Università degli Studi di Torino
Scuola di Dottorato in Scienza e Alta Tecnologia

Tesi di Dottorato di Ricerca in Scienza e Alta Tecnologia
Indirizzo: Informatica

Augmenting User Experience in the Social Web by Means of Storytelling and Semantic Web Techniques.

Elisa Chiabrando

Tutor: Claudia Picardi, Ph. D.

XXV Ciclo, Novembre, 2012

Università degli Studi di Torino
Scuola di Dottorato in Scienza e Alta Tecnologia

Augmenting User Experience in the Social Web by Means of Storytelling and Semantic Web Techniques.

Elisa Chiabrando

Contents

List of Figures

List of Tables

Abstract

The present work is situated in the context of social and mobile application, and in particular in the context of a Social Web of Intelligent Things (*SWIT*).

We aim at augmenting the user experience by (a) introducing elements of Interactive Digital Storytelling, (b) exploiting semantic web techniques to capture and express conceptual similarity, and (c) thanks to such techniques, providing users with content they are interested in.

Concerning storytelling, we introduce a new kind of user-generated content, called "story fragments" or "facts", which users can provide to share their experience in a social and mobile context. Story fragments are machine processable and linked with semantic data describing the elements taking part in them. We propose to use story fragments as captions or "advanced tags" for other content types (*i.e.* videos or images) in order to be able to aggregate them according to semantic relations and storytelling principles. Story fragments describe "actions"; the correlation between them depends strongly on the situation or event where such action takes place. We devoted part of our research to building a better representation for events than we found in existing social networking environments, and in finding what factors make an event interesting for a given user.

Since we are interested in social media and services that exploit the Semantic web, we developed a similarity measure between ontologically defined concepts. This measure can be used to directly correlate story fragments, based on the things or people they talk about, or to find similarities between a users' interest and a given content type. This is relevant to the last part of our work, where we investigate the recommendation of story fragments and events by means of user modeling. For this purpose, we see an event as an aggregator of "story fragments", focused on a topic that might be interesting for a user. Similarly, "story fragments" are correlated and filtered according to users interests and to their pertinence.

1
Introduction

The research I will present in this thesis is situated in the context of social and mobile applications and is part of PIEMONTE[1] [4][5][6] (*People Interaction with Enhanced Multimodal Objects for a New Territory Experience*), a project supported by Regione Piemonte[2] whose partners were Department of Computer Science[3] (University of Turin), University of Gastronomic Sciences[4], the telecommunications company Telecom Italia[5] and Slow Food[6], a no-profit association for the promotion of "good, clear, fair food".

PIEMONTE (2009-2012) had at its core the idea of a "Social Web of Intelligent Thing", *SWIT*[23] for short, which is an evolution of the "Web of Things" and the "Smart Objects" paradigms.

In a *SWIT*, ordinary real-life things are enriched with intelligent and social capabilities that allow them (a) to interact with the people in a natural and personalized way, (b) to manage and exchange information with users and with other *SWIT*-things, and (c) to have social abilities, such as the possibility to establish relationship with other things and people.

[1] PIEMONTE project `http://www.piemonte.di.unito.it`
[2] Regione Piemonte `http://www.regione.piemonte.it/`
[3] UNITO `http://www.di.unito.it`
[4] UNISG `http://www.unisg.it`
[5] Telecom Italia `http://www.telecomitalia.com`
[6] SlowFood `http://www.slowfood.it`

In [23] the *SWIT* is been defined in two steps: (a) enunciating its *manifesto* as a set of principles of *SWIT* paradigm, and (b) describing functions and opportunities that a *SWIT* framework offers to people.

1.1 The SWIT manifesto

According to [23], the *SWIT manifesto* identifies four principles that allow things to become entities capable of intelligent and social behaviors:

(1) *SWIT* is about real-life concrete things with intelligent and social capabilities that allow them to interact with people and other *SWIT*-things.

(2) the enhanced capabilities of a *SWIT*-thing must be accessible by people in real life with a *bidirectional* and *natural* interactions. The idea is that either technology is embedded within the objects themselves or, less invasively, things are recognized by a mobile device (by means of image recognition or other cues). In any case, users interact with a digital avatar of the object, but the goal is to do so in a way that does not disrupt the flow of physical interaction in real life.

(3) *SWIT* intelligence allows things to *reason on* and *learn from* semantic information on a given domain, from user behavior during interaction and from different user-generated contents that others associate with the thing (*i.e.* photos, comments, tags).

There are five categories of intelligent functions:

- *content aggregation*, the capability to filter, synthesize and mashup knowledge and contents ranging from their intelligent selection to digital storytelling;
- *adaptation* to personalize the interaction with the user;
- *knowledge socialization* to share knowledge with other things;

- *social linking*, the capacity to discover ties between people and things, whether provided by the users themselves or suggested by what users do;
- *serendipity*, to discover new unexpected things and people by exploiting relations.

(4) *SWIT*-people and *SWIT*-things are *social actors* in a social network, they can have their friends and share information about themselves. *SWIT*-things can use their reasoning capabilities to discover similar or related things and to befriend them. Thus, the possible relations are: *user-to-user* (*i.e.* with relation of friendship or similarity), *user-to-thing* (*i.e.* with social actions, such as tags or likes), but, also, *thing-to-thing* (*i.e.* with relations of friendship or relatedness).

1.2 Functions and opportunities offered by a SWIT

A second characterization of a *SWIT* is given by the functions and opportunities it provides people with. They can be summarized as follows:

(1) Users on the move may *get in touch with* a *SWIT*-thing in two ways: (a) with direct contact when he starts a directly interaction with the thing (*pull mode*) or when the thing calls the user (in *push mode*); (b) otherwise by indirect mechanisms, such as *search engines*, *location-based access* or *recommendation* services.

(2) After contact the user can *access information about the thing*, both a general description about the thing itself (*i.e.* photos, description, location), social information provided by the users (*i.e.* tags, votes, comments, network of people who like the thing), social relationship of the thing.

(3) At any time, the user may *perform* some *social actions*: (a) to share other information with his community about a real thing (*i.e.* comments), (b) to express his opinion (*i.e* votes or tags) or (c) to bookmark it in his personal preferred list.

(4) The user can *navigate social networks* of users and things and can perform actions on both users and things in the similar way.

(5) The framework supports a *continuum of experience* of seamless interactions between things and users in reality and in a virtual environment (*i.e.* on the Web).

(6) Users and things must register on the system. Thus, the framework includes *user registration* functions for the users and *administration* functions for the things.

1.3 Wanteat: an implemented SWIT

The original *SWIT* proposal identifies "ingredients" for a possible architecture for a *SWIT* framework. The PIEMONTE project further developed a system, called *Wanteat*, that implements such an architecture, and offers a suite of applications that follow the *SWIT* principles. *Wanteat* focused on the domain of quality food and wine, and allows people to interact "*SWITly*" with food-related objects, people and places, to discover the cultural heritage of a territory starting from its cuisine and its markets.

Wanteat is an evolution of *Social Media*, builds upon *intelligent adaptive systems* and exploits ontologies for its knowledge base. The *Wanteat* system explores *user modeling* for a personalized interaction and *learns from users' behavior* to understand their interests, to improve their models and to discover new *social relations* between users and other users or things.

Relations between things are referred from ontologies and users' behavior. In fact, relationship between things may depend on their native and structure, as well as how users perceive them as similar, or related.

Wanteat has several system ontologies describing the main domain of application, gastronomy. Ontologies identify and detail the different foods and wines as well as the places where products can be tasted or bought, the geographical region where they belong, and the actions that people can perform on them.

Thus, the key ideas behind the *Wanteat* system are:

1. building and maintaining a social network of both people and things,
2. allowing users to interact with things belonging to the social network in augmented reality,
3. turning things into hubs that connect users with a larger world.

The development of *Wanteat* saw also the design and implementation of a novel interaction paradigm, called "the wheel" [14], that provides an intuitive navigation mechanism for the complex social and content network modeled by a *SWIT*.

1.4 Research aspects connected to the overall framework

Having introduced *SWIT* in general and *Wanteat* in particular, we can discuss how the different parts of this work are connected to the overall framework.

Also, using *interactive storytelling techniques* and *agent-based* technologies the things may acquire social intelligent behavior and they can synthesize and tell users to interesting stories, created by information from different sources.

Our first goal was to explore interactive digital storytelling techniques and improve intelligent and social abilities of things, and in particular in order to *aggregate user-generated content* in an interesting way.

Starting from the functionalities that a *SWIT* offers to the user, we can consider two types of user-generated content: (a) structured information or (b) unstructured information. Structured information is provides by votes, tags and bookmarks, but these actions not allow user to tell what he want and it strongly limits users' expressivity.

On the other hand, we have unstructured information, such as free-text comments: with these, the user may freely share his opinions but it is very difficult for the system to infer something or aggregate it with other content.

For these reasons, we introduce a new kind of user-generated content, the "story fragments" or "facts".

Story fragments describe "actions" by means of simple structured sentences, composed by a central predicate and a few roles that can be filled by entities in the system domain. Facts are machine processable and they are linked with semantic data describing the elements taking part in them. So, they are structured, but at the same time they allow significant freedom to the user, who can build facts by combining any element he wishes.

Story fragments can be used as captions or "advanced tags" for other content types (*i.e.* videos and images) so that they can be exploited for aggregation. Thus, users can share their experience in different ways in his social network, also from a mobile context. A considerable effort has been devoted to devise a mobile interface for guiding users in building facts.

Since we are interested in social media and services that exploit the Semantic web, we developed a semantic similarity measure [3] between ontologically defined concepts, build upon an existing similarity measures in the literature.

We proposed to use this measure to calculate the correlation between "story fragments" or "facts", based on the things or people they talk about, or to discover the similarity with a given content type or also to find similarities between a users' interest.

The correlation between two "story fragments" depends strongly on the situation or event where such action takes place. Users are influenced from the event where the actions are described and two different "story fragments" can become more similar to each other if contextualized in an event.

Due to this evidence, we devoted part of our research to building a better representation for events than we found in existing social networking environments (*i.e.* events in Facebook[7] or in Google+[8]), and in finding what factors make an event interesting for a given user (*i.e.* location, thematic, typology of the event). We hypothesized a possible events recommender and we created a simulator to evaluate our proposal.

In summary, our research evolved along three lines:

- we propose to use "story fragments" of "facts" as *advanced tags* (Chapter 3) to describe contents and use them as meta-information for selection, aggregation and presentation,
- we define a semantic similarity measure (Chapter 4) to compute the similarity of two story fragments, starting from semantic information on the things and people they mention,
- we analyze the concept of event as aggregator of "story fragments" about a topic, and study the relevant features that make it interesting for users (see Chapter 5).

1.5 Outline

The dissertation is organized as follows.

[7]Facebook `http://www.facebook.com/`
[8]Google+ `http://plus.google.com/`

In Chapter 2 we give an overview of related work that is relevant to our research. Chapter 3 introduces the research on "story fragments" or "facts", and discusses how to use the underlying ontologies to correlate them. Chapter 4 then presents the semantic similarity measure we devised with this goal in mind. Chapter 5 introduces the notion of "spatial-temporal object", as a representation of the context for a fact, and discuss a study on the factors that influences users' preferences and interests concerning events in a social context. Chapter 6 discusses the part of our work that focused more on user interaction. In particular, it presents two prototypes: *Wanteat Video*, which recommends videos in the *Wanteat* context expanding on the paradigm of "the wheel", and *Telleat*, the *Wanteat* extension that allows users to provide facts concerning entities in the *Wanteat* domain.

2

Related Work

The proliferation of low-cost pervasive and personal technology (mobile devices, internet connection) and social platforms changes the way we interact with the digital world:

(a) Twitter[1], blogs and online newspapers become the main channel for staying informed about world events,

(b) Facebook and Google+[2] keep us informed about news of our friends,

(c) aNobii, LibraryThing, Last.FM, Youtube, Flickr and del.ici.ous[3] collect content and collective preferences,

(d) Wikipedia[4] represents the common knowledge of internet users, that create and modify information in a collaborative way.

Furthermore, the technological evolution of devices (RFID, NFC, barcode, bluetooth, camera) allows the user to interact directly with real objects

[1]Twitter `http://www.twitter.com`

[2]Facebook `http://www.facebook.com`, Google+ `https://plus.google.com`

[3]aNobii `http://www.anobii.com/`, LibraryThing `http://www.librarything.it/`, Last.FM `http://www.lastfm.it/`, Youtube `http://www.youtube.com`, Flickr `http://www.flickr.com/` and del.ici.ous `http://delicious.com/`

[4]Wikipedia `http://it.wikipedia.org/wiki`

and locations. Applications such as Foursquare[5] and Instagram[6] capture real moments in a picture or in a geo-located coordinates and create an instant link between the digital social context and the real place. aNobii allows user to scan the barcode of a book and add it directly in his digital library. Thus, real objects became part of the digital world, with their descriptions, connections to people, images and geo-location coordinates and, at the same time, new objects are created in this virtual world (*i.e.* fan pages, groups, pictures and other media).

In the literature we find many studies aimed at sorting, filtering and selecting content in an intelligent way, in order to propose to users a selection that is effective and interesting to them.

Due to the widespread availability of smartphones, that allow users to access this information virtually anywhere and anytime, the selection and representation of "interesting" content requires to take into account the context (including where and when of the uses).

The content to be presented can be "user-generated" content, that people have chosen to share on the web (typically, on social networking platforms), but it can also come from RSS feeds and search engines. The selection, presentation and possibly recommendation of such content takes advantages not only of information of the user (profile, context, friends, past interactions, etc.) but also meta-information on the content itself, that is usually provided by the user (ratings, tags, opinions, etc.).

In the rest of this Chapter, we discuss relevant work in the literature on the three themes that have been key to our studies: *interactive digital storytelling*, especially in social/mobile environments (Section 2.1), semantic similarity measures (Section 2.2), and recommendation of events (Section 2.3).

[5]Foursquare `https://it.foursquare.com/`
[6]Instagram `http://instagram.com/`

2.1 Interactive Digital Storytelling

In Janet Murray's book "Hamlet on the Holodeck: The Future of Narrative in Cyberspace" [42] the author shows how computers are reshaping the stories we live by.

Stories are important for our life because they define how we think, play and understand the world around us. She discusses the properties and pleasures of computers: the procedural, participatory, encyclopedic and spatial characteristics and three main pleasures (immersion, agency and transformation) that provide the basis for an expressive narrative digital form. Murray believes that using the machine formats possibilities of expression available for storytelling considerably increase, by connecting research work on artificial intelligence with cultural forms (games, literature, television, movies, picture).

According to Murray, *Interactive Digital Storytelling IDS* is born in computer games to allow people to interact with the storyteller and characters in real-time. By a mixture of storytelling, user interaction and game technology, users can generate a dynamic and personalized story in the game timeline. Presently, however, *IDS* is an open problem in the literature and covers different contexts with several goals. *IDS* aims at creating a bespoken story according to user interactions with the system.

We are particularly interested in *IDS* research that takes into account mobility of users, social and collaborative story building, and natural interfaces for story authoriting:

- some projects use the available digital location information and *IDS* strategies to support user experiences in mobility [43][25],
- others are concerned mainly with educational aspects and collaborative creation of stories [16],
- others are focused on graphical authoring [27][53][18] .

For the user in mobility, we have "A Story to Go, Please" [43]. It is a location-based storytelling that supports users in mobility, thought a guided story space exploration. In this article the authors describe a system that automatically creates associative stories about interesting city spaces in the user location. It uses digital visual content uploading recently by other people in the same place and leaves the completion of the story to the visitor's own motivational and psychological attributes (for example the story of a specific user in a place will depend in part on the associative story system, in part from what at that moment the user wants to visit).

In this work the authors point out three essential elements:

- the content structure that allows the system to generate an associative story;
- the metadata structure and the geo-tagged data (position and orientation), that allow the system to select the interesting content and the way to present it;
- the generation rules to cause first the clustering of content into hypespots (a spatial unit that defines a real place covering approximately 150 meters), and second the creation of associative stories.

By these essential elements and despite some limitations (e.g. user have to tag their content), the proposed method might generate location-based stories on the fly for enhancing the user's experience on the move.

An implementation of a client-server platform for mobile storytelling is InStory [25]. The client device is able to store the users positions and their actions and send the information to the server database. All user can decide to join a group and interact with an other user or his group by sending an instant message. An *ah-hoc* group is also spontaneously created as a result of user characteristics, their current position and some kind of events. Moreover, the same users can upload different types of data (text or images) to integrate them into the system and particularly into the story created by the system. In this way the system always presents new media materials to the user.

In an educational context, we have PoliCultura [16], a project targeted at Italian school children which should enable the students of different age groups to design and construct interactive stories over a longer period of time. *1001stories* is a web-based authoring-delivery environment which enables children to combine images, text and mp3 files into interactive stories. The authors present their work as a platform encouraging *collective narratives*, but it seems that it is more similar to a system supporting *collaborative narratives*, where many authors collaborate in constructing a story. On the other hand, in our framework we try to enable real *collective narrative* construction where authors can work independently of each other when contributing their facts to the system and conceiving a story.

