will have. In the same way, since the score is weighted by the steadiness, the score of a stable association has more weight in the computations (Formula (2)) than the score of a low steadiness association. The user's score is added or subtracted on the basis of the operation she performs, that can be adding or removing an association.

We can consider, for example, a user tagging a resource with her context (*resource-tagging* operation). Although we have discussed only the context index and the *context-tagging* operation, the considerations are the same for the resources and therefore we provide an example considering both contexts and resources. In particular the user's mobile device senses the concrete tag `location:latitude=46.087654` and her context is enriched with the abstract tag `Udine` (retrieved by the system by means of a *context-inference* operation). If she just tags the resource with her context tags, considering just the context index, the abstract and concrete tags are associated, and in the score computation the user's score is added ($+S_{U_c}(t_k) \times \beta$). On the contrary we can now imagine that the user removes the tag `Udine` from her context, just before the operation of *resource-tagging*. In this way, she implicitly defines the weakness of the association between these tags, thus in the score computation the user's score is subtracted ($-S_{U_c}(t_k) \times \beta$). Considering the resource index, on the contrary, in the first case both the concrete and the abstract tag are associated to the resource; in the second case, on the contrary, just the concrete one.

The higher the user's score, the more effective the update operation: good users have more influence on the system than bad users. Finally, through β, the operations performed explicitly by users (*context-tagging* and *resource-tagging*) have more effect than implicit updates performed automatically based on the interaction of the community with the resources (*context-refinement* and *resource-refinement*). In our experiments (Section 5), since we concentrate only on explicit operations, we use $\beta = 1$.

As we have seen, the values in the indexes change dynamically on the basis of community interaction. However, this is not the only possible solution, as complementary approaches are possible: an example could be to use some geographic gazetteer for associating geonames to geographic coordinates provided by the concrete tags, so as to reinforce the rank of associated abstract tags that contain the same geographic names or names of close localities; the geonames could be useful also for retrieving more relevant resources, those containing the geonames or close geonames; ConceptNet can be exploited too; etc.

4.3 Users' scores

Two scores (in [0, 1]) are associated to each user and they define the goodness of the user in working with contextual information. S_{U_c} defines how good the user is in associating abstract tags to concrete tag, while S_{U_r} defines how good the user is in associating resources to tags. As previously, we concentrate only on the management of values related to concrete and abstract tags, since the approach is exactly the same for tags and resources.

Every time a new relation between an abstract and a concrete tag is created by means of a *context-tagging* operation ("filling a hole" in the context index), the user who performs the operation is associated to that connection (the U_{ij} user in Figure 5). Then, on the basis of the community behaviour, the score and steadiness values will be updated (Section 4.2) and this will also modify the user's score S_{U_c}. For example, if the user U is the first one to associate the abstract tag `run` to the concrete tag `device:movement=cyclic`, and other users in the community will later on do the same, the score and steadiness of that association will increase, and so the user's S_{U_c} score.

Each user's score is calculated as a mean over the scores of all the associations created by that user weighted by the steadiness of each association (the steadiness values are normalized by dividing them by the maximum steadiness value). After every *context-tagging* operation, the user's score is updated; in particular, considering an operation on the association between abstract tag a_i and concrete tag c_j made at time t_k, the score is updated according to the following formulae:

$$S_{U_c}(t_{k+1}) = \begin{cases} S_{U_c}(t_k) - x_{ij}(t_k) + x_{ij}(t_{k+1}), \text{ if the association } ij \text{ already exists,} \\ \dfrac{S_{U_c}(t_k) \times n(t_k) + x_{ij}(t_{k+1})}{n(t_k) + 1}, \text{ if the association } ij \text{ has just been created,} \end{cases} \quad (3)$$

where $x_{ij} = \frac{\sigma_{ij}}{\sigma_{\max}} \times S_{ij}$, t_k represents a discrete time instant, t_{i+k} the subsequent time instant, $\sigma_{\max}$ is the maximum steadiness value in the context index (the same for all users), and n is the number of associations between concrete and abstract tags made by the user. S_{ij} and σ_{ij} have been presented in Section 4.2.

New associations have a low steadiness σ_{ij} value, thus their score, as they are not steady yet, will have low influence on the user's score. Good associations will have high score S_{ij} and steadiness σ_{ij} values, and they will reflect on high users' score. In the same way, low users' scores are due to bad associations between tags. Since $S_{ij} \in [0, 1]$, also $S_{U_c} \in [0, 1]$.

In this approach, for simplicity, only the *context-tagging* operations that create new associations are considered for the computation of the users' score. An extension could be to consider, with a different weight, also *context-tagging* operations that confirm (or remove) existing associations: if a user associates an abstract tag to a concrete one, even if this association already exists, it would be used in the user's score computation. In this way a user would be "good" also because she confirms existing good associations.

4.4 Context and resource retrieval

Having defined how the score and steadiness values are computed and updated, we now show how to use them during the retrieval operations (Figure 4): with the *context-retrieval* operation, starting from the concrete tags sensed by sensors, the most relevant abstract tags are retrieved; and with the *resource-retrieval* operation, starting from the set of all the tags in the user's context, the most relevant resources are retrieved. These operations work on the context and resource indexes, respectively. Since they are similar, we discuss just the context-retrieval one, that is performed as follows.

1. Starting from the concrete tags in input, only the set of abstract tags that have been associated with at least one of the concrete tags are considered.
2. For each abstract tag a *rank* value is computed, to define an order of relevance for the abstract tags.
3. In order to limit the number of retrieved tags, only the abstract tags whose *rank* value is higher than the median of all *rank* values are retrieved. This is just a way to consider the most relevant ones; other solutions can work on fixed thresholds, as the first k tags, etc.