An interesting authoring environment is INSCAPE [27], a software tool for non-specialist user to create and experience interactive stories and simulations. Users can design interactive storyboards, edit and visualize the story structure, create 2D and 3D scenes and characters, incorporate various multimedia, such as sounds, pictures and videos, and publish the stories on the Internet. On the contrary of linear representation of some important applications such as Macromedia Director, Adobe Premiere and Flash, topological graphs are used to visualize stories where nodes are objects of the story and edges are interactive transitions or conditional relationships. A multimedia storytelling platform to create non-linear stories similar to INSCAPE is MIST [53] that uses MPEG-7 multimedia metadata standard in order to successfully combine various media types into stories. The users can create new or edit the existing stories, as well as just read the existing ones. The stories in MIST consist of elements containing structural information and elements containing media specific descriptions. Links between story elements are made using media files and their descriptions. MIST does not support the collaborative storytelling paradigm (each user can only create his/her own story). In [18] an extension of MIST, called PESE is proposed which takes into account Web 2.0 technologies. A user model is conceived where users with different roles can perform different media operations. In this framework many handheld clients can connect to a centralized server and subscribe to various stories at the same time.

2.2 Semantic similarity for content clustering

The content of a system can be described by sentences, phrases, titles, captions and tags. If we analyze the words within them, we can clustering content in some way according to their semantic meaning, without worrying about the nature of the content (images, videos or simple comments). This assumes that the multimedia content have been previously described by a user in a semantically meaningful way, which is quite common in a social system where the user would publish and share his content.

The final goal is to show contents from the same cluster to the system user and to ensure that they look pertinence in their meaning to him.

Having this goal in mind, we look for some measures available in the literature (we review them in the last part of this chapter) which can help to compute the semantic similarity between concepts and words.

In the literature there are various proposals for the calculation of semantic similarity between concepts. We can distinguish measures that compare individual ontology terms with others and measures that compare sets of terms.

In the first case the main approaches are *topological similarity* and *statistical similarity*. *Topological similarity* uses the available semantic information (thesauri, taxonomies, dictionaries and other kind of pseudo-knowledge bases) to define the distance between words, while the *statistical similarity* computes a model to understand the distance between concepts.

In the second type of measures, the comparison between sets can be made simply combining the semantic similarity (calculated using the approaches between terms) between all pairs of terms (*pairwise method*) or considering the sets of terms as mathematical sets, vectors or graphs (*groupwise method*).

2.2.1 Topological similarity approach

All concepts are represented in a multidimensional space as nodes and the edges define a direct association between them. The association is a concepts relationship, that can better quantify their semantic similarity.

There are two kind of topological strategies:

- the *edge-based strategies* use the information about the edges as data source (Rada [47], Sussna [54], Richardson [51]) and calculates the similarity between two concepts by the geometric distance between the corresponding nodes,

- the *node-based strategies* use the information about nodes and their properties (Resnik [49], Lin [38], Jiang and Conrath [34], Couto, Silva and Coutinho [26]) and calculate the semantic similarity as the share information in common about two nodes linked about edges.

The strategies are quite different and they have inherent strengths and weaknesses depending on the nature of strategy.

On the one hand we have the natural and intuitive *edge-based strategy*, that resolves the semantic distance using the detailed structure of a taxonomy. The hierarchical relations and features of taxonomy (its density, the domain of concepts, the types of relationships) influences positively or negatively the calculation of semantic distance. Rada et al. [47] and Richardson and Smeaton [51] demonstrate that the same distance method works efficiently or not depending on different domains and different taxonomies. In the medical domain of Rada the distance measure simulated human assessment with surprising accuracy, while the Richardson 's semantic distance are less accurate than expect, because it is susceptible to the structure of WordNet, a lexical database created by more people and with irregular densities of links.

On the other we have the *node-base strategy* that are not affected by the structure of taxonomy as early strategy and it is not sensitive to different kind of links [48], but this is also a weakness. It can generate coarse results, because it not distinguish pair of concepts in a sub-hierarchy with the same "smallest common denominator" (*i.e.* the strategy can give the same result for the pairs "screwdriver-table ware" and the pair "screwdriver-fork" with the common denominator "instrument" and "fork" as a child of "table ware"). Also it is more sensitive to polysemous words and multi-worded synsets, creating an exaggerate information content value [51].

For these reasons, *hybrid strategies* are developed to reduce weaknesses. Jiang and Conrath [34] described previously are an example of hybrid strategy, but we can also remember Richardson [51] or Smeaton and Quigley [52] (see section 2.2.3).

2.2.2 Statistical similarity approach

Other approaches define a **statistical similarity** between words.

In [29] the author uses a statistical technique LSA (Lantent Semantic Analysis) to compute model and simulate the semantic distance of words. LSA assumes that similar words occur in the same part of text. Given a document and the mathematical technique SVD (Singular Value Decomposition) it makes a matrix with paragraphs in the rows and counts of words in the columns. The cosine of the angle between any paragraph vectors is the comparison between words and the result is a numerical value in the range $[0, 1]$ (1 for very similar words, 0 otherwise).

Other works use web platforms as information source to measure the similarity between concepts. Cilibrasi and Vitanyi [22] define the concept of *Google Similarity Distance*, that uses the Google results and rankings. The idea starts from consideration that words and phrases are used in society and acquired meaning from the way they are used. If the equivalent of "society" is "database", the equivalent of "use" is "a way to search content in the database". So the database becomes the *World Wide Web* and Google the search engine to extract information on the page ranking that expresses the way to search content. The measure of similarity in this case is based on information distance and Kolmogorov complexity. The *Normalized Google Distance* between two search terms x and y is:

$$NGD(x,y) = \frac{max\left\{\log f(x), \log f(y)\right\} - \log f(x,y)}{logN - min\left\{logf(x), logf(y)\right\}}$$

where N is the total number of web pages searched by Google; $f(x)$ and $f(y)$ are the number of pages containing terms x and y, respectively; and $f(x,y)$ is the number of pages on which both x and y occur.

Instead an another work of Gabrilovich and Markovitch [32] uses the information of Wikipedia to compute the semantic similarity degree between words and between fragments of natural language text. Each Wikipedia concept is an article and is defined previously by a learning algorithm as a weighted list of words that occur in the article. Also by *inverted index* it is possible to find all articles where each word appears and define the relevance of corresponding concepts. The ESA (Explicit Semantic Analysis) algorithm works with a *semantic interpreter* that maps words and fragments of a text into weighted vector of Wikipedia concepts. Thus, the semantic similarity between two texts is build with the cosine metric of their vectors.
The ESA algorithm obtains good results respect to human judgments.

2.2.3 Application of semantic similarity measures

A useful Perl module created by Ted Pedersen's team[7] implements a variety of semantic similarity and relatedness measures based on information found

[7]Pedersen's Perl module `http://wn-similarity.sourceforge.net/`

in the lexical database WordNet. It supports the measures of Resnik, Lin, Jiang and Conrath, Leacock and Chodorow, Hirst and St.Onge, Wu and Palmer, Banerjee andPedersen, Patwardhan and Pedersen. In some recent works it evaluates some modified methods with node-based measures [45], that performs better the calculation of semantic similarity without sense-tagged text used in previously works.

The research of better measure of semantic similarity are important because are widely used in more applications of sense disambiguation, paraphrase detection and question answering.

Smeaton and Quigley [52] describe an application that retrieves the image captions based on the user query together with experimental results and evaluation. Starting from a corpus of image captions and a collection of queries, they index the queries and the captions by the words occurring in them and then use semantic similarity between index terms to calculate the query-caption similarity. The word-word similarity is determined using a set of hierarchical concept graphs derived from WordNet [31] and the measure of semantic similarity is based on the work of Resnik [48]. Among the experimental runs of different set similarity algorithms, the runs most interesting for our work are: (i) the run introducing a word-word similarity threshold to eliminate the words with low similarity and (ii) the run in which the most similar caption term for each query term and the most similar query term for each caption term are calculated and used in overall sum of similarity values.

In the application described by Tudhope and Taylor [57], the similarity measure is used to improve automatic generation of links in hypermedia navigation, based on three measures of similarity: subject, temporal and spatial. In case of subject and spatial dimensions, they calculate the shortest paths connecting two terms. Each traversal between two directly connected terms has a corresponding cost factor associated to it and in the subject dimension the cost factor depends on the type of relationship and the depth in the hierarchy. This ensures that the siblings deeper in the hierarchy are semantically closer than the siblings higher up. In the spatial similarity measure a variation of a branch and bound search algorithm produces multiple paths to a solution where semantic similarity is below zero. The temporal similarity measure calculates similarity between two time periods or points in time. As in our case, they employ a maximal set similarity algorithm [55] which sums the maximum similarities for each term with respect to the members of the other set and normalizes them.

2.3 Recommender system of intelligent things and events

In the recent years recommender systems have been gaining popularity in e-services as means to solve the problem of information overload, by tailoring the system response with respect to the specific user's preferences and needs.

Recommender systems try to select and rank items (shopping goods, services, web resources, etc.) which are likely to be of interest for the user, basing their projections on a variety of sources such as user interest declaration, or user profiles inferred by past user behavior, such as rating, tagging and purchasing.

Thus, we have two kinds of recommender systems:

- *content-based recommender* systems [44, 39] that consider the content of the resource to be recommended and match it with the user's preferences.
- *collaborative filtering* systems, that works on users' information and suggest items that other similar users have positively evaluated. Two users are considered similar if they had evaluated in the same way a massive number of same items.

Compared to the *content-based recommender* systems, *collaborative filtering* systems do not have analyzed the content and the system can suggest complex contents, such as images or videos, without considering what they are about. Though, these system require existing data on user to give accurate recommendations (*cold start* problem), they have troubles to work with a large number of user for its computational request (*scalability* problem) and, finally, the increase of number of contents reduces the quality of the recommendation, because the user has a set of favorite contents very small compared to the total (*sparsing* problem).

In order to combine the advantages of both approaches (and limit the drawbacks), it is more common to mix both collaborative and content-based techniques, *hybrid recommender* systems.

Since our research is concerning these topics proposing focuses on events, in the following section we describe some interesting works on event recommendation.

2.3.1 Event recommenders

Events are very peculiar items composed by several aspects (temporal and spatial aspects are only the most obvious) and they have been studied in several works both to identify a common model [59][60][56] and to suggest them to user [36][20][41][46].

A complete event model is described within Utz Westermann and Ramesh Jain's article [60] for representing events in heterogeneous multimedia applications. The model is composed by six structural aspects (temporal, spatial, casual, experiential, informational and structural aspect) and must satisfy two important properties: extensibility and adaptability. According to the authors a high degree of extensibility and adaptability should include an event schema or an ontological representation.

An interesting work about events and media descriptions using an ontological representation is the article [56] where events are defined in terms of the *four Ws*(*who*, *where*, *what* and *when*). Their ontological representation is an extension of LODE[8] descriptions with other fields, such as event categories and causal relations between events. The authors aim to provide a web event-based environment where user can select, explore, annotate and share media of an event with other users.

Similar work is Eventory [59], an event based media repository where user can create, explore and manage events and their multimedia content. Events are real-world occurrences that unfold over space and time. They have standard facets of *who*, *where*, *what* and *when* as essential aspects that describe them and *how* and *why* as descriptions of relationships between events or their media. In fact, events can be enriched with images, sounds, videos and free text and user can create personalized relationships between events. The authors design two forms that allow user to become aware of communities activities (notification and subscription) but not a real mechanism to suggest new interesting content. Events are complex items, with a strong spatial-temporal connotation, and as such are only valid for a short period of time. For this reason, in the context of events recommendation, pure collaborative filtering techniques are rarely applied because feedback on the items to be recommended, which are in the future, is missing or the events in which the users participated are no longer available.

An example of *collaborative filtering systems* for recommending events is PITTCULT[9] [36], a cultural events recommender based on trust relations, where standard collaborative filtering techniques are applied to users which are explicitly rated and evaluated as trustworthy, instead of applying them to random ad hoc similar users. Although the social aspect of recommendation is very elaborated, content and context information are not considered.

Instead, a content-based event recommender is *iCity* [20], an application providing information about cultural resources and events in the city of Turin, Italy. It exploits users' behavior in the system to infer their inter-

[8] LODE: An ontology for Linking Open Descriptions of Events http://linkedevents.org/ontology/

[9] PITTCULT `http://pittcult.sis.pitt.edu/`

ests in various items and to improve content-based recommendations. The idea is to infer the strength of users' interests in a class of events from the kind of actions they perform on them, (*i.e.* inserting, tagging, modifying, commenting, rating or visualizing on a map).

As explained above, a *hybrid approach* can be more effective in some domain. A hybrid approach to event recommendation, i.e. the combination of content-based recommendation and collaborative filtering, performs better than other approaches on almost every qualitative metric (accuracy, diversity, novelty, etc.), as shown in [28]. Several works mix collaborative and content-based techniques [41] [46][24].

In [41] the authors propose a collaborative ranking of future events. Users' individual preferences for past events are represented as parameter vectors and matched with event descriptions. User parameters are decomposed into shared and individual components which are used to induce similarity between users and add collaborative filtering dimension.

CUPID [46] is an event recommendation platform for the Flemish cultural scene. In order to make personalized and accurate recommendations for specific users, an advanced collaborative filtering algorithm is used, in which user profiles are extended with probable future attendance. Then, the recommendation process is completed by applying content-based filters to the pool of possible items to be recommended, based on user profile, in order to exclude undesired items (events that are too far, not available etc.).

In [24] an event is recommended to a user if it is similar to the events that this user, or similar users, have liked in the past. Namely, a content-based system is extended with a collaborative feature by positioning a user within a network of related individuals and allowing her to explore new areas those users have appreciated. Three kinds of relations are defined: one between users, one between items, both expressing similarity, and one between users and items, expressing preference.

2.3.2 Online services for finding events

Recently, some *services* appeared on the Web that support users in finding events, even if they do not provide recommendation.

First of all, Facebook provides a service to create and manage event pages. Users can make their decision whether to participate after seeing: i) a description of the event, ii) which of their friends will attend the event, iii) the location of the event on an map. The events have not a reputation for the recommendation, nor thier content is formally specified, apart from a few general categories. Thus the user model and the user interests are not taken into account.

Other services are the ticketing website *SeatGeek* and *Lanyrd*.

SeatGeek[10] launched *Columbus*, an event recommender that exploits collaborative filtering to match users with events they might like; while *Lanyrd*[11] is a directory of conferences, events and speakers which relies on Twitter's social relationships. Visitors can see events their friends are attending or speaking at, submit new events, add talks that they have given and build up their speaker profiles.

[10] SeatGeek `http://seatgeek.com/columbus/`
[11] Lanyrd `http://lanyrd.com/`

3

Facts as structured information

The first part of our work has been to introduce a new kind of user-generated content in a social context: the **story fragments** or "facts".

The story fragments are simple structured sentences that users can use to communicate with each other to share their experience in a social and mobile context. They are structured and linked with semantic data (for example with an ontology), so that they are machine processable and it is possible to calculate their similarity according to our semantic similarity measure (see Section 4).

Our goal is to use them as captions or "advanced tags" for other types of content (text, pictures or other media) in a social network in order to aggregate and suggest them to users.

In section 6.2 we will discuss an iPhone-based application, *Telleat*, that allows user in mobility to provide the system with a fact concerning a content of the social network.

3.1 Facts as story fragments in the literature

The *narrative fiction* applied to multimedia documents has already been tackled by Zarri [62], who proposes NKRL ("Narrative Knowledge Representation Language"). NKRL is aimed at describing the *meaning* of complex multimedia narrative contents. He introduces the notion of "elementary event" such as a *spatio-temporal instantiation*. This instantiation is a "predicative occurrence" of n-*ary structures* that he call "templates" and identifies as general categories of basic events (i.e. "thinking about something" or "moving to physical space"). Another article [37], that our work extends, is inspired from the Zarri's article and develops the idea to represent the predicate occurrences with n-*ary* relationships.

In [37], an "elementary event" is called *fact*, which we also call "story fragment". Facts are characterized by:

- a *predicate*, an predicate occurrence that defines the type of action represented by the fact;
- a set of *role fillers*: domain entities that play a role (a n-*ary* relationship) within the action, together with the type of role they play.

Examples of predicates are: `Drink`, `Walk`, `Listen`, whereas examples of domain entities are: `Yesterday`, `I`, `Wine`, `March 23rd`, `Beauty`.
The set of roles fillers are composed by following roles:

- sbj (subject): who or what carries out the action (*Mary* ate the apple.);
- obj (object): the thing(s) the action is carried out upon (Mary and Peter bought *the apple*.);
- whr (where): the place where the action takes places (Elizabeth play *at the park*.);
- whn (when): the time indication in which the action takes places (Mark will graduate *in May*.);
- mdl (modality): any indication of the modality that further specifies how the action is performed (Mary is cooking the cake *with oven*.);

- why (cause or goal): the reason of an action (Mary is cooking the cake *for sister's birthday.*);
- ctx (context): something related to the action that does not explicitly fit in any of the other roles (Maria's grandmother goes to the her doctor *with blood tests*).

In [37], the authors introduce two ontologies represented in the Semantic Web Recommendation OWL 2[1]: the *predicate ontology* to define and organize all predicate used in the facts and the *domain ontology* to describe elements of role fillers both in general classes (i.e "Wine") and very specific classes (i.e. "Bordeaux Grand Cru ACME 2001"). Each predicate class restricts the set of role fillers (i.e. the predicate "Go" do not admits obj as fillers) and the typology of entities in specific role (i.e "Drink" admits only "Liquids" in the obj role).

There are two types of predicate classes: the *abstract predicate classes*, that are non-lexicalized concepts and the *concrete predicate classes* that can be instantiated. Each abstract class contains a set of concrete predicates with a common meaning and common role constraints (i.e. abstract class "Drinking Concept" contains "Drink", "Swallow", "Imbibe", etc. and accepts only "Liquid" elements in the obj role). These classes depend on specific natural language considered and are not associated with a specific verb. Each concrete class can have subclasses as well, which further specify certain actions (i.e. "Eat" and its subclass "Devour').

Another interesting proposal is in [30], where narratives atomic events are linked together by using metadata and content descriptions. The authors present a case study where the Finnish national epic Kalevola are reformulated creating a narrative structure with "intelligent" linkings, an interactive graph. The user may browse the graph and the recommendation links change on the fly. The links are automatically created by SPARQL and semantic

[1]Semantic Recommendation OWL 2 `http://www.w3.org/TR/owl2-overview/`

rules and each recommendation is explained to user.

The predefined structure of the facts in [37] allows the system to create association between them, such as in Kalevola work [30]. The authors individuate three different links:

- **Fact/Role-filler Link** translates a subordinate clause of natural language in a connection between facts. In particular this link uses a fact as role filler of another fact.
- **Role-filler/Role-filler Link** allows to specify that an element in a role filler is the same entity as that in the role filler of another fact.
- **Fact/Fact Link** is more generic link between facts and it is similar to conjunction or juxtaposition (disjunction and adversative are not defined explicitly) of natural sentences.

We start from these ideas to extend the concept of *fact* in a social context.

3.2 Semantic organization of domain items and predicates

As explained in the previous section, also in our work facts are represented in a structured way, corresponding to basic sentences in a natural language. The elements, that are part of a fact, are defined in a system ontology, in the *predicate* or in the *domain* ontology.

Starting from the facts contributed by the users of the system, we build a *facts repository* and a *domain ontology*, represented in the Semantic Web recommendation OWL 2, that contains:

- the *domain ontology* that defines and organizes the domain entities that can be used as role fillers,

- the *predicate ontology* that defines and organizes the predicates used to describe the actions performed in the facts.

Only the elements present in the ontology may be used in the construction of a fact and chose a verb it is possible to identify the set of role fillers supported by the predicate. Thus, we introduce *basic blocks* in order to make the language suitable for inputting facts without a keyboard, but rather using an interface designed for mobile devices (see Section 6.2). These basic blocks correspond to the role fillers (e.g. sbj obj or whr) and can be seen as an enriched form of the *tags* that are widely used in Web 2.0 to label an item, e.g., a blog post or a photo, with words the item is related to. In the tags the type of relation is not specified, even though the tag is typically used to express what the photo or post is about. In our work, facts describe *explicitly* which relation holds between, e.g., users, (food) items and other domain entities.

The *predicate ontology* contains both the abstract and the concrete predicate classes and is organized as follows:

- there is a first level where a main class, called `Actions`, defines all properties of a predicate corresponding to role fillers (`hasSubject` for sbj or `hasPlace` for whr) and their restrictions (in whr block there is only elements of geographical ontology).

- in the second level there are some abstract classes that aggregate the predicates of third level in according to their meaning (predicate of movement, of thinking, ect.)

- the concrete predicates are subclasses of `abstract classes`. They have a specific name and impose further restrictions on the roles (e.g., `Ingest`, `Eat`, `Drink`, `Taste` must have as object some `Food` or in the case of `Ingest` also a `Drink`). In some case a role is not admissible (e.g., `Swim` has no object) and it is possible to specify an empty range for the property `hasObject`. In other case the role can enrich from multiple fillers, that can be fillers of subproperties. For example the `whr` role has two fillers, the origin and the destination of a movement. So we can have the `hasPlaceOfOrigin` and the `hasPlaceOfDestination` in the same fact (e.g. "Mary moved from Turin to Milan yesterday").

A **fact** is an individual in the OWL 2 ontology which is asserted to be an instance of the subclasses of third level. The fact has role fillers in the domain ontology, respecting the type and number restrictions imposed in the action ontology.

As an example, consider the individual `f128` of subclass `Go` such that:

- `f128` is an instance of `Go`;
- `f128` has `Elizabeth` (an instance of `Woman`) as subject (role `sbj`);
- `f128` has `Mary's birthday` as place (role `whr`);
- `f123` has `yesterday` as time (role `whn`);

and the other roles have no fillers. `f128` would be expressed in a natural language as "Elizabeth went to the Mary's birthday yesterday".

3.3 Pertinence

The pertinence module computes the pertinence between facts according to the measure presented in [37].

In calculating pertinence between two facts, the following is taken into account:

- different facts use different predicates taken from the predicate ontology to express their content;
- each fact has only one predicate since every fact can be decomposed into more facts with only one predicate;

- each predicate has a set of associated values for different role fillers;
- each role might have 0 or more role fillers;
- the semantic similarity between two predicates or two role fillers is calculated using a measure presented in Chapter 4;
- in the case of two role fillers defined as two different users, the semantic similarity between them is calculated using the distance between users on the users graph of the social network;
- co-location estimates the possibility for the actors to meet while performing the described actions. To calculate co-location it is possible to use Google Maps API Web Services[2] that measure the distance between two spaces using their latitude and longitude.

Summing up, given two facts f and g, the *pertinence* of the fact g for the fact f is given by:

$$\textsc{pert}(f, g) = \alpha_0 \text{SP} + \Sigma_{i=1}^{m} \alpha_i \text{SRF}_i + \beta \text{COLOC} \tag{3.1}$$

where $\textsc{sp}$ is *semantic similarity of predicates*, $\textsc{srf}_i, i = 1, \ldots, m$ are *semantic similarities of role fillers* (m is the number of role fillers for f), $\textsc{coloc}$ is the *colocation* and $\alpha_0, \ldots \alpha_m, \beta \in \mathbb{R}$ are weights.

In the special case where similarity is being calculated for two users, we use their social network relationship instead of semantic similarity. Also, the pertinence calculation can be enhanced by introducing weights for certain roles depending on the verb.

[2]Google Maps API Web Services
https://developers.google.com/maps/documentation/webservices/?hl=it

3.4 Linking facts and user content

Pertinence helps the system to search for facts that are pertinent to those that a user has created or browsed. Two other possible uses are:

- to allow the user to create custom content to be added to domain,
- to use the idea of linking in [37] to make interesting stories.

Both uses are more interesting, but there are some problems. First, the custom content is not ontological concept and the system can not calculate the pertinence in the same way. Secondly, the linking between two facts can be done manually by a user or can be suggested by the system to the user by the pertinence.

3.4.1 Custom content

In the case that the desired content are not present in the system ontologies it is important to allow the user to define it. A custom content is characterized by a identifier number of the content and a URI of the user that create it, the owner. Some additional informations (a name, a small description and a picture of content) add optional information. This information is useful to users of the social network to use custom content within their facts.

The custom contents are not instances of ontology classes and maintain their information in a knowledge base. They are only visible from the owner and from the owner's friendships and they have not restrictions respect to the role where they may appear because they have not a typology such as ontological concepts.

Also, the system can suppose a similarity between two facts with custom contents only if custom contents are the same (they have the same identifier number) in both facts.

An example of custom content is the "cake of Mary's grandmother". It can be used as `obj` of predicate `Eat`, such as in `f131` "Elizabeth ate the cake of

Mary's grandmother", but it will appear in the same role with the predicate `Drink` because it has not a class restrictions.

3.4.2 Linking facts

The previously fact `f128` "Elizabeth went to the Mary's birthday yesterday" can be linked with another sentence `f129` "Elizabeth has tasted a Cabernet 2006" or also `f130` "Elizabeth is Mary's cousin". In the first case, the second individual fact `f129` is a coordinate sentence, structured in this way:

- `f129` is an instance of `Taste`;
- `f129` has `Elizabeth` (an instance of `Woman`) as subject (role sbj);
- `f129` has `Cabernet 2006` as object (role obj);

while the latter is a subordinate clause ("Elizabeth, who is a Mary's cousin, went to the Mary's birthday yesterday"):

- `f130` is an instance of `Be`;
- `f130` has `Elizabeth` (an instance of `Woman`) as subject (role sbj);
- `f130` has `Mary's cousin` as object (role obj);

The idea is to store in a knowledge base the typology of linking between two facts and use them to suggest facts with a interesting connection. These linkings are characterized by:

- the *id* of the primary fact (e.g `f128`)
- the *id* of the secondary fact, that is connected to the primary fact (e.g `f129` or `f130`)
- the typology of linking, that defines the secondary fact as "coordinate" or "subordinate" sentence

- the definition of typology, in the case of coordinate sentence we have the used conjunction (“and”, “but”, “or”), while in the case of subordinate sentence we have the associated role (sbj, whr, whn).

3.5 Conclusions

In this chapter, we have described *fact* as simple structured sentences, composed by a predicate and some role fillers. Role fillers are ontological entities that enrich the “action” in its main roles (*i.e.* subject or space description). We hypothesize a system that can guide the user in the creation of facts (see the prototype in Section 6.2), suggesting the ontological elements of the system (predicates and role fillers) that will form the user facts, or a system that is able to create automatically facts by analyzing the user actions of the social network (*i.e.* `f133 Ten of your friends went to the concert`).

In this way, the system can calculate a correlation between two different facts with the goal to aggregate them according to their pertinence.

Moreover, we propose to use facts as “advanced tags” of complex content (*i.e.* videos and images, such as in the *Wanteat Video* prototype in Section 6.1) for describing what they represent semantically. In this way the system can aggregate these particular contents using facts and suggest them to the users according their interest.

Finally, we have highlighted some possible ways of linking facts, toward the long-time goal of aggregating them in clusters according to narrative principles.

4

Semantic Similarity in Heterogeneous Ontologies

s In the previously Chapter 3, we introduced a new kind of user-generated content, the "facts", and we have computed their pertinence using a formula (see Formula 3.1) that compares predicates and all ontological entities in the roles.

For comparing an ontological element with another we have used our semantic similarity measure [3]. It was built on the principles of topological similarity approaches (see Subsection 2.2.1) and improves the semantic similarity measure between two entities.

4.1 Distance based similarity measures

There are two main strategies. The first consists in using of *entropy* and the content information (the *node-base strategy*), as in [49]; the second kind (the *edge-based strategy*) is to use the ontology graph structure, more precisely, using directly the *distance between nodes*, as introduced in [47]. Several measures of semantic similarity are based on such a distance; for a discussion and comparison with entropy-based strategy see [17].

We also recall the existence of a third approach that combines the *node-base strategy* with *edge-base strategy*, which we will call the *hybrid strategy*.

The node-base strategy

The *node-base strategy* is also called *information content* [48] and defines the semantic similarity as the measure of shared information in common between two concepts.

Resnik [49] presents a *node-base strategy* where the measure of semantic similarity is based on information content in an IS-A taxonomy. The information content of a class in a taxonomy is given by the negative logarithm of the probability of occurrence of the class in a text corpus. This means that the more abstract classes provide less information content, as opposed to more concrete and detailed classes lower down in the hierarchy. Concept probabilities are computed as relative frequencies (each noun in the text corpus is counted as an occurrence of each class that contains it). The closest class that subsumes both compared concepts, called a *most informative subsumer*, provides the shared information for both, and gives the measure of their similarity:

$$\textsc{sim}(a, b) = max_{c \in S(a,b)}[-\log p(C)] \tag{4.1}$$

where $p(c)$ is the probability of encountering an instance of concept c, and $S(a, b)$ is the set of all concepts that subsume a and b. Resnik argues that his approach shows better performance results than edge-counting approach, using human similarity judgements as a benchmark.

A very appealing approach to measuring semantic similarity is given by Jiang and Conrath [34] where edge counting approach is improved with information content one. The strength of a link connecting a child to its parent is the difference between information content values of the parent and the child. The weight of a link, in addition to link strength, takes into account local and average densities, depth of the parent node in the hierarchy and link type. Then the distance between two nodes is calculated as the shortest path linking the two nodes, using weighted links to traverse the paths. Experimental results show that this combined approach performs better than information content approach.

Lin [38] proposes an information-theoretic definition of similarity derived from a set of assumptions about similarity. It is measured as the ratio between the amount of information that two concepts have in common and the amount of information needed to fully describe them, thus taking into account similarities, as well as differences, between compared terms. His similarity measure is not tied to a particular knowledge representation and is applicable to any application with a probabilistic model. This allows using his measure in the applications in which similarity measure could not be introduced before.

The edge-base strategy

The *edge-base strategy*, or *conceptual distance approach*, derives from work by Rada [47] who formalized two important points to design his semantic distance measure:

1. the behavior of conceptual distance resembles that of a metric with its main properties
    - $f(x, x) = 0$ zero property,
    - $f(x, y) = f(y, x)$ symmetric property,
    - $f(x, y) \geq 0$ positive property,
    - $f(x, y) + f(y, z) \geq f(x, z))$ triangular property
2. the semantic distance is often directly proportional to the number of edges.

The simplest form to calculate the distance between two nodes A and B is the shortest path from A to B (*i.e.* the minimum number of edges that separate two nodes), but in an ontological space the distance between two adjacent nodes are not equal. So the edge that separates two adjacent nodes is normally labelled with a numerical value, that defines the weight of edge. It is therefore necessary to reformulate the semantic distance as the sum of edge weights along the shortest path from one node to the other. This assumes that all edges are weighted, in some case by an automatic algorithm, if necessary. The automatic method that defines the weight of edges has to consider structural characteristics of taxonomy: local density (number of child links from a node), depth of a node in the hierarchy, type of link and the relationship strength of a link. In [54] and in [51] the authors take in account these factors for calculating accurate weights.

Some recent work has further refined the edge-base measures, especially in bioinformatics. For example, the IntelliGO [13] algorithm calculates the semantic similarity between two genes, described in the Gene Ontology (GO), using a novel vector space model. The genes are defined by a vector-based representation of their annotations and the weights are used to evidence codes to be checked.

Our semantic similarity measure

Adopting the *node-base approach* requires a reasonably complete ontology, which is something we do not want to assume. Also, in the context of Social Web, the ontology of the domain is bound to grow depending on the social network usage, and it is fairly difficult to make any assumption about the completeness or even the homogeneity of the semantic information.

Due to the limitations of the ontologies we need to work with, we focused on this second type of measure to define a new notion of *semantic similarity*. In our case it is difficult to obtain the frequencies of concepts in the taxonomy (and consequently their probabilities) or have them provided a priori by domain experts. Hence, we take a closer look at two distance based similarity measures that could be suitable to be used in our setting.

- **Leacock and Chodorow** [35] use a similarity measure for word sense disambiguation in a local context classifier. The most similar nouns from the training set are substituted for the ambiguous ones in testing. The authors use the normalized path length in WordNet [31] between all the senses of the concepts being compared. The path length is measured in nodes rather than links, and the semantic similarity is computed as

$$\textsc{sim}_{\mathrm{LC}}(a,b) = -\log\left(\frac{Np}{2 \times \mathrm{MAX}}\right) \tag{4.2}$$

where Np is the number of nodes in path p from a to b and MAX is the maximum depth of the taxonomy. The length of the path between two same words (i.e. between members of the same synset) is 1.

Modifying slightly this measure we obtain:

$$\textsc{sim}_{\mathrm{LCD}}(a,b) = -\log\left(\frac{\textsc{dist}(a,b)}{2 \times \mathrm{MAX}}\right) \tag{4.3}$$

where $\textsc{dist}(a,b)$ is the distance from a to b.

The disadvantage of this similarity measure is that many pairs of non-similar words are estimated as similar, due to the equal edge lengths in their hierarchy.

- **Wu and Palmer's** [61] similarity measure accounts for the depths of the given words in the taxonomy and of their common subsumer, which characterizes their commonalities. Their measure is based on number of nodes on the paths between the compared nodes. The conceptual similarity between two nodes a and b, with the first subsuming node c, is computed as:

$$\textsc{sim}_{\mathrm{WP}}(a,b) = \frac{2N_c}{N_{(a)} + N_{(b)}} \tag{4.4}$$

where N_n is the number of nodes on the path from the root to the node n.

This measure can be slightly modified, in order to account for edge distances:

$$\textsc{sim}_{\mathrm{WP_D}}(a,b) = \frac{2\textsc{dist}(c)}{\textsc{dist}(a) + \textsc{dist}(b)} \tag{4.5}$$

where $\textsc{dist}(n)$ is the depth of n, i.e., its distance from the root.

4.2 Conceptual specificity

In the following discussion, we consider an ontology as a rooted, directed acyclic graph of inverse IS-A (subclass-of or instance-of) relations, i.e., the arc is directed from class to subclass or instance.

In general, the specificity of a concept is associated with the depth of the corresponding node in the ontology, while the distance between two concepts is associated with the length of the shortest path between the two nodes. However, due to the lack of completeness and homogeneity of the ontology we use, there are two issues that should be taken into account:

- The domain ontology is obtained by putting together several sub-ontologies detailing different aspects of the social network domain. In the gastronomic domain for example, we put together an ontology of wines, an ontology of places where wines are sold or tasted, and an ontology of producers. This is generally done by adding some fairly abstract concepts to the hierarchy that have no practical significance with respect to the domain itself - the best example is the `Thing` node that is usually the ontology root.