The *rank* value, for each considered abstract tag a_i, is computed following a modified version of the tf.idf weighting scheme: for all the sensed concrete tags c_j, the scores S_{ij} of the association between the concrete and abstract tag, weighted by the corresponding steadiness values σ_{ij}, are summed up; the result is multiplied by the ratio between the number of sensed concrete tags and the total number of concrete tags to which the abstract tag is related. In formulae:

$$A_i = \sum_{c_j} \sigma_{ij} \times S_{ij}, \text{ for each sensed concrete tag } c_j,$$

$$\text{rank} = A_i \times \frac{B_s}{B_a}, \tag{4}$$

	c_1	c_2	c_3	$\dots$	
a_1	$(U_{11}, S_{11}, \sigma_{11})$	$(U_{12}, S_{12}, \sigma_{12})$			$\rightarrow A_1 = \sigma_{11} S_{11} + \sigma_{12} S_{12}$
a_2					
a_3			$(U_{33}, S_{33}, \sigma_{33})$		$\rightarrow A_3 = \sigma_{33} S_{33}$
$\vdots$					

Figure 6 Computation of the first part of the rank value.

where B_s is the total number of sensed concrete tags, to which the abstract a_i tag is related, and B_a is total number of concrete tags to which the abstract tag a_i has been associated. The A_i value computation is explained in Figure 6.

Some considerations can be drawn. First, more concrete tags in the current context to which an abstract tag is associated, mean that its rank value will be higher, as the sum contains more items. Second, abstract tags with high score and steadiness will have a higher rank value. Third, abstract tags related to particular sets of concrete tags will have a higher rank value than very general ones that are associated to a high number of concrete tags (high frequency). This works exactly as the tf.idf measure: given a set of concrete tags, the importance increases proportionally to the number of times an abstract tag appears in the associations (tf), but is offset by the frequency of that tag in the whole associations corpus (idf).

Starting from this basic approach, the rank value computation can be enhanced exploiting more information. For example, an idea would be to weigh the tags based on their age in the user's context representation, giving more importance to the newest tag, and enhancing the importance of new contexts. This is left as future work.

4.5 Example simulation

After having presented the details behind our social evaluation mechanism, we provide a simple simulation, in order to show how the formulæ work and how the scores change based on the user interaction. To simplify the simulation we can imagine that the users' mobile devices can consider only location and time information. Also, to concentrate on how the system works, we simplify the representation of concrete tags, not showing real location and time data and abstracting from time and location granularity details.

The following are six *context-tagging* operations performed by two users:

1. User U1 associates the abstract tags `morning, awaken, home, breakfast` to his concrete context `loc1, t1`.
2. U2 associates `morning, work, office, computer` to `loc2, t1`.
3. U1 associates `home, noon, cooking, eating` to `loc1, t2`.
4. U2 associates `lunch, office, eating, computer` to `loc2, t2`.
5. U1 associates `home, evening, dinner, TV` to `loc1, t3`.
6. U2 associates `gym, sport, evening` to `loc3, t3`.

Table 1 represents the part of contexts index concerning these tags; each cell shows the steadiness-score product ($\sigma_{ij} \times S_{ij}$) for an abstract tag with respect to

Table 1 Example of associations from concrete to abstract tags in the contexts index.

Abstract tags	Concrete tags					
	Location1	location2	Location3	Time1	Time2	Time3
Morning	**.49**	**.49**		**.99**		
Awaken	.49			.49		
Home	**2.05**			.49	.99	.55
Breakfast	.49			.49		
Work		.49		.49		
Office		.99		.49	.49	
Computer		.99		.49	.49	
Noon	.99				.99	
Cooking	.99				.99	
Eating	.99	.49			**1.49**	
Lunch		.49			.49	
Evening	.55		.38	.38		.55
Dinner	.55					.55
TV	.55					**.55**
Gym			.38			**.38**
Sport			.38			.38

a concrete one, obtained after the six *context-tagging* operations, by means of Formulae (1) and (2). In this simulation each user starts with ϵ score and all tags start with ϵ score and steadiness. Being a simple simulation, the values shown are not very significant as they are not steady yet and so they are susceptible of sudden changes. They are anyhow useful to understand the dynamics behind our social mechanism.

We can draw some considerations; values of interest are highlighted in bold in the table. The tag `morning` has been associated to concrete tags `location1`, `location2`, `time1`, but it has a higher value related to `time1` since it has been associated with it several times. In the same way the abstract tag `home` is very relevant when the user is in the location `location1` and the tag `eating` is relevant for the time `time2`.

At a first sight, this process is similar to considering just the number of tags assigned, but it is not so: we consider also the user goodness in managing contextual information. In this simple case the users score are 0.38 for U1 and 0.25 for U2. Thus the operations performed by U2 have a lower weight than the operations performed by U1, since she is not as good as U1. To show this we can consider an abstract tag that has been used just one time: the tag `TV` has been associated to concrete tag `time3` by U1 and has score 0.55; in the same way U2 associated `gym` to `time3` and it has the lower score 0.38.

The association between resources and tags in the resources index is similar.

5 Experimental evaluation

In this section we present the experimental evaluation of the SCAB social approach to contexts and resources management and retrieval.

5.1 Aims and methodology

In general we were interested in understanding how the overall approach proposed in Section 4 is effective; more in detail we were interested in:

- Retrieval effectiveness:
  - Q1.1. Which is the effectiveness of *context-retrieval* and *resource-retrieval* operations?
  - Q1.2. How does the resource retrieval effectiveness depend on the number of tags describing the user's context and on the number of resources to be retrieved?
- User behaviour:
  - Q2.1. Is the system robust to the behavior of bad users? Is it too dependent from the presence of good users?
  - Q2.2. Has the order of the users an effect on the social evaluation mechanism?
  - Q2.3. Has the possibility of removing tags and resources an effect on the system effectiveness?
  - Q2.4. Which is the user's tagging behavior for contexts and resources?