- Due to the lack of homogeneity among the different sub-ontologies, it can easily happen that concepts that have the same depth in the graph are perceived by users as having different conceptual specificity. Moreover, the perceived specificity may vary depending on the user context. For example, if the context is a wine fair, the `Wine` concept is not specific at all, since everything at the fair has probably something to do with wine. On the other hand, if the context is a Farmers' Market, then `Wine` becomes more specific, being one among many other things that can be found on the market stalls.

Given the above considerations, we propose to distinguish two types of nodes in the ontology:

- *Ground nodes*: representing the notions perceived as practically relevant in the domain. These nodes should have a finite specificity value, possibly depending on the context. The nodes with the same specificity value should be perceived by users as similarly relevant from an ontological point of view in the considered context.[1] Moreover, specificity should be monotonic with respect to depth.

- *Sky nodes*: representing abstract notions that are not considered relevant enough. These nodes should have a specificity value equal to $-\infty$.

In order to partition the ontology into the *Sky* and *Ground* sets of nodes, we require that a domain expert pre-selects a set S of *surface* nodes (highlighted in Figure 4.1) which is the basis for defining *Ground* nodes and their depth. The nodes in S represent the first domain-relevant concepts one encounters while traversing the ontology.

Definition 1 *Given the ontology $\mathcal{O}$ as a rooted, directed acyclic graph $\langle N, E\rangle$, and given a set S of surface nodes, the set of* Ground nodes *in $\mathcal{O}$ is the set*

$$Ground = \textsc{reach}(S)$$

of nodes that are reachable from nodes in S.

[1] Ontologically relevant means that it offers a useful distinction in relation to a theme. For example, even if a person does not consider *genre* an important factor in judging a book, he or she cannot dismiss *genre* as an irrelevant concept in the litterature domain, if only to be able to say that "genre does not matter".

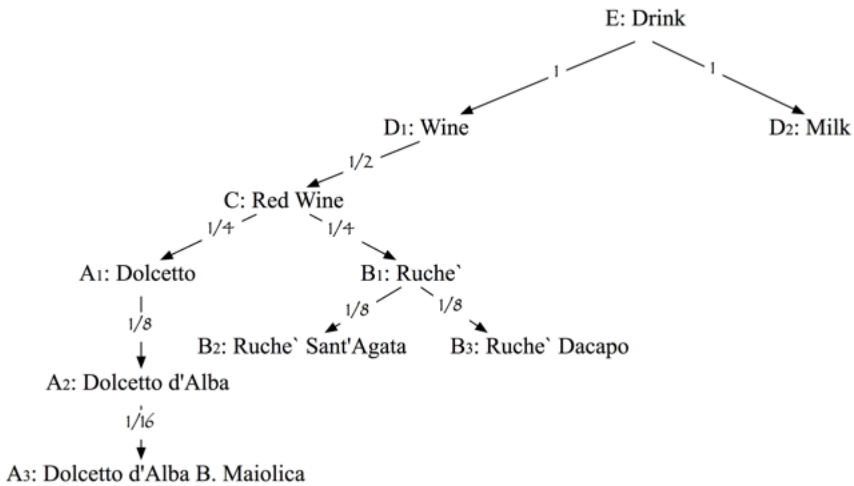

Figure 4.1: Weighted connection path between nodes

A finite specificity value is assigned directly to surface nodes, with 0 as a default value. As explained above, such an assignment may be context dependent: in the example, `Wine` could have specificity value 0 in general, but a negative specificity value in the context of a wine fair.

In defining the specificity for a ground node n, we consider the paths from surface nodes to n. We consider that each step in such a path adds some specificity to a concept corresponding to node n with respect to its ancestors. Since we do not want to burden the modeler with measuring such additional specificity for each step, nor we expect to have, e.g., statistical information on the frequency of occurrence of the concepts, we consider each step as a unit of "additional specificity".

For example, suppose that there are two paths of different length from surface nodes of specificity 0 to node n: the shorter one, covers, of course, the same specificity distance in fewer steps. However, in our interpretation, this means that some of its paths are longer than the minimal unit, and then we use the length of the longer path. If a surface node s has a non-zero specificity, we have to add it to the length of the path from s to n. We summarize the definition of specificity as follows:

Definition 2 *Given the ontology $\mathcal{O}$ as a rooted, directed acyclic graph $\langle N, E\rangle$, a set of surface nodes S, and a function $\text{SPEC}_S : S \to \mathbb{N}$ providing the specificity of surface nodes, the* conceptual specificity *value of a node $n \in N$ is defined as follows:*

- *if $n \in S$, then $\text{SPEC}(n) = \text{SPEC}_S(n)$.*
- *if $n \in Sky$, then $\text{SPEC}(n) = -\infty$.*
- *if $n \in Ground \setminus S$, then*

  $\text{SPEC}(n) = max\{\text{SPEC}(s) + num(p) \mid s \xrightarrow{p} n, s \in S\}$

where $s \xrightarrow{p} n$ means that p is a path from s to n and $num(p)$ is the number of edges traversed on the path p from s to n (including the node n).

4.3 Conceptual distance

The semantic distance between two ontological concepts is often based on their shortest connecting path (in terms of number of edges) in the ontology graph. However, in our case, such a distance also depends on the specificity of the traversed nodes, since we want to account for the fact that two specific nodes further down in the ontology graph are more similar than two more general nodes higher up in the hierarchy.

A very simple example of this is the following: if we consider the concept `Drink` and its two child nodes `Wine` and `Milk`, we have that the shortest path between `Wine` and `Milk` has the length 2. If we consider the `Cabernet` concept (descendant of `Wine`) and its child nodes `Cabernet Wonder 2004` and `Cabernet Merveille 2007`, we have that again the shortest path between the two children has the length 2. However, since `Cabernet Wonder 2004` and `Cabernet Merveille 2007` are children of a more specific node, their conceptual distance is perceived as smaller than the distance between `Wine` and `Milk`.

This problem is due to the fact that in the standard edge counting approaches to similarity, all the edges are usually considered of the same length, thus representing uniform distances between the nodes. In the taxonomies with certain very dense sub-taxonomies, like ours, this is a problem, since it does not reflect the fact that the descendants lower down in the hierarchy are considered conceptually closer than the ones higher up,

For this reason we define an edge length function that is parameterized with respect to the specificity of its source node (i.e., its depth in the concept hierarchy), ensuring that edge lengths decrease exponentially when going deeper underground (while being infinite for sky nodes means that we are not even interested in such distances). In particular:

Definition 3 *Given an edge $e : s \to t$, the edge length is given by*

$$\text{LEN}(e) = k^{-\text{SPEC}(s)}.$$

where $k \in \mathbb{N}$ is a constant ($k \geq 2$).

We then compute the conceptual distance between any two nodes as follows:

Definition 4 *Given two nodes $n_1, n_2 \in \mathcal{O}$, their* conceptual distance *is the length of the shortest path*[2] *connecting them via an ancestor node:*

$$\begin{aligned}\overline{\text{DIST}}(n_1, n_2) = \min\{&\text{LEN}(p_1) + \text{LEN}(p_2) \mid \\ &\exists g \textit{ such that } g \xrightarrow{p_1} n_1,\ g \xrightarrow{p_2} n_2\}.\end{aligned}$$

Notice that whenever a path from n_1 to n_2 crosses a *Sky* node, then $\overline{\text{DIST}}(n_1, n_2) = \infty$. This corresponds to the case when n_1 and n_2 belong to different sub-ontologies, and are therefore concepts of a different sort (e.g. a

[2] considering the graph undirected.

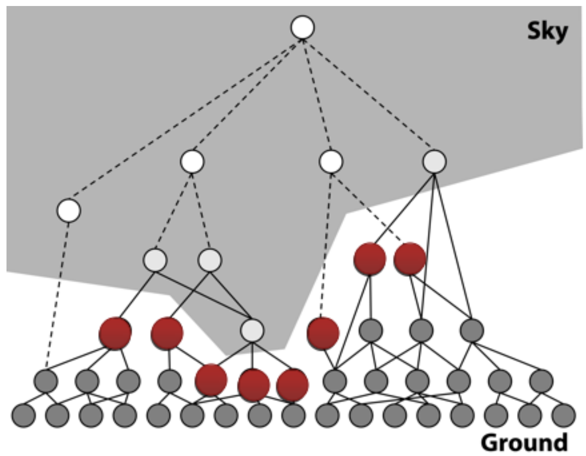

Figure 4.2: The sky nodes of an ontology

`Wine` and a `Restaurant`). We are not claiming that there is no relationship between these two concepts, but that they are ontologically distant.

4.4 Semantic similarity revisited

In this section, we propose a new measure of similarity between ontological concepts, based on the conceptual specificity (introduced in Section 4.2) and the conceptual distance (defined in Section 4.3). The new similarity measure is obtained by adapting Leacock and Chodorow's definition of similarity [35], given in Section 4.1, with the modified notion of distance in Section 4.3.

Definition 5

Given two domain entities n_1 and n_2 (nodes in the domain ontology), we define their similarity as:

$$\overline{\mathrm{SIM}}_{LCd}(n_1, n_2) = -\log\left(\frac{\overline{\mathrm{DIST}}(n_1, n_2)}{2 \times \mathrm{MAX}}\right) \tag{4.6}$$

where MAX *is the maximum length of a path from a surface node to a* *terminal node in the ontology. When* $n_1 = n_2$ *then* $\overline{\text{DIST}}(n_1, n_2) = 0$ *and the nodes* n_1 *and* n_2 *become infinitely close.*

As we would see in Section 4.5, this new measure of similarity brings some improvements over the original measure of Leacock and Chodorow [35]. It resolves the problem of the same distance between the nodes further down and the nodes further up in the hierarchy: the more specific concepts become less distant, and consequently more similar, than the more general concepts.

4.5 Evaluation

4.5.1 Goals of the experiment

This section describes a simple experiment we conducted in order to evaluate our proposal of measuring the semantic similarity between ontological concepts. In particular, our main goal was to compare human judgements with the results computed by the system. Moreover, we wanted to analyze the performance of Leacock and Chodorow's similarity measure [35] and Wu and Palmer's similarity measure [61] using a standard distance between the nodes in the ontology and using our modified conceptual distance.

4.5.2 Description of the experiment

A total of twenty persons were chosen among the contacts and colleagues of the authors, according to an availability sampling strategy.[3] Half of the subjects were used as a reference group, the other half was used as a control group to measure the correlation of judgements of different human subjects, as in [49]. All subjects were native Italian speakers. They were asked to rate the similarity of 28 pairs of Italian verbs from the system ontology, assigning

[3]Even though non-random samples are not statistically representative, they are often used in psychology research and usability testing, during early evaluation phases.

the values on the 5-point scale from 0 to 4 (0 meaning not similar at all, 4 meaning very similar) to the verb-pairs.

The pairs range from the ones expected to be classified as highly similar by human subjects (and measured as relatively highly similar), such as (Italian for) `Talk-Speak`, to the other extreme of pairs classified as almost not similar, e.g. `Die-Chat`.

The ordering of pairs was random for each subject.

4.5.3 Results and discussion

Table 4.1 reports the correlation between the reference group human judgements and the control group human judgements, as well as the correlation between the reference group human judgements with the following similarity measures:

- $\textsc{sim}_{\mathrm{WP_D}}$: Wu and Palmer's measure [61] with the standard (uniform) edge distance;
- $\overline{\textsc{sim}}_{\mathrm{WP_D}}$: Wu and Palmer's measure with our conceptual (exponentially decreasing) edge distance;
- $\textsc{sim}_{\mathrm{LC_D}}$: Leacock and Chodorow's measure [35] with standard (uniform) edge distance;
- $\overline{\textsc{sim}}_{\mathrm{LC_D}}$: Leacock and Chodorow's measure with our conceptual (exponentially decreasing) edge distance.

The correlation for the replication experiment with the human subjects (i.e. the comparison of the two groups of human subjects) is similar to the one reported in [49]. The best results with respect to similarity measures is obtained using our modification of Leacock and Chodorow's original measure, with exponentially decreasing edge distances.

Even though in our experiment the original Wu and Palmer's measure provides better results than the original Leacock and Chodorow's measure,

Similarity method	Correlation
Control Group	0.9060
$\textsc{sim}_{\mathrm{WP_D}}$	0.7942
$\overline{\textsc{sim}}_{\mathrm{WP_D}}$	0.6371
$\textsc{sim}_{\mathrm{LC_D}}$	0.7637
$\overline{\textsc{sim}}_{\mathrm{LC_D}}$	0.8558

Table 4.1: Correlation results for semantic similarity

our modified notion of distance does not fit well with Wu and Palmer's measure (as long as we are interested in a linear correlation with human judgements). In fact, in $\textsc{sim}_{\mathrm{WP_D}}(a, b)$, where the denominator is equivalent to $\textsc{dist}(a, c) + \textsc{dist}(b, c) + 2\textsc{dist}(c)$, replacing $\textsc{dist}$ with $\overline{\textsc{dist}}$ assigns a higher weight to the upper edges, and then the term $2\overline{\textsc{dist}}(c)$ dominates the term $\overline{\textsc{dist}}(a, c) + \overline{\textsc{dist}}(b, c)$. As a result, while $\textsc{sim}_{\mathrm{WP_D}}(a, b)$ tends to use most of the range $[0, 1]$, when using the modified distance, the similarity values obtained with the measure $\overline{\textsc{sim}}_{\mathrm{WP_D}}(a, b)$ are squeezed towards 1.

4.6 Implicit relevant classes

The definitions for specificity and distance given in the previous sections are based on the structure of the ontology as a graph. The underlying idea is that the ontology explicitly includes all and only the concepts that are considered "important", i.e. relevant for such structural measures. However, it may be the case that several orthogonal structures are important, e.g., in the case of cheese:

- based on the type of milk: cow's milk cheese, goat cheese, sheep cheese;
- based on freshness, from fresh cheese to aged cheese;
- based on texture, from soft cheese to hard cheese.

At the same time, some other features (e.g. shape) will be present in the ontology, as a part of knowledge about types of cheese, but would not be considered relevant for classification of types of cheese (cylindric cheeses etc).

In this case, we can assume that most intersections of relevant classes are relevant, without expecting the modeler to explicitly introduce all such intersections (fresh goat cheese, semi-hard cow's milk cheese, etc)[4].

Suppose that C_1 and C_2 are two types of a `Fresh Goat Cheese`, as in Figure 4.3. Both the specificities of C_1 and C_2 and their conceptual distance from one another, as defined previously, are affected by the presence or absence of the class `Fresh Goat Cheese`. The presence of the class `Fresh`

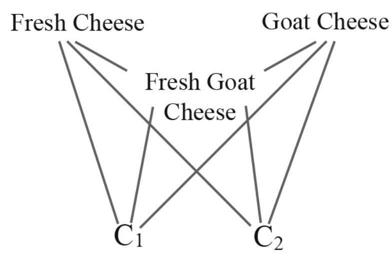

Figure 4.3: An example of intersection

`Goat Cheese` would add specificity to C_1 and C_2 and reduce their distance because the path through `Fresh Goat Cheese` would be shorter (significantly shorter, given the exponential decrease of distance) than the paths through the less specific `Fresh Cheese` and `Goat Cheese`.

Let us discuss in detail how specificity could be redefined for ground nodes, if all such intersections were present. Consider a ground node n and let $P(n) = \{p_1, \ldots, p_m\}$ be the set of all its parents having no descendants in $P(n)$ (see Figure 4.4).

Without the intersection classes, the length of the longest path from surface nodes to n would be obtained by adding 1 to the specificity of the parent with the highest specificity. The presence of all the intersections would increase, however, from 1 to m the number of edges from any p_i to n. Therefore,

[4] Some of these intersections, e.g. hard fresh cheese, may be useless in the sense that they fail to have subclasses and instances in the domain; however, this does not affect the modified definitions below, since we only consider the intersections that are superclasses of some existing class.

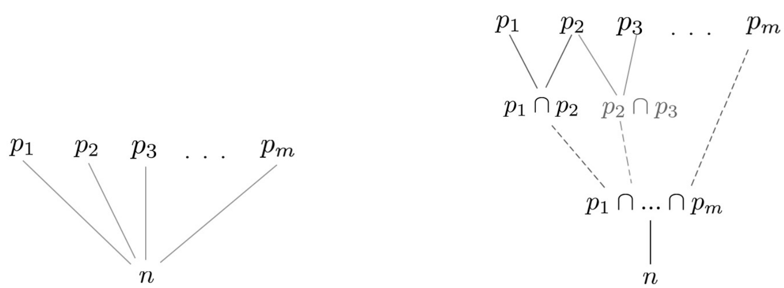

Figure 4.4: Modified specificity

Definition 2 of conceptual specificity is modified by defining the specificity for ground nodes inductively using the specificity of surface nodes as follows:

Definition 6 *For a non-surface ground node n, let*
$P(n) = \{p_1, \ldots, p_m\} =$
$\{p \mid p$ *is a parent node of n, p has no subclass in P(n)*$\}$.
Then

$$\text{SPEC}(n) = max\{\text{SPEC}(p_i) \mid i = 1, \ldots, m\} + m.$$

In order to modify the definition of conceptual distance between two nodes n_1 and n_2, while simulating the presence of all intersection classes, we have to consider the minimal upper bounds $\{g_1, \ldots, g_m\}$ of the nodes n_1 and n_2 (`Fresh Cheese` and `Goat Cheese` in in the above example). In general, such upper bounds might have different depths and the paths to n_1 and n_2 might involve several edges. With the addition of all intersection classes, the shortest path (in the sense of Definition 4) would go through the single minimal upper bound

$$\hat{g} = g_1 \bigcap \ldots \bigcap g_m.$$

The length of the paths $\hat{g} \xrightarrow{p_1} n_1$, $\hat{g} \xrightarrow{p_2} n_2$, whose sum

$$\text{LEN}(p_1) + \text{LEN}(p_2) = \overline{\text{DIST}}(n_1, n_2)$$

can be computed from the specificity of $\hat{g}$ and the number of edges in the paths.

The specificity of $\hat{g}$ would be the one of the most specific g_i, increased by $m - 1$ (similarly to Definition 6).

The number of edges in p_1 (and similarly for p_2) is computed as follows. We consider the set $I = \{i_1, \ldots, i_h\}$ of the intermediate nodes on the paths from any g_j $(j = 1, \ldots, m)$ to n_1, excluding those that are defined as intersections of two ancestor nodes in the paths (to avoid counting them twice, since we are simulating the presence of intersections). Nodes that lie between $\hat{g}$ and n_1 are those that can be obtained by intersecting $\hat{g}$ in all possible ways with the nodes from I.

In particular, node n_1 can be reached from $\hat{g}$ traversing $h + 1$ edges, at any step intersecting with one of the h members of I, e.g. through the nodes $\hat{g} \cap i_1$, $\hat{g} \cap i_1 \cap i_2$, ..., $\hat{g} \cap i_1 \ldots \cap i_h$.

In this case we calculate $\textsc{len}(p_1)$ as:

$$\textsc{len}(p_1) = \sum_{j=0}^{h} k^{-\textsc{spec}(\hat{g})+j}$$

4.7 Conclusions

In this chapter, we studied a new semantic similarity measure to calculate the semantic distance between entities in an ontology. It is based on *conceptual specificity* and *conceptual distance* and the evaluation of our approach shows an improvement over other distance measures in the literature. Our goal was to overcome the problem of measuring semantic similarity incomplete ontologies, that however model concepts both specific and general.

We devised this measure as a part of the computation of pertinence. However, we believe it can be applied to a wider scope of situations.

For example, it has been proposed in [21] to exploit it within the context of user modeling and in particular to properly propagate the users' interest within an ontology. Users' interest may in fact reverberate on similar entities where similarity can be computed according to our measure.

5

Users and events

In this chapter, we tackle the problem of *selecting* and *ranking* real-life events proposed to people within a social networking service (*i.e.* we observe the decision process which regards attending events).

We started considering events after discovering how a strict measure of semantic similarity between two facts by the pertinence module (see Section 3.3) does not take into account an important aspect: the spatial-temporal context where the fact takes place. The *Telleat* evaluation (see Section 6.2.3) has highlighted how users are influenced by situation associated to facts. In many case, two facts are judged similar or pertinent by humans if they are contextualized in the same event.

We thus introduced in our semantic knowledge base a representation of "spatial temporal objects" as an entity with the following features:

(a) having a participatory aspect (*i.e.* an activity is offered and people can decide whether to join or not),

(b) requiring the physical presence of people in the location where the activity takes place,

(c) being offered within a given time interval and requiring a certain amount of time to be completed.

For example, Facebook[1] "events" can be regarded as activities according to this definition.

Introducing this type of objects in our knowledge base led us to investigate how events are different from other content types when it comes to suggesting and recommending things to users.

5.1 Relevant factors in event recommendation

The domain of "event recommender" is particularly complex since the users take into consideration a lot of factors in making their decisions: the content of the event (for example, a baroque music concert), the presence of friends, reputation or the popularity of the event, its distance from the user's current position, time constraints etc.

Traditionally, recommender systems do not take all these factors into account simultaneously. In particular, content-based recommenders [44] consider only the user's preference for the content, whereas collaborative filtering recommenders consider similarity of users' tastes or preferences. More recently, social recommender systems have started to use data regarding users social relationships in filtering relevant information to users. However, "to date, results show that incorporating social relationship data - beyond consumption profile similarity - is beneficial only in a very limited set of cases" (see [12]).

The starting idea of this study is that the "inconclusive results are, at least to some extent, due to an under-specification of the nature of the social relations". In other words, social relations are usually not taken properly into account in the recommendation process.

Following this assumption, our primary objective is to investigate how social, context and content factors impact users in making decisions. In particular, the main questions which inspired our research are:

[1] Facebook `http://www.facebook.com`

RQ1 *Is the basic information about an event, such as its type and topics, enough to provide satisfying recommendations to the users?*

RQ2 *Does additional information enhance recommender system accuracy?*

RQ3 *Which types of factors are most useful?*

RQ4 *Would letting the users explicitly voice their preferences regarding these additional factors bring some improvements in the recommendation? Or is there a better way to estimate the influence of these additional factors?*

RQ5 *Is there any dependence between factors?*

We chose certain additional factors which describe events, as significant examples of social, context, and content properties. In particular, the factors we consider are the following:

- *Content features*: the basic features which describe an event are its *type* (i.e. the category which the event belongs to) and its *themes* (i.e. the topics the event is related to), since they can reasonably represent its content [44];

- *Context features*: temporal and spatial properties are the most important features of context usually considered in recommender systems [11]. As spatial properties of the events, we consider in particular the *reachability* of the events, i.e. how feasible it is for the user to attend the event, taking into account user's position and propensity to move.

- *Social features*: the presence of friends and the overall community opinion on the event (e.g. expressed by means of overall ratings of the event). We decided to include social dimensions, since events attendance is strongly influenced by activities of friends and acquaintances.

We performed a user study in which we collected useful information about the behavior of the users when it comes to events and their attendance. The answers we obtained led us to conclude some interesting facts about what influences people decisions the most and what additional factors they take into account. Also, we discovered how all these factors can be combined in order to provide the users with optimal recommendation for their specific situations. The following give a summary of our findings:

RQ1 The basic content information about the event, *i.e.* its type and themes, does play a crucial role when deciding whether to attend the event or not.

RQ2 Users do take the additional information into account (distance of the event from the user, event rating and friends' participation) when making their decisions. Hence, including this additional information when recommending events does improve the performance.

RQ3 Type and themes are most useful and important factors but not equally, rather, themes carry more information and influence the users more. Other types of factors increase the accuracy of recommendation and are average rating, friend participation and reachability of event.

RQ4 The importance of factors explicitly declared by users does not correspond to their real rating and attendance behavior. Better results are obtained by assigning different importance weights to the additional factors.

RQ5 The combined influence of the basic content information about the event, with the three additional factors can significantly improve event recommendation. The best way to decide the importance of additional factors is to calculate their importance dynamically based on the results obtained for the importance of type and themes.

Then, we applied different weights, calculated in different modalities (with the ratio $1 : 2$ for type and themes, equals for other additional factors, explicitly declared by users, calculated by a collected experimental data or dynamically calculated based on users' interest) for all these factors in a content-based recommendation algorithm, in order to discover the impact on recommendation accuracy. Moreover, we can say that showing users additional factors change their decision about the event and increase the errors in the recommender system that does not take them into account.

We aim at applying our findings in the design of social recommender systems, for improving the prediction accuracy. Thus, we discuss implications for the design of recommender systems .

In this chapter, in order to test our hypothesis, our approach was to create a prototype of recommender system. In particular, our recommender is composed of a data model, that gives a description of the users and of the domain (Section 5.2.1) and a process for selection & ranking (Section 5.4) that considers such factors.

The process we describe can be adapted to several recommender systems and is flexible to user configuration.

5.2 Approach

In order to test our hypotheses, we developed a prototype of an content-based event recommender. In it, as in typical content-based recommenders, *ranking* is the core of the recommendation process, and it is based on *scores*. A score is assigned to a pair (u, o), where u is a user and o an object to recommend, which expresses an estimate of how much user u would be interested in o. If we denote by U the set of all the users of the recommender system and by O the set of all the objects available for recommendation, then the score can be represented as a function $\sigma : U \times O \to I_\sigma$, where I_σ is a closed interval over $\mathbb{R}$.

In the rest of this section we will clarify the assumptions we make on users and objects and which information we assume to know about them (Section 5.2.1). Moreover, we will describe our prototype's scoring function (Section 5.2.2)

5.2.1 Data models

The data model we adopted in our prototype encompasses a model of the event and a model of the user.

Spatial-Temporal Objects model

So far we have talked about recommending *events*. However, we technically refer to the objects we consider for recommendation as *spatial-temporal objects*, or *STOBs* for short. We aim at including in this definition any activity that:

- takes up an allotted time within a possibly wider temporal frame;
- takes place in a well-defined physical location;
- can be proposed to other people (possibly belonging to a restricted group) that can accept or decline the invitation.

While private parties or tennis lessons would not probably fit into the shared notion of "event", they can be definitely regarded as *STOBs*. Facebook notion of "event" is actually very similar to our *STOB* and we have to conclude that the name "event" is chosen for lack of a better option.

More formally, the properties we model for each *STOB* o are the following:

Contextual Properties include i) temporal properties, and ii) spatial properties. Temporal properties are the expected duration of the activity associated with o, denoted by dur_o, and the time frame in which the activity may occur, expressed as an interval $[start_o, end_o]$. Spatial Properties define the area loc_o where the activity takes place, described as a circle $(\hat{c}_o, \hat{r}_o)$ with center $\hat{c}_o = (\hat{x}_o, \hat{y}_o)$ and radius $\hat{r}_o$.

Content Properties are meant at capturing the nature of the event. We characterize with labels the *themes* of the activity that is taking place and the *type* of activities of the event. Examples of themes, *i.e.* for activities related to wine & food, are fish or cheese, while examples of type are dinner or tasting. We associate with each $STOB$ o a non-empty set THM_o of theme labels and a non-empty set TYP_o of type labels. We denote by $\overline{THM}$ and $\overline{TYP}$ the set of all available labels for themes and types, respectively, and we assume these two sets to be disjoint.

Social Properties capture the interest expressed by users on o. We consider two social properties: *ratings* and *participation*. *Ratings* can be expressed as a partial function $rating : U \times O \rightarrow I_\sigma$; the value $rating(u, o)$ expresses how user u finds o potentially interesting, without of course having a direct knowledge of o, since we assume it has not yet taken place.[2] For convenience, we also define the set $Raters(o)$ of those users who have provided a rating for o.Notice that we assume ratings fall in the same range of the final score provided by the system; regardless of the rating options presented to users, a normalization function can always guarantee this to be true. *Participation* is simply expressed by the subset $Part(o) \subseteq U$ of those users that have confirmed their participation to o.

[2]This distinguishes $STOBs$ from other objects that can be recommended, such as books or movies, where users can express their rating after having experienced the object. Recurring events may allow this type of a-posteriori ratings, which would then represent an additional factor in the recommendation.

User Model

The user model is meant at capturing those features of a user that we deem relevant to the recommendation process. We are not concerned here with how this information is determined or gathered: it may be partly explicitly provided by the users themselves, and partly inferred via some learning strategy from their interaction history.

We therefore assume that the user model can provide to the recommender system the following information:

Spatial Frame: includes a *user's position in space* and a *propensity to movement*. In general, the user's position may be either determined by geopositioning, or correspond to a generic "home " location, or be explicitly provided (consider for example the following situation: "Next week I will spend two nights in New York, can you recommend something to do over there?"). We represent the user's position in a similar way as we did for STOB locations, that is, as a circle $(\hat{c}_u, \hat{r}_u)$ with center $\hat{c}_u = (\hat{x}_u, \hat{y}_u)$ and radius $\hat{r}_u$. The propensity to movement is expressed as a distance measure mov_u and tells us the maximum distance the user would be willing to travel in order to participate to an activity (notice that this, also, can be local to the specific situation: in his everyday life a person may not want to cover long distances for an evening out, while on a holiday trip he could feel more explorative).

Temporal Frame: expresses the time window for which the recommendation should be provided, and we will represent it as a pair $(start_u, end_u)$. Again, these values may be determined in different ways. For example a user may have a default preference for receiving recommendations of *STOBs* occurring in the upcoming month, but he may also ask for recommendations specific to a given time/place of his choice.

Profile: describes the interests of a user toward the themes and types of STOBs, as defined above. For our purposes, it suffices to represent interests as a function $int : U \times (\overline{THM} \cup \overline{TYP}) \rightarrow I_\sigma$. Again, we assume for the sake of presentation clarity and without loss of generality, that the interest values fall in the same range of the final recommendation score provided by the system.

Social Network: represents the network of friendships among users. Although this information can be expressed at different levels of complexity, for the present work we are only concerned with the friendship relation, seen as a binary, non reflective, and possibly non-symmetrical relationship $F \subseteq U \times U$: if $(u, v) \in F$ then u has declared v to be his friend (and not necessarily vice-versa). Therefore, for a given user u, we can define $F_u \subseteq U$ as the set $\{v \mid (u, v) \in F\}$.

5.2.2 The Score Function

The score function we will use is the weighted sum of several scoring factors, where each scoring factor is in turn a function $f : U \times O \rightarrow I_\sigma$ which captures a particular aspect that users may consider in deciding whether they are interested in a $STOB$. Thus we have:

$$\sigma(u, o) = \sum_{i=1}^{n} \omega_i \cdot f_i(u, o) \quad \text{where} \quad \sum_{i=1}^{n} \omega_i = 1.$$

Let us first discuss which are the **scoring factors** we will consider in our approach and then evaluate.

Given a user u and a $STOB$ o, a scoring factor is a function $f : U \times O \rightarrow I_\sigma$ that captures a particular aspect that use u can be expected to consider when deciding whether he is interested in the activity offered by o. In setting up our scoring function, we have considered the following scoring factors:

Thematic Interest: $thi : U \times O \rightarrow I_\sigma$ is defined as the average interest expressed by u (through his user model) toward the themes of o:

$$thi(u, o) = \frac{\sum_{lab \in THM_o} int(u, lab)}{|THM_o|}.$$

Type Interest: $tyi : U \times O \to I_\sigma$ is analogous to thematic interest, but takes into considerations the $STOB$ type rather than its themes:

$$tyi(u, o) = \frac{\sum_{lab \in TYP_o} int(u, lab)}{|TYP_o|}.$$

Average Rating: $rat : O \to I_\sigma$ computes a weighted average rating, where the weight of a rating given by user v corresponds to the interest expressed by v towards the $STOB$ themes, as a measure of reliability. This scoring factor depends only on the $STOB$ and not on the user u whom we are computing the score for.

$$rat(o) = \frac{\sum_{v \in Raters(o)} rating(v, o) \cdot thi(v, o)}{\sum_{v \in Raters(o)} thi(v, o)}.$$

Reachability: $rch : U \times O \to I_\sigma$ aims at capturing the propensity of a user u to cover the existing distance between himself and the $STOB$. Given the upper and the lower bound of our scores, $ub(I_\sigma)$ and $lb(I_\sigma)$, we want $rch(u, o) = ub(I_\sigma)$ when the user's and $STOB$'s centers ($\hat{c}_u$ and $\hat{c}_o$) coincide, while $rch(u, o) = lb(I_\sigma)$ when the distance between the two areas is equal or greater than the user's *propensity to movement* ($dist(\hat{c}_u, \hat{c}_o) \geq \hat{r}_u + \hat{r}_o + mov_u$). This can be obtained by using the following piecewise linear function:

$$rch(u, o) = \max\left(lb(I_\sigma), \frac{lb(I_\sigma) - ub(I_\sigma)}{\hat{r}_u + \hat{r}_o + mov_u} \cdot dist(\hat{c}_u, \hat{c}_o) + ub(I_\sigma) \right).$$

Friend Participation: $frn : U \times O \to I_\sigma$ counts how many of a user's friends participate in the $STOB$, and scores maximum value for a participation of 10 people. Thus:

$$frn(u,o) = \min\left(ub(I_\sigma), \frac{ub(I_\sigma) - lb(I_\sigma)}{10} \cdot |Prtc(o) \cap F_u| + lb(I_\sigma)\right).$$

The global scoring function can thus be expressed as:

$$\sigma(u,o) = \omega_{thi} \cdot thi(u,o) + \omega_{tyi} \cdot tyi(u,o) + \omega_{rch} \cdot rch(u,o) + \omega_{rat} \cdot rat(o) + \omega_{frn} \cdot frn(u,o)$$
$$\text{where} \quad \omega_{thi} + \omega_{tyi} + \omega_{rch} + \omega_{rat} + \omega_{frn} = 1. \quad (5.1)$$

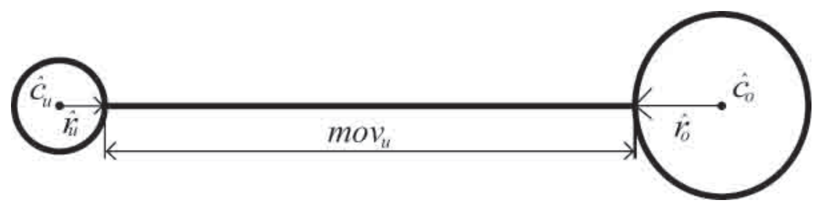

Figure 5.1: Reachability is minimal when the distance between user's area and event area is equal or greater than the user's propensity to movement mov_u

5.2.3 Weights calculation

In order to test the research questions stated in Section 5.1, we proceeded to compute a number of variants of the scoring function (5.1), each taking into account different combinations of scoring factors and different weights. Our goal was to compare the capability of these different functions to predict user's interest in an STOB, therefore determining the appropriateness of a given combination of weights.