Although the user evaluation approach is usually considered the most appropriate for CAR applications, it has also some, not negligible, drawbacks, when compared with benchmarks. First, it has the main aim to study how the system is accepted by users and satisfies them, whereas benchmark evaluations can evaluate different aspect of the retrieval process and alternative implementations. Second, a mature prototype to test has to be available, complete of an effective user interface. This goes against the purposes of developers, and it forces significant implementation decisions uncovered by evaluations. Third, user evaluation is more complex to perform, more time demanding, and it is more dependent on users' subjectivity than benchmarks.

Our aim is not to evaluate the whole SCAB application, but to study the effectiveness of the proposed social approach to context and resources management and retrieval, and to understand the goodness of the proposed algorithms. For this, a benchmark evaluation seems more suitable. Also, in the IR community, the benchmark-like evalution has a strong tradition, that predates the well known TREC (Text REtrieval Conference, http://trec.nist.gov/) and dates back to the 60s with the Cranfield experiments, if not earlier, and the problem tackled by the SCAB can be framed as an IR problem. Finally, as discussed in [18], although benchmark-like evaluations in the IR field usually rely on several participants to exploit pooling, it is also possible to adopt this approach for single research groups, and thus without pooling. Therefore, as done in previous work [12, 13] we took inspiration from TREC and we adopted a TREC-like benchmark evaluation named SREC (Social Retrieval Evaluation Collection).

We are well aware that the choice between benchmark and user study is a difficult one. Indeed, as we have already mentioned in [13, 14], the evaluation of a novel and highly interactive context-aware retrieval application like the SCAB puts researchers in a paradoxical situation: on the one hand, users and tasks seem needed, thus a user study based on some task analysis seem mandatory. On the other hand: (i) user studies would be far too expensive at an early development stage; (ii) there are no realistic users and tasks until the system is launched on a large scale; (iii) if a "wrong" system is launched to start the wave and have some users, then the system will fail, and there will be no users at all; and (iv) to fully evaluate the approach, some contextual data should be available on the Web, which is not yet the case.

Pushing this line of reasoning to its extreme consequences, one might wonder if evaluation somehow hinders development in the IR field. Indeed, when a paradigm shift is going to happen, it is quite unlikely that current evaluation methodologies can cope with the new scenario: they will evaluate the revolutionary system on the basis of the current evaluation techniques, that could be not appropriate for such a system and could lead to negative results. To give some examples: were SMS (the short text messages) evaluated? Was the Web evaluated? The obvious answer to these questions is no: these services have been used in creative and unforeseen ways by the users—far beyond the limits of evaluation methodologies well established at that time. This is an extremist's position, to be taken with caution, since evaluation is fundamental for new systems; however, it explains the difficulty of evaluation.

5.2 The SREC benchmark

SREC is constituted by the usual three components: the statements of information needs, the collections of elements to retrieve, and a set of relevance judgments. The statements of information needs (TREC topics) are context descriptors, representing different examples of user's situations in different domains. Each descriptor is composed by a title, a description field describing the situation, and a concrete tag field containing data from sensors that could be realistically associated to that particular situation (Figure 7). SREC includes five context descriptors, which differ for user activities, location, time, etc.: student at university studying for an examination, sunny Saturday afternoon at the park, touristic holiday in London, shopping in London, and wine and food trip to London. These descriptors also differ for specificity: the first one presents a very specific situation, with precisely defined

```
<contextDescriptor>
<title> Context descriptor 2: afternoon in the park </title>
<description>
It is summertime and it is a sunny Saturday afternoon. You have taken
advantage of a beautiful day to go out with your friends and do some sport
at the town park. You reached the park with your bycicles, supplied with the
necessary for spending a good afternoon: sports wear, picnic basket with food
and beverage, balls, ipod. The park is crowded: there are children playing,
different ages people running, and young people engaged in sport activities.
</description>
<concreteTags>
location:latitude=46,08442 - location:longitude=13,18919
time:date=10-07-2010 - time:hour=15-00
user:moving=yes - user:speed=slow
environment:temperature=29°celsius - environment:sound=moderate -
environment:light=bright
device:used=no - device:orientation=vertical - device:audioProfile=sound -
device:moving=slow
resource:bluetooth:112233445566 - resource:bluetooth:445566112233
</concreteTags>
</contextDescriptor>
```

Figure 7 A (part of a) context descriptor.

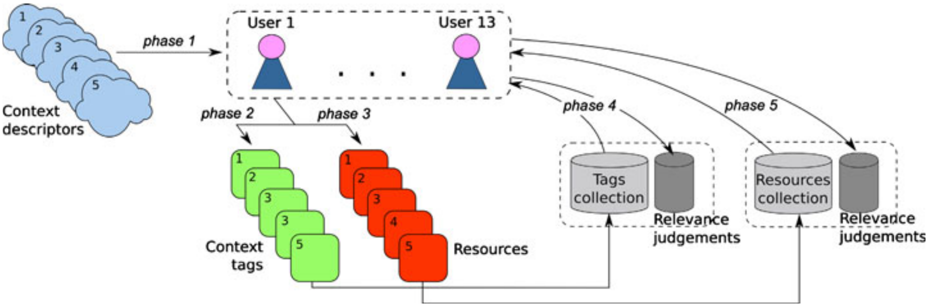


Figure 8 Construction of the collections.

location and activity; the other ones, on the contrary, are more general, without details on activities, etc.