We computed six variants of the scoring function. Two of them consider only thematic and type interest (*thi* and *tyi*):

- $\sigma^0_=$: equal weights for thematic and type interest;

- σ^0_{fine}: fine-tuned weights based on the collected data.

The remaining four variants consider also the three additional factors (*rch*, *rat*, *frn*), determining their weights in the following way:

- σ_{fine}: fine-tuned weights based on the collected data;
- σ_{dyn}: dynamic weights, computed as a function of the value of σ^0_{fine}, based on the collected data;
- $\sigma_{\cong}$: equal weights for all the additional factors;
- σ_{user}: weights provided by users according to self-observation.

In order to fine-tune the weights for those scoring functions which required it, we collected data from potential users, to be used as a test set. We published an online questionnaire where people could rate certain events, all related to wine and food, that we imagined would take place within the context of *Salone Internazionale del Gusto*[3] in Turin, Italy.

Data Collection

We created a fictional situation were the subjects were told they were about to participate to the *Salone Internazionale del Gusto*, and that they had booked an hotel in Turin, very close to the main site of the fair, for the duration of the event (5 days). This allowed us to set the *user position* as the same for all users (namely $\hat{c}_u = (0, 0)$ and $\hat{r}_u = 0.1km$), thus considering a relative system of coordinates centered in location of the fair.

We created 15 $STOB$ objects, whose relevant features are shown in table 5.1, which we divided into three groups called **A**, **B** and **C** (groups were formed with the intention to guarantee a certain variability of features among the $STOBs$ in each group). For each $STOB$ we recorded:

[3] Salone Internazionale del Gusto is a large, 5-day fair on the theme of sustainable & quality food taking place every two years in Turin, Italy, which organizes and promotes events such as dinners, tastings and debates, on the whole Italian region of Piemonte. `http://salonedelgustoterramadre.slowfood.com/`

- one or two themes and a type;
- the distance between the user center c_u and the $STOB$ center $\hat{c}_o$, and a fixed $STOB$ radius $\hat{r}_o = 0.1$;
- a made up "number of friends" that participate to the $STOB$;
- a made up average rating.

The 7 themes (wine, beer, cheese, cold cuts, fish, oil, coffee) were chosen among classical food-related topics while the 4 types (dinner, tasting, workshop, debate) correspond to real activities taking place during *Salone Internazionale del Gusto*.

(a)

(b)

Figure 5.2: Example of events showed to the users: (a) with information limited to themes and type and (b) with full information

Then, we asked 100 users, 18-70 years old, recruited according to an availability sampling strategy[4], to answer a questionnaire, providing the following information:

- A score 0-10 expressing their interest for each $STOB$, *having only the $STOB$ description along with its themes and type* (see Figure 5.2);

[4]Much research in social science is based on samples obtained through non-random selection, such as the availability sampling, i.e. a sampling of convenience, based on subjects available to the researcher, often used when the population source is not completely defined.

STOB	Group	Themes	Type	Distance [km]	Friends Nr.	Av. Rating
o_1	A	fish	workshop	0	0	8
o_2	B	coffee	workshop	0	3	3
o_3	C	wine	workshop	36	1	8
o_4	A	beer, cheese	workshop	4	0	9
o_5	B	beer	tasting	0	1	7
o_6	B	cheese, wine	tasting	6	1	6
o_7	C	cold cuts	tasting	0	0	6
o_8	B	beer	debate	6	2	8
o_9	C	wine	debate	0	0	7
o_{10}	C	cold cuts	debate	0	5	5
o_{11}	A	coffee	debate	22	4	8
o_{12}	B	fish, oil	dinner	6	0	9
o_{13}	A	fish, beer	dinner	77	b	4
o_{14}	C	cheese	dinner	3	3	6
o_{15}	A	cheese, cold cuts	dinner	0	4	4

Table 5.1: *STOBs* in the experimental set up

these will be referred to as the *initial* user scores, and $\tau_u^0(o_j)$ will denote the init score given by user u ($u = 1, \ldots, 100$) to *STOB* o_j, $j = 1, ..., 15$.

- For *STOBs* in group **A**, a score 0-10 expressing their interest *after being informed of the distance* between the STOB and them; these will be referred to as the D user scores, and denoted by $\tau_u^D(o)$.
- For *STOBs* in group **B**, a score 0-10 expressing their interest *after being informed of the average rating* for the STOB; these will be referred to as the R user scores, and denoted by $\tau_u^R(o)$.
- For *STOBs* in group **C**, a score 0-10 expressing their interest *after being informed of the number of friends participating* to the *STOB*; these will be referred to as the F user scores, and denoted by $\tau_u^F(o)$.

- a value 0-10 expressing their interest for each theme and type;
- their propensity to movement, as a distance expressed in kilometers.

We decided not to ask for D, R and F user scores for all the 15 $STOBs$ because we did not want interferences between the different types of information. Users may, for example, remember distances or ratings when providing the F score, and unconsciously take them into account in their answer. Also, the repetitiveness of the task may have encouraged some to give rote answers.

In order to fine-tune our weights, we needed to compute the different scoring factors (*thi*, *tyi*, *rch*, *rat*, *frn*), in order to be able to determine their degree of correlation with the user-provided scores.

Computing scoring factors

Given the above data, for each pair (u, o_j) of a user $u = 1, \ldots, 100$ and a $STOB$ o_j, $j = 1, \ldots, 15$ we computed the five score factors in the following way:

Thematic Interest *thi*(u, o) and Type Interest *tyi*(u, o): we used the interest scores explicitly provided by users in our questionnaire. For $STOBs$ with two themes, we computed the average between the two values. Resulting values belong to the interval $[0, 10]$.

Average Rating *rat*(o) and Friend Participation *frn*(u, o): we used the two corresponding parameters we had associated with each $STOB$. These are therefore simulated values. Notice that also these values fall into the interval $[0, 10]$.

Reachability *rch*(u, o): we used the formula provided in section 5.2.2, with $I_\sigma = [0, 10]$, $dist(\hat{c}_u, \hat{c}_o)$ and $\hat{r}_o$ as specified in the $STOB$ description (that is, $(0, 0)$ and 0.1 respectively), $\hat{r}_u = 0.1$ and mov_u as specified by the user in the questionnaire when asked about his propensity to move.

Fine-tuning the weights for σ^0_{fine} (only thematic and type interest)

In our approach, fine-tuning the weights corresponds to answering the following question:

If we assume to use σ^0_{fine} to predict user's interest, which are the weights that maximize the correlation between our prediction and the actual score provided by the user?

Since σ^0_{fine} considers only thematic and type interest, we only have to assign two weights, whose sum needs to be 1. We can then rewrite the scoring function in Eq. (5.1) as a function of one of the two weights:

$$\sigma^0_{\mathit{fine}}(u, o, x) = x \cdot \mathit{thi}(u, o) + (1 - x) \cdot \mathit{tyi}(u, o), \tag{5.2}$$

We then solve the following problem:

Which is the value of x that maximizes the Pearson correlation coefficient for the samples given by the pairs of the form $(\sigma^0_{\mathit{fine}}(u_i, o_j, x), \tau^0_u(o_j))$, for all $u_i, i = 1, \ldots, 100$ and $o_j, j = 1, \ldots, 15$ (the first member of each pair being the value computed by our scoring function with weight x, and the second being the initial user score)?

By computing the Pearson correlation coefficient for different values of x, we obtain 0.67 as the optimal value. Therefore, for σ^0_{fine} we have

$$\theta_{\mathit{thi}} = 0.67 \text{ and } \theta_{\mathit{tyi}} = 0.33.$$

Fine-tuning the weights σ_{fine} (with additional factors)

We assume that the ratio between the importance of themes and types remains the same as in σ^0_{fine}. We then focus on each of the additional factors individually and compute its relative weight with respect to thematic and type interest:

Firstly, we calculate *the relative weight for reachability.*

We consider D (distance) user scores $\{\tau^D_{u_i}(o) \mid i = 1, \ldots, 100, o \in \mathbf{A}\}$.

We assume that the ratio between the importance of themes and types is expressed by the values θ_{thi} and θ_{tyi} found in the previous step. Then we include in the scoring function the reachability factor as follows:

$$\sigma^D(u, o, x) = x \cdot rch(u, o) + (1 - x) \cdot \sigma^0_{fine}(u, o). \tag{5.3}$$

Similar to what we did before, we solve the following problem:

Which is the value of x that maximizes the Pearson correlation coefficient for the samples given by the pairs of the form $(\sigma^D(u_i, o, x), \tau^D_{u_i}(o))$, for all $u_i, i = 1, \ldots, 100$ and $o \in \mathbf{A}$ (the first member of each pair being the value computed by our scoring function with weight x, and the second being the distance user score)?

By answering this question, we obtain a relative weight for reachability $\theta_{rch} = 0.18$ as the optimal value.

Finally, we calculate the *relative weights for average rating and friends participation.*

We proceed in the same way for the remaining weights. We consider R (rating) and F (friends) user scores, respectively, together with the *average rating* and *friend participation* factors. Moreover, in order to determine the weight for the average rating we consider only the STOBs in group **B**, while for friends participation we consider only the STOBs in group **C**.

We obtain $\theta_{rat} = 0.06$ for the average rating, and $\theta_{frn} = 0.15$ for friends participation.

The final weights ω we selected for σ are then obtained by normalizing the θ values. Table 5.2 shows both relative weights θ and final normalized weights ω.

Computing dynamic weights for σ_{dyn}

Dynamic weights are computed by assuming a dependency between the initial interest of the user for the themes and type of an STOB, and the relevance of the other factors. In other words, distance, presence of friends, and social

f	*thi*	*tyi*	*rch*	*rat*	*frn*
θ_f	0.67	0.33	0.18	0.06	0.15
ω_f	0.48	0.24	0.13	0.04	0.11

Table 5.2: Weights for σ_{fine}

rating, may have a different impact depending on how much a user is interested in the STOB *per se.* For example, if a person is extremely interested in wine, then this person may be willing to travel longer distances or participate in the STOB even when the STOB has lower rating or no friends participate.

If it is so, then the recommendation accuracy may be further improved by computing different weights for the additional factors, depending on the score obtained by σ^0_{fine}, which considers only themes and type.

Therefore, for each STOB group **A**, **B** and **C** we partitioned the samples $(u, o, \tau(u, o))$ according to the score $\sigma^0_{\mathit{fine}}(u, o)$. In order to have a reasonable number of samples for each element of the partition, we used the following thresholds in the scores: $0, 4.5, 6, 7, 8, 9, 10$, thus obtaining six subsets of about 80 samples each.

Then we computed again weights for reachability, average rating and friend participation, with the same method that we used for σ, working separately on each sub-group of samples.

Table 5.3 shows the final ω-weights for each subset.

The aim of the dynamic weight calculation was to study the performance of the recommendation process and, as we will show in Section **??**, it provides the best results w.r.t. the other weights calculation methods we propose. Unfortunately, due to the limited amount of available samples, it is not possible to draw definitive conclusions on the relevance of distance, presence of friends, and social rating depending on user's interest in STOB's themes and type.

Subset	ω_{thi}	ω_{tyi}	ω_{rch}	ω_{rat}	ω_{frn}
$\sigma^0_{fine}(u,o) \in [0, 4.5]$	0.47	0.23	0.08	0.07	0.15
$\sigma^0_{fine}(u,o) \in (4.5, 6]$	0.49	0.24	0.15	0.0	0.12
$\sigma^0_{fine}(u,o) \in (6, 7]$	0.45	0.22	0.135	0.04	0.155
$\sigma^0_{fine}(u,o) \in (7, 8]$	0.44	0.22	0.12	0.08	0.14
$\sigma^0_{fine}(u,o) \in (8, 9]$	0.47	0.23	0.12	0.05	0.13
$\sigma^0_{fine}(u,o) \in (9, 10]$	0.47	0.23	0.13	0.04	0.13

Table 5.3: Dynamic weights

Weight sets for $\sigma_{\cong}$ and σ_{user}

We introduce the scoring functions $\sigma_{\cong}$ and σ_{user} in order to see how our fine-tuned weights (both fixed and dynamic) compare against (i) a function where all the additional factors have the same weights and (ii) a function where the weights of the additional factors are explicitly given by the user. However, in order to determine these scoring functions, we needed in both cases to set weights for thematic interest *thy* and type interest *tyi*, and guarantee that the weighted sum is equal to 1.

As we will see in the following section, σ_{dyn} performs slightly better than σ_{fine}. We therefore chose the weights for thematic and type interest as the average of the weights of σ_{dyn}. In this way we obtained $\omega_{thy} = 0.47$ and $\omega_{tyi} = 0.23$.[5]

For $\sigma_{\cong}$, it follows that $\omega_{rch} + \omega_{rat} + \omega_{frn} = 0.3$ and therefore $\omega_{rch} = \omega_{rat} = \omega_{frn} = 0.1$.

For σ_{user}, given the "weights" $\overline{\omega}^u_{rch}, \overline{\omega}^u_{rat}$ and $\overline{\omega}^u_{frn}$ expressed by users, we rescale them so that their sum is 0.3. The scaling factor is thus $\xi = 0.3/(\overline{\omega}^u_{rch} + \overline{\omega}^u_{rat} + \overline{\omega}^u_{frn})$ and we have $\omega_f = \xi \cdot \overline{\omega}^u_f$ for $f = rch, rat, frn$.

Table 5.4 shows the weights for all the scoring functions we discussed in

[5]Notice that the resulting weights are very close to those in σ: $\omega_{thi} = 0.48$ and $\omega_{tyi} = 0.24$.

this section, and which will be used throughout the rest of the paper.

Function	ω_{thi}	ω_{tyi}	ω_{rch}	ω_{rat}	ω_{frn}
$\sigma_0^=$	0.5	0.5	-	-	-
σ_{fine}^0	0.67	0.33	-	-	-
$\sigma_{\cong}$	0.47	0.23	0.10	0.10	0.10
σ_{user}	0.47	0.23	$\xi \cdot \overline{\omega}_{rch}^u$	$\xi \cdot \overline{\omega}_{rat}^u$	$\xi \cdot \overline{\omega}_{frn}^u$
σ_{fine}	0.48	0.24	0.13	0.04	0.11
σ_{dyn}	(see Table 5.3)				

Table 5.4: Scoring function variants

5.3 Experimental Results and Discussion

In order to answer our research questions, we needed to compare the performance of the different weight sets in predicting how much the users are interested in an event. We decided to carry out two experiments: in the first one we interviewed a new set of users on the same events we used for data collection (see Table 5.1), while in the second one we interviewed a different set of users on a different group of events (shown in Table 5.5).

Our goal, in performing two different experiments, was to see whether the evaluation of the weight sets led to the same results across different users and events, providing evidence that the answers we can draw from experimental results are general enough.

5.3.1 Experimental setting

The two experiments had the same structure: each of them involved a group of users (respectively, U_1 and U_2) and a group of events (respectively, O_1 and

O_2). In both cases subjects were recruited according to an availability sampling strategy; 200 people participated in U_1, and 109 in U_2. Both O_1 and O_2 consisted of 15 events; as we already mentioned, events in O_1 were the same ones we used for data collection and weight calculation (see Table 5.1), while O_2 introduced 15 new events whose types and themes were only partially overlapping with those of the events in O_1 (see Table 5.5).

In the first experiment, we initially asked users to express their potential interest in each event, *knowing only the event description along with its themes and type*. In the following, these will be referred to as the *init* user scores and $\theta_u(o, init)$ will denote the init score given by user u to event o. This step was not included in the second experiment.

Then, in both cases users were asked to provide:

- A score $0 - 10$ expressing their potential interest in each event, *being aware of all the factors (themes and types, distance, average rating and friends' participation)*. These will be referred to as the *fin* user scores and $\theta_u(o, fin)$ will denote the final score given by user u to event o.
- A value $0 - 10$ expressing their personal appreciation of each theme and type.
- A value $0 - 10$ expressing how they perceive they are affected by distance, average rating and friends' participation, when deciding if they are interested in an event.
- The maximum distance (in kilometers) they are generally willing to cover for the sake of participating to an event, in order to compute reachability.

5.3.2 Results

In order to evaluate the performance of each weight set in predicting the users' interests, we computed the RMSE (*Root Mean Square Error*) between

STOB	Themes	Type	Distance [km]	Friends Nr.	Rating
o_1	beer	meeting	7	4	8
o_2	meat, fruit and vegetables	cooking course	0	1	6
o_3	chocolate, wine	dinner	18	10	9
o_4	oil, wine	dinner	0	2	8
o_5	cheese, oil	workshop	0	0	7
o_6	beer, chocolate	workshop	0	10	6
o_7	fruit and vegetables	cooking course	0	6	4
o_8	wine	meeting	0	5	8
o_9	fruit and vegetables, wine	dinner	6	3	6
o_{10}	fruit and vegetables	cooking course	22	1	9
o_{11}	meat, wine	workshop	55	2	3
o_{12}	meat, cheese	cooking course	0	6	10
o_{13}	cheese, wine	meeting	0	0	7
o_{14}	cheese, wine	dinner	65	5	7
o_{15}	oil, fruit and vegetables, wine	workshop	0	2	5

Table 5.5: Set of events in the *Experiment 2*

the predictions given by the scoring function ("system scores") and the values explicitly provided by users ("user scores"). RMSE is a well-accepted statistical accuracy metric for recommender systems [10, 58, 33, 12]. A lower RMSE denotes a higher degree of accuracy; as a consequence, a negative variation in the RMSE is regarded as an improvement. As pointed out by Arazy et al. [12]: "Even small RMSE improvements are considered valuable in the context of recommender systems. For example the Netflix prize competition[6] offered a one million dollar reward for an RMSE reduction of 10 percent."

Tables 5.6, 5.7, 5.8 and 5.9, discussed below, illustrate the results of our experiments.

[6] http://www.netflixprize.com

In the first experiment, we obtained two sets of user scores: those provided knowing only the types and themes of the events (*init* user scores) and those provided knowing also distance, average rating and friends' participation (*full* user scores).

		System weights						
Set	**User scores**	ω_{thi}	ω_{tyi}	ω_{rch}	ω_{rat}	ω_{frn}	**RMSE**	$\Delta(\%)$
$\boldsymbol{\sigma^0_=}$	*init*	**0.50**	**0.50**				**2.583**	-
$\sigma^0_=$	*full*	0.50	0.50				2.909	+12.62%
$\boldsymbol{\sigma^0_{fine}}$	*init*	**0.67**	**0.33**				**2.475**	-
σ^0_{fine}	*full*	0.67	0.33				2.874	+16.12%

Table 5.6: Results of the comparison of users' ratings collected in the first survey and system output obtained taking into account only themes and type

Table 5.6 shows the behavior of the scoring function in Experiment 1, under the hypothesis that the recommender system knows only the user's preference with respect to themes and types, but has no information concerning other factors. In this case only two weight sets are meaningful: $\sigma^0_=$, which assigns equal weights to themes and types, and σ^0_{fine}, which assigns fine-tuned weights to themes and types (see *Fine-tuning the weights* in Section 5.2.2). The first column reports the chosen weight set. The second column shows which type of user scores we are considering, while the next five columns show explicitly the weight values in the weight set. The last two columns report respectively the RMSE between the system predictions and the user scores, and the variation of the RMSE expressed as a percentage with respect to the reference value shown in bold face.

It is easy to see that, while computed weights σ^0_{fine} perform better than equal weights $\sigma^0_=$ when all other conditions are the same, there is a significant deterioration of the RMSE (+16.12% for σ^0_{fine} and +12.62% for $\sigma^0_=$) when the user is assumed to possess full information about the event.

Set	**System weights**					**RMSE**	$\Delta(\%)$
	ω_{thi}	ω_{tyi}	ω_{rch}	ω_{rat}	ω_{frn}		
$\sigma^{\mathbf{0}}_{fine}$	**0.67**	**0.33**				**2.874**	-
σ_{user}	0.47	0.23	$\xi \cdot \overline{\omega}^u_{rch}$	$\xi \cdot \overline{\omega}^u_{rat}$	$\xi \cdot \overline{\omega}^u_{frn}$	2.754	-4.19%
$\sigma_{\cong}$	0.47	0.23	0.10	0.10	0.10	2.751	-4.29%
σ_{fine}	0.48	0.24	0.13	0.04	0.11	2.729	-5.06%
σ_{dyn}	(see Table 5.3)					2.687	-6.50%

Table 5.7: *Experiment 1*: recommendation accuracy for different weight sets.

Table 5.7 also shows figures concerning Experiment 1. Here we consider only *full* user scores, and we evaluate the performance of all scoring functions that include the additional factors, comparing their accuracy with that of σ^0_{fine}.

The columns of this table have the same meaning as the columns in the previous table (Table 5.6); rows are shown in order of decreasing RMSE. The best results are obtained for this experiment with dynamic weights (there is a RMSE improvement of -6.5% with respect to σ^0_{fine}), followed by fine-tuned weights (-5.06%). Error increases with equal "additional" weights (RMSE gets to -4.29% with respect to σ^0_{fine} and $+2.38\%$ with respect to σ_{dyn}), which perform more or less the same as user-defined weights.

Table 5.8 concerns Experiment 2, with the same approach as the one adopted in Table 5.7. It is easy to notice that the order of the rows is the same, meaning that by changing the event set we still obtain the best results with dynamic weights. Second best are fine-tuned, non-dynamic weights, followed by equal "additional" weights and user-defined weights.

Table 5.9 does not introduce any new information: it compares the RMSE results of Experiment 1 and Experiment 2 (reporting both the RMSE abso-

	System weights						
Set	ω_{thi}	ω_{tyi}	ω_{rch}	ω_{rat}	ω_{frn}	**RMSE**	$\Delta(\%)$
$\sigma^{\mathbf{0}}_{\mathit{fine}}$	**0.67**	**0.33**				**2.934**	-
σ_{user}	0.47	0.23	$\xi \cdot \overline{\omega}^u_{rch}$	$\xi \cdot \overline{\omega}^u_{rat}$	$\xi \cdot \overline{\omega}^u_{frn}$	2.795	-4.75%
$\sigma_{\cong}$	0.47	0.23	0.10	0.10	0.10	2.772	-5.51%
σ_{fine}	0.48	0.24	0.13	0.04	0.11	2.756	-6.05%
σ_{dyn}			(see Table 5.3)			2.715	-7.45%

Table 5.8: *Experiment 2*: recommendation accuracy for different weight sets.

	RMSE		$\Delta(\%)$	
Set	Exp. 1	Exp. 2	Exp. 1	Exp. 2
$\sigma^{\mathbf{0}}_{\mathit{fine}}$	**2.874**	**2.934**	**-**	**-**
σ_{user}	2.754	2.795	-4.19	-4.75
$\sigma_{\cong}$	2.751	2.772	-4.29	-5.51
σ_{fine}	2.729	2.756	-5.06	-6.05
σ_{dyn}	2.687	2.715	-6.50	-7.45

Table 5.9: Comparison of the results obtained in Experiments 1 and 2

lute value and its improvement with respect to σ^0_{fine}). If we compare the results of Experiment 1 and Experiment 2 we can see that:

- In general, the RMSE got slightly worse in Experiment 2; this was to be expected, since Experiment 2 focused on a different group of events than those considered when initially computing the weights.

- The degree of improvement obtained by the scoring functions is consistent across the two experiments (even slightly better in Experiment 2). This suggests that our findings are not dependent on the group of users and/or events.

5.3.3 Discussion

Let us consider again the research questions we introduced in Section **??**, and discuss how the results of our experiments can provide us with answers.

Our first question concerns the consequences of not taking into account the additional factors in the recommendation process.

RQ1 What is the accuracy of pure content-based recommendation (i.e. considering only the themes and type) for events? As users typically know more about the event than its theme and type, how does this knowledge affect their choices and the recommendation accuracy?

Table 5.6 shows that, if we consider a purely content-based scoring function, the RMSE changes significantly depending on whether users know more about the event than the system does ($+12,62\%$ for $\sigma^0_=$ and $+16.12\%$ for σ^0_{fine}). In other words, if we ask users to evaluate their interest in participation based only on themes and type of the event, their answers would be quite different, and more similar to the score provided by the recommender. However, this is not the typical situation in real-life: users do know things such as the event location and its distance from their own location, they are aware of whether any of their friends would participate, and in a Web 2.0 context they are likely to know the opinion of the community about the event. Results show unmistakably that not taking the additional information into account is significantly detrimental to recommendation.

Table 5.6 also shows that, by using the $2:1$ ratio between thematic interest *thi* and type interest *tyi*, we obtain a slight improvement in the RMSE: -4.18% against *init* user scores and -1.2% against *full* user scores. This results prompted us to use the 2:1 ratio in the rest of the scoring functions.

After evaluating the consequences of not considering the additional factors, our second research question investigates the advantages of including them.

RQ2 If we provide the system with additional information, does this enhance recommender system accuracy? Namely, if we include reachability, average rating and friends participation in the recommendation, does the accuracy improve?

If we look at the results of both experiments as summarized in Table 5.9, we can see that all scoring functions that include the additional factors improve in their accuracy, regardless of the chosen weight set. In Experiment 1, the decrease in the RMSE ranges from -4.19% (same weights for the three additional factors) to -6.50% (dynamic weights). In Experiment 2, it ranges from -4.75% (same weights for the three additional factors) to -7.45% (dynamic weights).

Next, we investigate the relevance of each additional factor.

RQ3 To which extent should each additional factor contribute to the recommendation process? More precisely: do we obtain any advantage by fine-tuning the weights of the additional factors?

We can attempt to answer this question by comparing the results obtained in our two experiments with weight set σ_{fine} (fine-tuned weights) and weight set $\sigma_{\cong}$ (all additional factors with the same weight). We can see in Table 5.9 that fine-tuned weights provide a RMSE improvement of -0.8% in Experiment 1 and of -0.58% in Experiment 2. Although the improvement is not very significant, we can nevertheless conclude that there is some advantage in fine-tuning the weights.

An alternative option to fine-tuning the weights is to directly ask the users about their importance for them. This brings us to the next research question:

RQ4 Would letting the users explicitly voice their preferences regarding additional factors provide any improvement in the recommendation?

Table 5.9 shows that the answer is "no": user-defined weights (row σ_{user}) actually provide worse accuracy in recommendation than assigning equal weights to all additional factors.

Finally, we want to see whether the relevance of the additional factors actually depends on how much the user is interested in the event content. For example, distance or lack of friends may not matter much if a person is very interested, or if she is not interested at all, while they may count when the person is undecided or has a moderate interest.

RQ5 Does the relevance of additional factors depend on how much the user is interest in the event content?

We can try to answer this question by comparing the performance of dynamic weights σ_{dyn}, which are computed under the hypothesis that such a dependency exists, with the other weight sets. Actually, it turns out (see again Table 5.9) that dynamic weights provide the best accuracy, so the answer to the last question is "yes, there is a dependency." We can indeed observe an improvement over fine-tuned weights of -1.54% for Experiment 1 and of -1.49% for Experiment 2.

As a concluding remark, we can observe that the most significant improvement in the RMSE is obtained by including our additional factors in the scoring function in the proportion of 7:3 with respect to content-based factors. Adjusting the relative weights of the additional factors produces a further, albeit slight, improvement.

5.4 A proposed architecture for the recommendation process

The reasoning process is studied with the aim to provide a recommendation of event in the context of a *SWIT* (see Chapter 1, [23]), where "things", including events, are regarded as autonomous entities. Thus, we consider that an event may propose itself to users and, therefore, the scoring function for a given user u needs only to rank events that have "invited" u to participate.

As consequence, our reasoning process involves two separate phases:

- the *selection* phase, where the system discards the *STOBs* that are not at all interesting for the user or that do not satisfy his spatiotemporal constraints, creating a *pool* of potentially interesting *STOBs*,
- the *ranking* phase, where the *STOBs* in the *pool* are sorted by taking in account the scoring function introduced in Section 5.2.2.

The advantages of this approach is that it does not need to compute the scoring function for all pairs of users and *STOBs*

Figure 5.3 describes the whole recommendation process, showing how it interleaves with user interaction. In fact, although for this work we neither explicitly addressed the problem of how the recommendation results are displayed to the user, nor we investigated interaction modalities that allow him to fine tune the recommendation, we believe that a successful recommendation scheme must provide enough information to the UI Module so that it can handle these tasks in a flexible and affordable way.

As we already said, *selection* creates a working pool of *STOBs* that will be the input for the rest of the process. However, we assume that the user may want either to manually add *STOBs* he likes to this pool, to be later reminded of them, or to remove once and for all those she discards, in order not to see them anymore. Since we do not want a rerun of the *selection* process to overwrite these operations, the process saves the add/remove actions performed by the user in a user operations repository.

5.4.1 Selection

The goal of *selection* is to build, for each user, a *pool* of potentially interesting *STOBs* to recommend to him, ruling out everything that is definitely out of

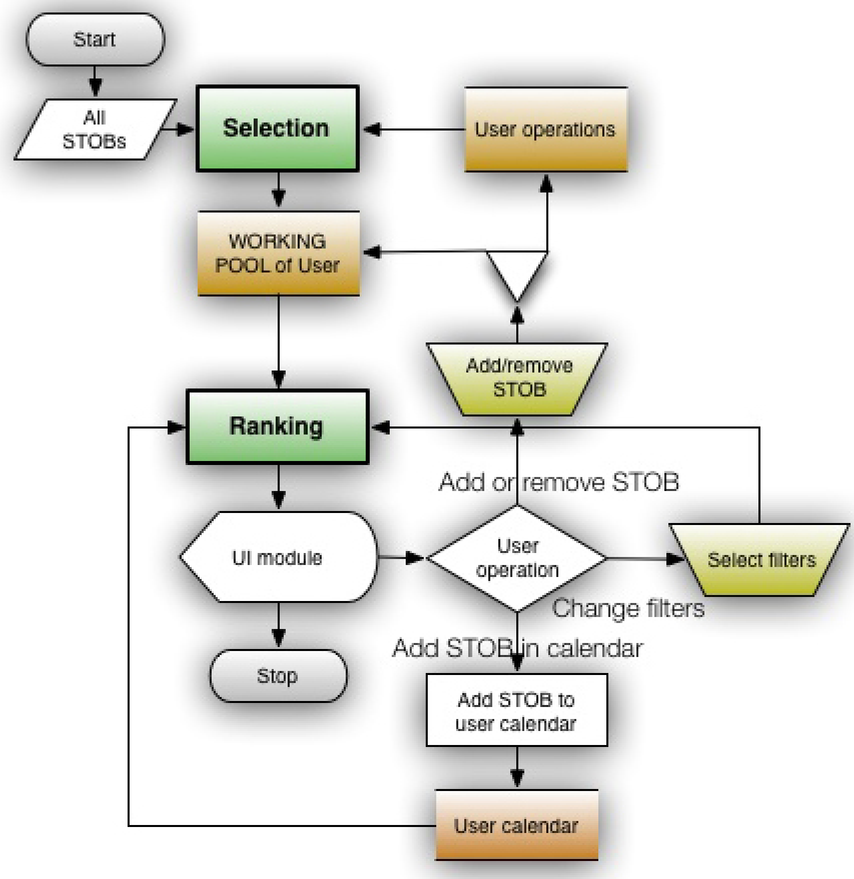

Figure 5.3: Selection and Ranking of *STOBs*

the user's interest. Of course, if the user stumbles upon an *STOB* that had been previously ruled out, he may have the possibility of adding it back to his pool. However, the rationale of the *selection* phase is to prevent the user from being flooded with *STOBs* proposals and to make recommendation more effective by focusing it on a well-defined group of items.

The *pool* determined through the *selection* phase goes through an additional filtering process as more transient user constraints are applied to it. The output of this phase is the *working pool*. To create an efficient *pool* some criteria have been selected:

- all *STOBs* that have already occurred are removed from the list of *STOBs* in the case of proactive context, while in on demand context only *STOBs* in the requested timeframe are considered;
- all *STOBs* geo-localized outside the maximum radius r_{MAX} of the user *user* are not considered;
- all *STOBs* with a user's interest, that does not exceed a minimum threshold are eliminated from the list.

In some cases, there are *STOBs* that have a user's interest with a high value. If this interest exceeds a maximum threshold the criterion of spatial distance could not be considered.

The *working pool* can change dynamically as a consequence of the user's interaction with the system (i.e. he decides to change the recommendation time frame) or it can be edited manually (i.e. the user decides to discard a *STOB* he is not interested in).

5.4.2 Ranking

The aim of the *ranking* phase is to sort the *STOBs* stored in the users *working pool* considering different factors (see Section 5.2.2).

We can further split it in two sub-phases: (1) the computation of the *individual ranking factors*, and (2) their *merge into the global scoring function* (see the function 5.1).

The simulator is not based on a real social network, therefore some factors work on dynamic values that are given as input and not calculated on user behaviors or on his real current network:

- of each user, we know his user model, his relationship network, his position and his propensity to move (as pre-calculated values),

- of each event, we have the average ratings and the participants.

Thanks to these information it is possible to calculate all *individual ranking factors*. Also for the *Reachability* the simulator uses the services of Google Maps API[7] to calculate the distance between the $STOB$ and the user position.

The factors weights are predefined and follow the values found in the Section 4.5.

At any time, the user can decide to add an $STOB$ from his calendar to keep track of it; on the contrary, the removal of $STOB$ that is not more interesting.

He also can change his filters to display $STOBs$ according to a specific criterion (*i.e.* interests, date, proximity, popularity, interest of his friends). In this case, the list of $STOBs$ are modified according to the new criterion through a modification of the weights.

5.5 Conclusions

In this chapter, we investigated the concept of “event” and its features (content, context and social features). We examined several factors that may have an impact on how users perceived events as interesting. We carried out two studies with users that allowed us to assess the validity of a number of hypotheses on the impact of the different factors on recommendation.

We thus saw how, although the relevance of rankings, social participation and physical distance/reachability is definitely lower than the relevance of the event type and themes, neglecting such factors brings about a significant deterioration in the accuracy of the recommendation.

On the overall, adding events to our $SWIT$ knowledge base was a significant challenge, as they proved to have features that make them quite different

[7]Google Maps API `https://developers.google.com/maps/`

from other content types with respect to recommendation and presentation to users.

An aspect we have not yet considered but that we see as relevant, is how user interaction can facilitate a collaborative recommendation process where the user, through his behavior, offers cues to the system.

6

User interaction in SWIT

The last part of our work focuses more on user interaction in a *SWIT* context, and in particular in the framework of the *Wanteat* system. A natural, flowing interaction is crucial to an effective implementation of a *SWIT*. In fact, according to *SWIT* principles, users interact with digital avatars of real things in a way that does not disrupt the flow of physical interaction in real life.