Since we are working with both tags and resources, our benchmark contains two different collections. They have been built socially by a community of 13 participants (computer science students), with the process shown in Figure 8. By means of an interactive test, we asked the participants, for each context descriptor, to empathize with the described situation (phase 1), to define the tags that best describe the situation (phase 2), and the most suitable resources (Web pages, applications) for that situation (phase 3). All the tags and resources defined by the participants represent our two collections: they contain 460 unique tags and 150 unique resources. The tags have been found by thinking about the situation and describing it. The resources have been found by the participants considering the applications they use on their mobile devices, and surfing online by means of traditional search engines. The relevance judgements, for both tags and resources, have been made socially by the same participants in a different session of the test, that was held two weeks later. Each participant expressed her opinion on all tags (phase 4) and resources (phase 5) in the collections using a five level relevance scale, from 1 (not relevant) to 5 (extremely relevant). These judgements have been used to compute, for each tag and resource in the collections, its relevance value, defined as the mean between all the relevance judgments expressed by the participants on that element. This value, named *"True" relevance* is exploited afterwards in the experiments. This session has been temporally separated from the first one in order to have the participants to forget the information they previously associated and thus have the most truthful judgements. Indeed, it happened that some participants associated to a context descriptor some resources that they judged not relevant for that situation two weeks later. The collections in our benchmark can evolve and increase by adding more participants to the test.

5.3 Experiments

We used SREC to evaluate the effectiveness of the proposed approach, and in particular of the algorithms described in Section 4. We performed six different experiments, grouped into three pairs. All the six experiments were carried out in the same way (Figure 9). Given a context descriptor, each participant performed

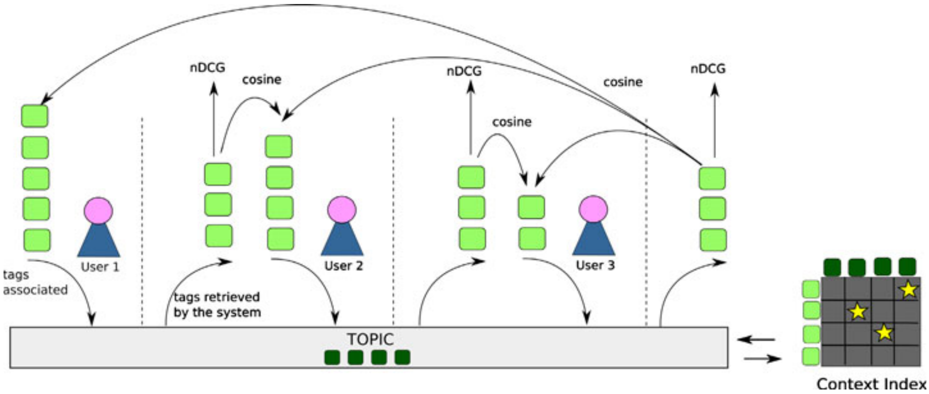


Figure 9 Example of analysis during the experiments (3 users and considering just tags).

a *context-tagging* and a *resource-tagging* operation. After each participant, *context-retrieval* and *resource-retrieval* operations were executed, and nDCG (Normalized Discounted Cumulative Gain, a standard IR metric that emphasizes quality at the top of the ranked list) values for the retrieved tags and the resources were computed, exploiting the "True" relevance value described in Section 5.2. For each participant, we also compared the vectors of tags and resources she defined with the vectors of tags and resources retrieved by the system, computing the cosine similarity between them.

Finally, all the experiments have been performed changing the number of the tags and resources retrieved by the system: we repeated the experiment for 5, 10, 15, 20, 25, 30, 35, 40 tags and for 5–15 resources. We did not consider more than 40 tags since in a mobile environment we want to avoid an information overload. In the same way we considered at most 15 resources since in a mobile environment users are unlikely to scroll long lists of retrieved resources.

In the first experiment we started from an empty system, without any information on tags and resources. For each context descriptor, following the order of participation to the test, we simulated a series of *context-tagging* operations, associating the participants' proposed abstract tags (during phase 2) with the concrete tags associated to the descriptors. Then we simulated a series of *resource-tagging* operations, associating the participants' proposed resources (during phase 3) with both the concrete tags in the descriptor and the abstract ones just defined. The aim of this experiment was to analyze a cold start of the system, observing how the retrieval effectiveness for tags and resources changes with the participants participation.

The second experiment was very similar to the first one, except for the information exploited: instead of using all tags and resources defined by the participants during phases 2 and 3, we used only those tags and resources judged extremely relevant by the participants during phases 4 and 5. In particular, for each participant, we considered the set of all tags and resources inserted before her, and we associated only those ones judged as extremely relevant by that participant. For example, given participant 3, we considered only tags and resources associated by participants 1 and 2, and among them we associated only those that participant 3 defined extremely relevant. In this way we simulated the situation in which the users are aware of the information already inserted and can select which tags and resources to associate (as a recommender system).

The third and the fourth experiments were similar to the first two, but were executed on a non empty system. The state reached by the system during the first experiment is used as the initial state for the third and fourth experiments, simulating users that perform the same operations in another day. This is not a situation far from reality, since people are creatures of habit: as stated in [6], they make regular trips to the same few destinations, and often perform the same activities. While the first two experiments were aimed at studying a cold start, in this two we wanted to observe how effectiveness changes by increasing the amount of information available.

The fifth and sixth experiments are built on the previous two, and add the possibility of removing tags and resources considered not enough relevant. The removal of a tag decreases the strength of the association between that tag and the sensors values, while the removal of a resource decreases the strength of the association between that resource and the tags describing the user context (Sections 3.2.2 and 4.2). In particular, in the fifth experiment, we considered the participants' proposed abstract tags and resources and the tags/resources judged extremely relevant are added, while the other ones are removed. In the sixth experiment, for each participant, we considered the set of all tags and resources inserted before her, and we associated only those ones judged as extremely relevant by that participant, removing the other ones.

5.4 Results

5.4.1 General data

The collection of tags is composed by 460 unique tags (743 considering also the repetitions): the average is 92 tags per context descriptor, the minimum is 67 and the maximum 11. The collection of resources is composed by 150 unique resources (213 considering also the repetitions): the average is 30 resources per context descriptor, the minimum is 21 and the maximum 38.

Figure 10a shows the overall number of tags and resources for each context descriptor. While no general consideration can be made upon resources, we notice that the context descriptor related to a very specific situation, context descriptor 1, has fewer associated tags than more general context descriptors.

Considering the set of tags inserted by each participant on each context descriptor, the largest set has 27 tags, the smallest one has four tags, with an average of 11.5 tags per set. Considering the resources, the largest inserted set has ten resources, the smallest 1, with an average of 3.8 resources. The distribution of the dimensions of the set of tags and resources inserted is presented in Figure 10b. We also observed that only the 17% of the inserted tags occur in the context descriptor; all the other tags have been creatively generated by the users.

We analyzed the nDCG value of the sets of tags and resources inserted by the participants during the test, exploiting the "True" relevance value defined in Section 5.2. The average value for tags is 0.86, the minimum is 0.52, and the maximum 0.98; for the resources the average value is 0.84, the minimum 0.49, and the maximum 0.99. Finally, considering how users judged tags and resources, we noticed that average variance for tags is higher than resources: users tend to agree more on resources than on tags, as the last ones are probably more subjective.

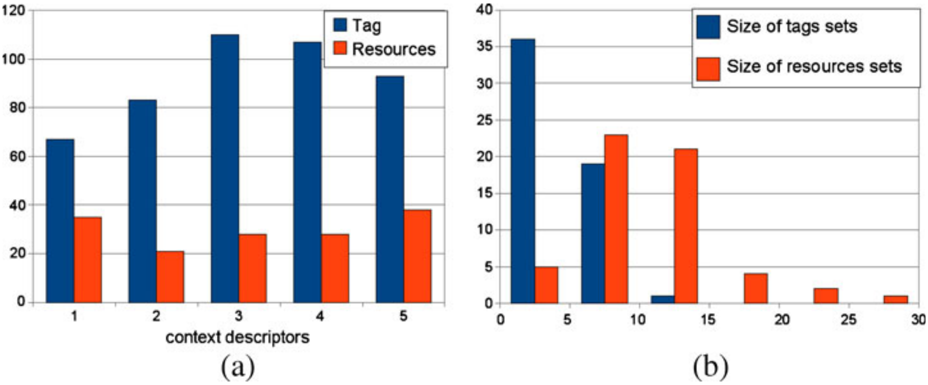


Figure 10 Distribution of tags and resources: **a** number of tags and resources for each context descriptor; **b** dimensions of set of tags and resources inserted.

These considerations answer our question Q2.4 (Section 5.1), related to the users' behavior on tagging contexts and resources.

5.4.2 nDCG of tags

The nDCG values for the tags retrieved by the system during the six experiments are shown in Figure 11. In particular the graphics shows the results for each experiments and the average results (bold line).

Figure 11a presents on the vertical axis the nDCG value and on the horizontal one the number of tags. We can notice that the average nDCG value slightly decreases with the increment of the number of tags, stabilizing at around 30 tags. This means that by increasing the number of retrieved tags the system retrieves fewer high relevant tags, or rank them lower. Figure 11b presents on the horizontal axis the participants in the order they performed the test (phases 1–3). The vertical dashed lines refer to the two participants (2 and 8) that inserted the less relevant tags: the nDCG value related to them is lower than the average nDCG value.

First of all we can notice a "big hole" just after the first of the two "worst" participants. Participant 2 associated to the context descriptor several not relevant tags that became part of the set of retrieved tags for the subsequent participants,

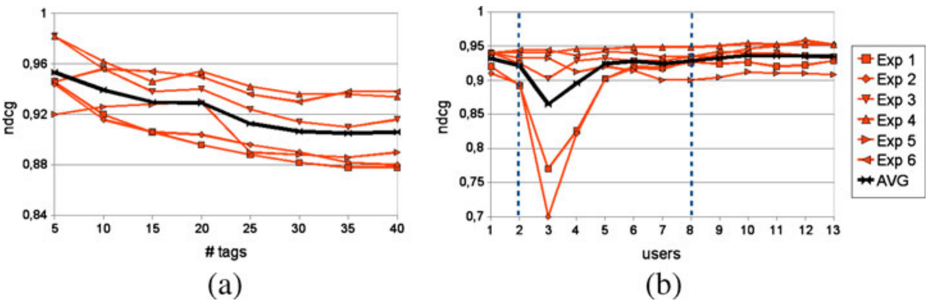


Figure 11 Mean nDCG for the tags retrieved, on all contexts.

in this way decreasing the overall performance. The information inserted by the subsequent participants weakened the associations between the situation (context descriptor) and these not relevant tags, giving more importance to the relevant ones. In this way, just after a couple of participants, the not relevant tags disappeared from the set of retrieved ones. Also the fact that participant 2 was one of the first ones compounded to the problem just presented. In fact the first participants represent a bootstrap for the system: at this stage, without much information and users participation, the system can not distinguish between good and bad tags and resources. This is confirmed by participant 8: she also associated several not relevant tags; in this case, however, the associations among context descriptor and tags computed by the system after the first seven participants are strong enough to prevent bad tags from standing out.

A second point of interest is that, except for the hole, the general trend in Figure 11b is slightly increasing. This means that, through user participation, the associations between situations and tags provided are refined, and the most relevant tags stand out increasing the effectiveness of the system. Finally, in Figure 11b we can notice the first two experiments are the ones with the lowest nDCG value (lowest lines). These experiments are those where less tagging operations have been performed (Section 5.3), thus lower is the number values inserted, the more the SCAB is vulnerable to bad tags (like those inserted by participant 2).

All these results answer questions Q1.1 and Q2.1.

5.4.3 nDCG of resources

Figure 12 shows the nDCG values for the resources retrieved by the system during the different experiments. In this case we have three graphics (one more than in

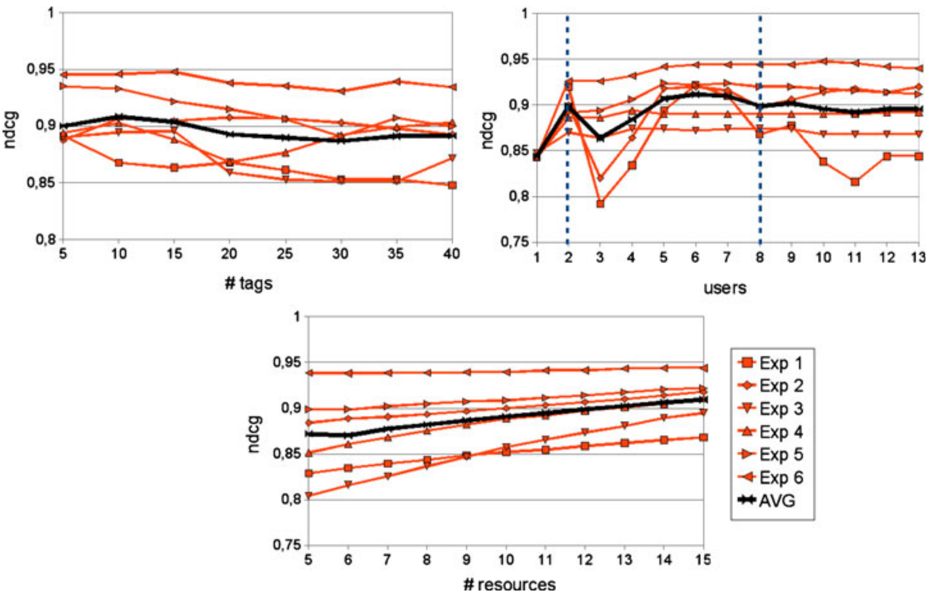


Figure 12 Mean nDCG for the resources retrieved, on all contexts.

Figure 11): since we are working with resources, we are interested in showing also how the retrieval effectiveness depends on the number of retrieved resources. As we can see the general results are slightly lower when compared to the results related to tags. This is due to the double retrieval: first the tags that best describe the user situation are retrieved, then starting from them the most relevant resources are retrieved. In this way, the retrieval errors due to not relevant tags are added to errors due to not relevant resources. This answers question Q1.1.

Considering the number of retrieved tags, we observed the highest nDCG for resources value with 10 tags , thus the increase of the number of tags retrieved by the system does not improve the overall effectiveness. This answers question Q1.2. On the contrary, considering the number of retrieved resources, the effectiveness improves increasing the number of resources, having a maximum at 15 resources. This means that more are the resources retrieved, the higher is the number of relevant resources among the ones retrieved; this answers question Q1.2. Observing the effectiveness related to the participants behaviour, , we can make the same considerations stated for the tags. For the first participants, because of the lack of information, the effectiveness of the system is strictly related to the information inserted by each single participant (e.g., participant 2): this means that the system will have good/bad performance based on the relevance of the resources associated by each single participant. With the increasing of the number of participants, the system performance tends to steady and to be less dependent on a single user action. Once many users have associated tags and resources to the same or similar sensors data, subsequent too bad (too good) information from a single user, like participant 8, does not heavily influence the system; only the behaviour of the whole community of users will have effect. This answers question Q2.1.

5.4.4 Similarity

Besides the nDCG values for tags and resources, we studied the average similarity of the sets of tags inserted by participants, with the sets of tags retrieved, at each step, by the system (and the same for resources). The step-by-step similarity has a correlation value of 0.64, while the final on 0.75. Even if the two results are not very high, they show that there is a correlation among the values: the retrieved tags are more similar to the tags inserted by users, when these ones are good tags; on the contrary when users insert not relevant tags, the similarity values decreases.

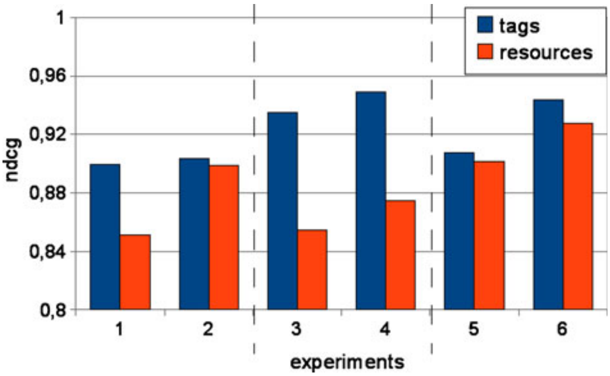
5.4.5 Experiments' effectiveness comparison

Some last considerations can be made observing the results from the point of view of the six experiments performed, described in Section 5.3. There are three groups of two experiments each, the former exploiting tags and resources freely chosen by users, the latter tags and resources judged relevant by each user.

Considering the nDCG value for tags and resources (Figure 13), in each group, the second experiment has always higher values than the first one. This means that if the user can see the tags or resources already inserted and choose among them, the system becomes more effective. Thus the integration of a tags suggestion module can be very useful for the system.

Analyzing tags (see figure), we notice that the second group of experiments has higher values than the first one. This means that carrying on the association process

Figure 13 nDCG values in the different experiments, for tags and resources.



refines the tags in the system, making the more relevant tags to stand out. On the contrary, introducing the possibility of removing tags (third group) does not improve the effectiveness, as the average nDCG value are slightly lower than the second group values. This is probably due to the subjectivity of tags: since users have different opinions about tags, it happens that a tag removed by a user is added again later on by other users, thus balancing the results and making the removal operation useless.

The situation is different for the resources nDCG values; the third group of experiments has the highest values, and the second group has lower values than the first. The main reason for this decrease is that the best resources are not always related to the best set of tags. In particular, the information refinement achieved in the second group experiments has increased the importance of tags that from one side are very relevant, but from the other side that have less relevant resources associated: this increased importance is spreaded to the related resources outcropping also those ones not relevant. The possibility of explicitly defining that a resource is not relevant for a context (removal) solves this drawback: this is the reason why the third group experiments have the highest average nDCG values. Consequently we gather that the possibility of removing not relevant resources can improve the overall SCAB effectiveness, while the removal of tags can make it worse. This answers question Q2.3.

5.5 Discussion

From this experimental evaluation we have the confirmation that the proposed approach is feasible; even with a limited number of participants, the collected data demonstrate that the system works and it has good potentialities. The social approach refines the information collected in the system and increases its relevance.

The nodal point is represented by the users and, in particular, by the quality of tags and resources they associate. This means that the system is useful if and only if users provide relevant contents. Otherwise, system effectiveness would be low and the "healing" in the results shown in Figures 11b and 12 would not be possible.

The whole community of users, through the interaction with contextual information and resources, defines what is good and bad in a context. However what is good for the community, is not necessarily good for each single user: people are very different and their information needs are very different as well. A small group of users, with information needs very different from the community ones, represent the typical case in which our system fails. However this happens just in the case of a small number of users: if this number increases, the weight of all the information changes, and thus the resources contextually suggested. This means, as demonstrated by the results, that the SCAB approach is not perfect, but it is statistically good. A further personalization layer, as described in Section 3.2.2, would combine community generated contents with user preferences in order to adapt the general system to each single user needs.

In our system we work on different levels: physical data, contextual tags, and resources. In particular the tags layer allows to decouple resources from physical data, in order to make the system more general: resources associated to the context `park` are valid for all the parks, without the need of associating each resource to each physical park in the world. However this abstraction brings a further issue, since not always the best set of tags describing a context allows to retrieve the best set of resources. As we noticed in the results (Section 5.4.3), in some cases increasing the quality of tags leads to a decrease of the quality of the final resources. This aspect is not critical, but deserve further investigations in order to increase the overall system effectiveness.

A last considerations concerns the weight of the single users in the overall process. We observed that the first users that participated to the test generally have the highest scores. For this reason we performed a further experiment: we repeated the previously described experiments (see Section 5.3) changing the users' order. In particular we executed the experiments 100 times with a randomly generated users' order; at the end we measured the average users' scores for contexts and resources. Figure 14a shows the average score for the experiments not considering the possibility of removing tags and resources. As we can see, first users have the highest scores. This situation is normal, since the proposed approach favours users that participate first. In fact they have the possibility to choose among large amount of contents and thus they have higher probability to associate the best tags and resources, increasing the probability to have good scores. In addition the systems awards only the users that create a relation between concrete and abstract tags, and tags and resources.

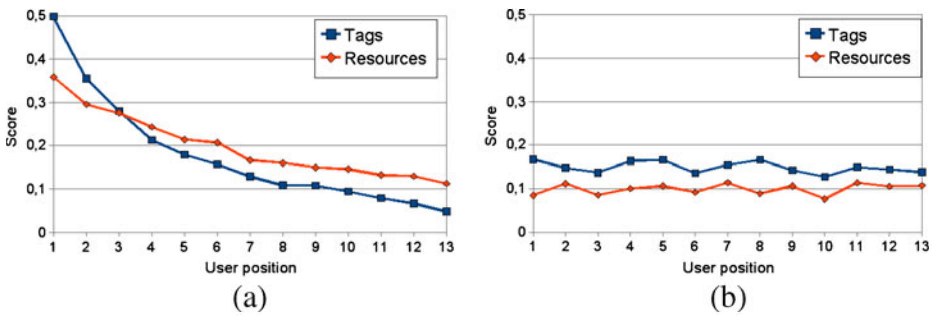


Figure 14 Users' average score for tags and resources. Simple (**a**) and considering the removal possibility (**b**).

When considering also the possibility of removing tags and resources, the results are very different, as shown in Figure 14b: there is a small difference among these scores; in addition the scores are lower and they are very near to the score of the last users in the previous experiment, that does not consider the removal possibility. In this case, although the first users have a high probability of associating the best tags and resources, they have also a high probability to be penalized by subsequent users, when they provide bad information. This answers question Q2.2.

6 Conclusions

In this paper we have presented the Social Context-Aware Browser, a general purpose solution to Web content and services perusal by means of mobile devices. Introducing the main ideas, we have shown how current approaches to contextual knowledge management are unsuitable for our solution. Thus the SCAB is not merely an application, but it is a novel paradigm for the information access based on context, where the community of users is called to manage the contextual knowledge through collaboration and participation. We presented the underlying motivations, the conceptual model, and the implementation approach. Then we described an experimental evaluation discussing the results and the lessons learned.

As future work, we aim at studying privacy issues: as all context data are communicated to a central server, serious privacy issues arise, and techniques for privacy protection must be adopted. In addition, we plan to adapt the social computation to each user profiles and needs, through a personalization layer. Since our project is community based, users participation and motivation are essential, and this leads to the question: "why the users should use our system?". We have three main reasons: first, our system could be an important added value to the users, that can improve their experience in context-aware and proactive Web contents perusal. Second, we consider the popularity of the Web 2.0 services we have been inspired by: people are used to tag and share information exploiting different services (e.g., Youtube, Facebook, Foursquare, etc.). Third, context-aware retrieval is gaining increasing importance at a fast pace: the 85% of the participants in a study on mobile information needs [21] responded positively about a tool that could predict their information needs and provide appropriate information at right time.

However, only an appropriate user testings can answer to that question. For this reason, we aim at performing a broader and more complex user-centered evaluation. This will help us to understand if the SCAB and the social approach is effective in the real world. This stage will involve studies in a full mobile and real world environment, where we must move beyond performance and usability and consider utility or impact measures. That is, how do the proposed system change the work that users are doing? We want to study how people interact with the system, how they perceive it, and how the same system is useful in simplifying Web interaction in the everyday life.

Moreover, we plan to study the concept of subcommunities in order to understand if it can be benefical in the SCAB approach. In this paper we have worked just with a single community of users: this simplifies the understanding of the general ideas behind out approach but some issues can arise, mainly related to the users' goodness. For example, an on average good user can be very good with contexts and resources related to art and museums, but bad with those ones related to sport. Therefore, instead of having a single huge community, we can work with different subcommunities, in order to make the user evaluation and the whole process more effective.

Finally, the vision we described represents an extension of the idea of Web, where resources and documents are indexed not only based on their content, but also based on the context they are relevant to. A future step is to answer to some questions generated by this work. Could this model be helpful in understanding the relationships between context and content? Which is the connection between context and information needs? Is the context alone enough to understand what the user needs and how she needs it (textual information, audio, image)?

Acknowledgements We acknowledge MoBe srl. (www.mobe.it), that partially supported this research, the participants to the test, and the reviewers for the useful comments.

References

1. Broll, G., Siorpaes, S., Rukzio, E., Paolucci, M., Hamard, J., Wagner, M., Schmidt, A.: Supporting mobile service usage through physical mobile interaction. In: Proc. of IEEE Int. Conf. on Pervasive Computing and Communications, pp. 262–271 (2007)
2. Ceri, S., Daniel, F., Facca, F.M., Matera, M.: Model-driven engineering of active context-awareness. World Wide Web **10**, 387–413 (2007)
3. Chen, H., Mohapatra, P.: A context-aware HTML/XML document transmission process for mobile wireless clients. World Wide Web **8**, 439–461 (2005)
4. Coppola, P., Della Mea, V., Di Gaspero, L., Menegon, D., Mischis, D., Mizzaro, S., Scagnetto, I., Vassena, L.: The Context-Aware Browser. IEEE Intell. Syst. **25**(1), 38–47 (2010)
5. Golder, S.A., Huberman, B.A.: Usage patterns of collaborative tagging systems. J. Inf. Sci. **32**(2), 198–208 (2006)
6. Gonzalez, M.C., Hidalgo, C.A., Barabási, A.L.: Understanding individual human mobility patterns. Nature **453**, 779–782 (2008)
7. Graham, R., Eoff, B., Caverlee, J.: Plurality: a context-aware personalized tagging system. In: Proc. of the 17th International Conference on World Wide Web, WWW 2008, pp. 1165–1166 (2008)
8. Jones, G.J.F., Brown, P.J.: Context-aware retrieval for ubiquitous computing environments. In: Proc. of Mobile HCI Workshop on Mobile and Ubiquitous Information Access. Lecture Notes in Computer Science, vol. 2954, pp. 227–243. Springer (2004)
9. Keidl, M., Kemper, A.: Towards context-aware adaptable web services. In: Proc. WWW Alt. '04, pp. 55–65. ACM (2004)
10. Korpipää, P., Mantyjarvi, J., Kela, J., Keranen, H., Malm, E.: Managing context information in mobile devices. Pervasive Computing **2**(3), 42–51 (2003)
11. Lopez de Ipiña, D., Vazquez, J.I., Abaitua, J.: A context-aware mobile mash-up plaftorm for ubiquitous web. In: Proc. of 3rd IET Intl. Conf. on Intelligent Environments, pp. 116–123 (2007)
12. Menegon, D., Mizzaro, S., Nazzi, E., Vassena, L.: Evaluating mobile proactive context-aware retrieval: an incremental benchmark. In: Proc. of the 2nd International Conference on the Theory of Information Retrieval (ICTIR), pp. 362–365 (2009)
13. Mizzaro, S., Nazzi, E., Vassena, L.: Retrieval of context-aware applications on mobile devices: how to evaluate? In: Proc. of Information Interaction in Context (IIiX '08), pp. 65–71 (2008)
14. Mizzaro, S., Nazzi, E., Vassena, L.: Collaborative annotation for context-aware retrieval. In: Proc. of the WSDM '09 Workshop on Exploiting Semantic Annotations in Information Retrieval, ESAIR '09, pp. 42–45. ACM (2009)
15. Myrhaug, H., Whitehead, N., Göker, A., Faegri, T.E., Lech, T.C.: Ambiesense—a system and reference architecture for personalised context-sensitive information services for mobile users. In: Proc. of the Second European Symposium on Ambient Intelligence (EUSAI '04), pp. 327–338. ACM (2004)

16. O'Reilly, T.: What is Web 2.0, Design Patterns and Business Models for the Next Generation of Software (2005). URL: http://www.oreilly.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html
17. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-oriented computing: state of the art and research challenges. Computer **40**, 38–45 (2007)
18. Sanderson, M., Joho, H.: Forming test collections with no system pooling. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '04, pp. 33–40. ACM, New York (2004)
19. Schmidt, A.: Ubiquitous Computing—Computing in Context. Ph.D. thesis, Lancaster University (2003)
20. Sheng, Q.Z., Pohlenz, S., Yu, J., Wong, H.S., Ngu, A.H., Maamar, Z.: ContextServ: a platform for rapid and flexible development of context-aware web services. In: International Conference on Software Engineering, pp. 619–622 (2009)
21. Sohn, T., Li, K.A., Griswold, W.G., Hollan, J.D.: A diary study of mobile information needs. In: Proc. of CHI 2008, pp. 433–442 (2008)
22. Sungrim, K., Joonhee, K.: Folksonomy-based information retrieval in context-aware environment. Int. J. Comput. Sci. Netw. Secur. **8**(11), 252–257 (2008)
23. Sungrim, K., Joonhee, K.: Information retrieval using context information on the Web 2.0 environment. Int. J. Comput. Sci. Netw. Secur. **9**(10), 62–65 (2009)
24. Truong, H.L., Dustdar, S.: A survey on context-aware web service systems. Int. J. Web Inf. Syst. **5**(1), 5–31 (2009)