In this Chapter, we introduce two studies of user interaction that contribute to the *Wanteat* suite of appplications. These applications target different categories of users (e.g., tourists, food producers, etc.) and different contexts. The world of gastronomy is conceptually represented as a mixed social network of users and domain objects (e.g., products, restaurants or territories), linked by various types of ontological and social relationships that are reflected in the dynamic user interface of "the wheel" [2].

Wanteat mobile[1] (see Chapter 1) is the first client application implemented in the project. It focuses on an expressly designed interaction paradigm, "the wheel" which we will further discuss in the following as it plays a central role also in one of our applications.

Thanks to the ontological representation of knowledge, we have extended the *Wanteat* application with videos, implementing the *Wanteat Video* [1].

[1] Wanteat `http://www.wanteat.it`

Each videos is characterized by a title, a description, some tags, comments and rating, and is always connected with at least one content in "the wheel" from which inherits both its ontological and social relationships.

Starting from the idea of facts as "advanced tags", we have studied another user interface, *Telleat*, that allows user to tell something about entities within the system in a playful way.

Thus, the following two sections describe respectively:

Wanteat Video - an application for Apple iPad™ which allows user to explore content of *Wanteat* and watch the related videos. It exploits the paradigm of "the wheel" and users have a central role both in producing content and in affecting system behavior. The videos are always connected with some content and users can explorer them by content navigation on "the wheel".

Telleat is an application for Apple iPhone™ that is implemented as an add-on to the *Wanteat* system, allows user to tell facts (see Chapter 3) about domain objects and their experience with them in a playful way. A pertinence module on the server evaluates the pertinence between new facts. Facts and pertinence values are passed to the *Wanteat* system that will decide how to use them in suggesting further content.

6.1 Wanteat Video

Wanteat Video is an Apple iPad™ application that extends the functionality of *Wanteat* with multimedia content. Also it maintains the *Wanteat* structure of mixed social network of user and objects with their ontological and social relationships.

Each network member (user or domain objects) is described by means of textual information, images and a series of related videos: for example, the videos which describe *Agnolotti del Plin* (a typical Piedmontese recipe)

Figure 6.1: Wanteat Video

might explain how they are prepared, tell stories about village festivals when they are typically served or about renowned producers. The peculiarity of *Wanteat Video* is that users do not navigate videos directly, but they have to access objects in the gastronomy domain to discover related videos.

Users of *Wanteat Video* have a very central role. On the one hand, acting as content producers, users can contribute to enrich the existing contents with first-hand evaluations in the form of comments, bookmarks, ratings or tags, and to create new explicit or implicit relationships among them. On the other hand, user features were taken into account at various levels in order to design the behavior of *Wanteat Video*. At the most general level, *Wanteat Video* targets final users (e.g., tourists, food enthusiasts) and is especially meant for relaxed fruition contexts, given that the device requires two-handed operation on the part of a stable standing or seated user, and that watching videos is a time-consuming activity. At a more specific level, *Wanteat Video* distinguishes among different user categories, according to the context of use. To prototyping purposes, we defined a restaurant context (users are at a restaurant, waiting to be served) and a personal use context (users want to learn more about the gastronomic heritage out of personal interest). Other possible contexts include a food festival one (users are planning to visit or are visiting a food exhibition). Finally, at a user-specific level, content presentation is personalized according to user location and interests, which are inferred from user actions and represented in an explicit user model.

6.1.1 Interaction Model

Wanteat Video allows users to interact with mixed social networks of users and domain objects, where network elements are represented by means of descriptive textual information, images and videos.

As content consumers, users can explore the gastronomy domain by browsing the available contents and following the relationships which link them. As content producers, users can enrich contents with their comments, ratings, tags and bookmarks, thus fueling the system with information which can serve as a basis for establishing new relationships among domain contents. Thus, users do not only generate new content, but contribute to its organization.

In order to support user activities, an interaction model was adopted for *Wanteat Video* which allows to: i) *select some specific content* users are interested in; ii) *access information* about some specific content; iii) *explore the gastronomy domain*, moving from a certain content to related ones; iv) *generate new content*. All interaction is based on touch, which is a natural way for expressing interest.

Interaction Model

User interaction always starts with the selection of some specific domain content. This can be the end point of navigation, if users only aim at accessing information about such content, or serve as a starting point for the exploration of the gastronomy domain, if users are also interested in related content. Since *Wanteat Video* can be used in different contexts, different options for selecting such initial content are provided.

(a) (b)

Figure 6.2: The interactive menu

In the *restaurant context*, an interactive menu is used (see Figure 6.2(a)). Based on the idea that selecting an item from a restaurant menu corresponds to put it in ones own dish, we experimented with a prototype where icons of the available dishes are presented which can be dragged and dropped onto the image of a dish for selection. In the *personal context*, several options are provided (see Figure 6.2(b)). First, a standard search facility is offered.

Second, users can select some content from their bookmarks. Third, they can ask for a personalized recommendation. Finally, they can geo-localize themselves and select some content which is located in their surroundings.

Exploring the gastronomy domain

Domain-specific and user-generated relationships among objects and users define possible navigation paths for exploring the gastronomy domain. A navigation model was devised, based on the visual concept of a wheel, which centres on such relationships: given an element which represents the current focus of interest, related objects are presented among which a new focus can be selected, thus defining a step-by-step curiosity driven exploration model. In a wheel, the central area, the external border and the spokes connecting the centre to the border can be distinguished. While the wheel centre is well suited to represent the current focus of interest, related elements can be arranged along the wheel border and spokes are a good symbol for relationships.

In the wheel-like interface we designed, the element which represents the current focus (*wheel focus*) occupies the centre of a circle (see Figure 6.2). In order to reduce information overload and ease navigation, the surrounding area was divided into four differently coloured sectors, corresponding to high level categories of possible related users and objects. Such sectors vary according to the type of element in focus. For example, when the element in focus is a product, sectors will be labelled products, users, cuisine and territory, while they will be labelled recipes, products, users and territory when the element in focus is a recipe.

Sectors can be expanded on demand to let users access their content (see Figure 6.3).

When a sector is expanded, it takes the form of a quarter-circle, while the other sectors and the element in focus are momentarily miniaturized, in order to let users concentrate on the content of the current sector. Contents are

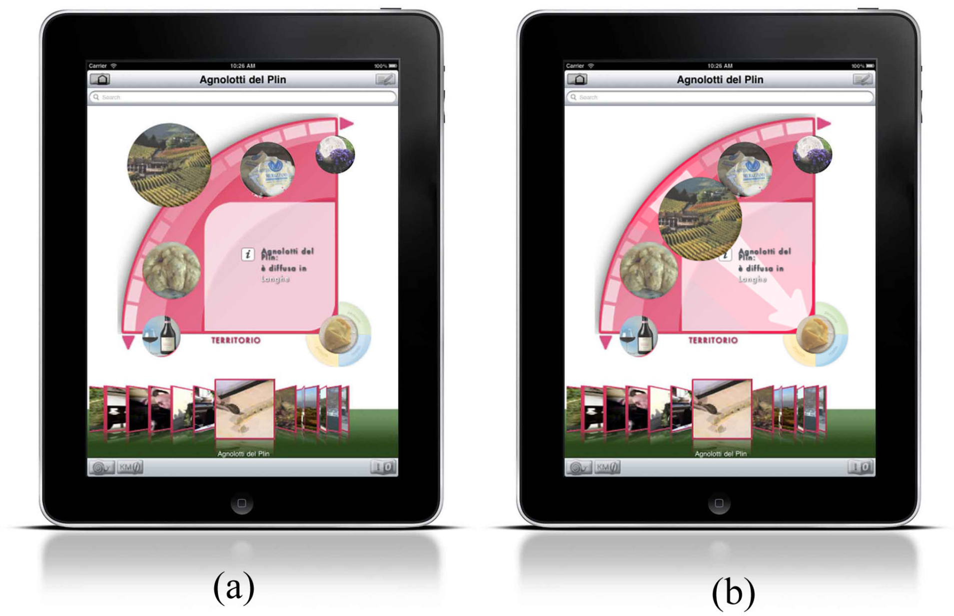

Figure 6.3: An open sector

visualized by means of circular icons which can be browsed by dragging and rotating them along the sector border, as in an old-style telephone selector. The element which occupies the central position along the sector border (sector focus) is given more prominence by magnifying it. Moreover, a short description of the relationship which links the wheel focus to the sector focus is provided. For example, if both elements are products, the relationship description can explain that they are similar (e.g., in the case of two cheeses prepared with the same type of milk and similar production techniques), or that they go well together (e.g., in the case of a cheese and a matching wine).

Users can change the wheel focus by dragging the current sector focus towards the centre of the wheel, i.e., onto the miniaturized wheel icon. When the wheel focus is updated, all wheel elements are reconfigured consequently. Finally, a set of customizable filters is provided as a facility for helping users to find elements which match their interests. On the one hand, users can decide what types of elements should populate the wheel model, choosing among categories such as products or restaurants. On the other hand, they can focus only on the elements which are in harmony with a particular philosophy, e.g. Slowfood ethics or organic production.

Accessing Information

Users can access detailed information about any element, provided that it is the current wheel or sector focus, by simply touching it. A multimedia file (element detail) for the selected element is presented, which consists of two sections. The first section contains descriptive textual information, accompanied by images and user generated contents such as tags and ratings, while the second section contains videos. Since videos are complex objects themselves and multiple videos can be associated to a certain content, a specific organization and interaction model for videos was devised, which we will present in the subsection *Videos*.

Generating New Content

According to Web 2.0 principles, part of *Wanteat Video* contents are user-generated: more specifically, users can enrich objects with their tags, comments, ratings and bookmarks. Users can perform such social actions whenever a single element is clearly identifiable as the object of their actions, that is, when the wheel focus is visible (no sectors being expanded) or when they are accessing an element detail. The action menu can be visualized on demand by clicking on the edit button in the navigation bar, which is active only when actions are allowed. In order to maintain a clear reference to the element which is being rated, commented, bookmarked or tagged, the forms for social actions are displayed in a popover window which leaves part of the navigation interface visible.

Videos

As mentioned above, videos are always associated to specific elements of the domain knowledge base and contribute to describe them, e.g., they can show an interview to a producer, how to prepare a traditional recipe or the history of a certain town.

Each video is annotated with a label which classifies the content of the video following a set of predefined general categories. For instance a video telling the *history* of a particular country or the evolution of a recipe over the years is labelled as history. In that sense categories represent a first coarse-grained filter that helps users in fast video identification.

Videos can be accessed in three situations: when the whole wheel is visible (no sectors are expanded), when a specific sector is expanded and when an element detail is visualized.

In the first case, videos associated with all the elements in the wheel are presented in a dedicated section in the bottom part of the interface (see Figure 6.3). They are organized in the form of a list of video thumbnails that the user can scroll through the *cover flow* effect, commonly used by Apple devices to display photo albums or music covers.

In order to ease video selection, videos are grouped into sub-lists corresponding to the wheel sectors: a border of the same colour of the corresponding sector surrounds video thumbnails. For instance, a red border indicates videos associated with an element in the territory sector and so on. Grey borders identify videos directly associated to the wheel focus. When a user changes the *wheel focus*, all videos change accordingly with the new wheel configuration. When a sector is expanded the list of videos is automatically filtered, in order to maintain only the videos that are related to elements in the current sector.

Finally, when an element detail is visualized, only the videos directly related to the current element are presented. In this case the videos are organized according to the general video categories and a filter which allows to show or hide videos based on such categories is provided. When users select the thumbnail of a video they would like to see, a multimedia file (*video detail*) is presented with the video in play mode together with its main features (e.g., the title and a short description, see Figure 6.4).

Figure 6.4: A video detail

As it already happens for domain elements, users can perform various Web 2.0 actions on videos (tagging, rating, commenting or bookmarking, inserting facts as *advanced tags* to describe the semantic content of the video). Evidence about user social actions will appear in the video detail, e.g., user comments, average rating and tags.

6.1.2 Knowledge Model

In Section 6.1.1 the interaction model has been defined as a step-by-step curiosity driven exploration model. Users explore the domain following the relations between the domain items and thereby obtain new sets of related videos to play.

In order to support this model the requested information is represented as a set of connected resources organized in a Resource Oriented Architecture (ROA)[40]. Resources in this context are the objects of the domain (products, production places, territories, recipes, restaurants, etc) and the users of the system (together with the social actions they perform).

Although at a logical level resources are described as single uniform entities, the knowledge describing a resource derives from heterogeneous sources, as well as the kind of relations connecting resources are different. In particular three main kinds of knowledge are managed by the system:

- *Domain knowledge*, that maintains the knowledge about the items in the gastronomy domain. Products, recipes, locality and all the involved actors (restaurants, producers, vendors) are modeled as part of this knowledge. An ontology representation is used to this purpose (see Section *Ontology Model*)
- *Social knowledge*, that maintains information about users and their social actions (comments, tags, etc)
- *Multimedia knowledge*, where videos related to the domain are stored

Ontology Model

Ontologies describe the domain elements by expressly representing their properties and the relations among them, providing a structured and organized description of the domain. Moreover, several ontology-reasoning strategies make properties and information that are implicit in the knowledge base available, by means of inference.

The items of the *domain knowledge* are modelled as elements of the ontology, which contains very general classes such as *Recipe* or *Resturant*, as well as very specific ones, such as *Ristorante La Baritlera* (a restaurant near Turin) or *Agnolotti del Plin* (a typical recipe of Piedmont cuisine).

In order to describe this heterogeneous knowledge, ranging from products to territories described at a general, as well as at a specific level, the domain model is composed of several ontologies, focused on different parts of the domain.

Relationships in the domain

The relations defined among resources drive user browsing. As mentioned in previous sections relations are classified according to different sectors and types. It is also interesting to note that relations among the items can have very different origins, in particular they can be:

- defined in the ontology directly (by means of object properties or ISA relations) or derived by means of reasoning; e.g. a recipe is related to a food farm if it uses a product produced by the farm as an ingredient;
- defined by rules that relate different elements in the ontology based on their properties; for instance rules are defined which relate wines and recipes based on their ingredients;
- generated explicitly by users:
  - users link themselves to the elements of the domain by means of their social actions (comments, tags, etc);
  - users link domain elements among them, e.g.; they can associates two recipes or a recipe and a particular wine in a comment;

- system generated:
  - users can be connected to similar users or to domain elements which are related to similar users;
  - an element can be connected to another element because most users who like the first element also like the second one;
- defined by the usage context; for instance in the restaurant context:
  - the menu of the day makes relations between the clients in that restaurant and the recipes;
  - clients of the restaurant are related among them and with past clients of the same restaurant.

Notes on Architecture and implementation

The here described Resource Oriented Architecture is realized by a Service-Logic server component that, exploiting the Restlet framework[2], defines the resources and provides the needed RESTful [40] web services to the clients, so that they can inquiry and modify them.

The underlying data model managed by the ServiceLogic includes the ontology (see Subsection *Ontology Model*) defined using the Ontology Language OWL 2, and standard database technologies. In particular a Video DB is used to maintain the video features and the associations between the videos and the domain objects. The ServiceLogic has in charge the logic needed to integrate these heterogeneous knowledge sources in uniform resource descriptions.

The communication between the server component and the client applications takes place by means of JSON strings, exchanged through HTTP.

[2]Restlet framework `http://www.restlet.org`

6.1.3 Wanteat Video as a user medium

Personalized content fruition

Offering an engaging user experience is an important goal in *Wanteat Video*. Personalization is a popular way to support user experience where the contents and behaviour of an application are adapted to the preferences and needs of a specific user or category of users. In our system, personalization is provided at different levels.

At the highest level, the appearance and behaviour of *Wanteat Video* is customized in the option-setting phase according to the specific context of use, which in turn identifies a category of users. As a first step, we identified two possible contexts with their corresponding user categories:

- *Restaurant*. Users in this context correspond to the restaurant customers, who are primarily interested in the dishes which are served at the restaurant.

- *Personal*. This is a generic relaxed fruition context. Users in this context are interested in exploring the gastronomy domain according to their personal preferences.

In the first case, the initial content selection metaphor is context-specific (e.g., an intelligent menu for the restaurant context) and only contents which are physically available (e.g., a dish which is actually served at the restaurant) can be selected as a starting point. The same kind of personalization would characterize a further possible context such as the food festival one. In the second case, on the contrary, we did not limit the set of possible initial contents. However, we considered that user current location might be an important factor for identifying possible sub-contexts, assuming that users are likely to assign more relevance to local contents.

At a more specific level, *Wanteat Video* can act as a recommender system which, in each context, personalizes navigation according to the interests of a single user. User interests for domain objects are determined according to cognitive filtering: explicit user models describe the interests of each user with respect to all domain features, as they are represented in the domain ontology, and scores indicating the predicted level of interest of a certain user for a certain item are determined as a function of their interest for the ontology features which describe such an item. Interests are inferred from user social actions, considering that different actions can be more or less strong indicators of user interests. More specifically, we adopted an approach similar to the one described in [19].

Interface personalization is also provided by personalizing the filter bar (see *Exploring the gastronomy domain* in Section 6.1.1) according to user behaviour, in order to provide shortcuts for immediately activating/deactivating the most often used filters.

User-driven content organization

In *Wanteat Video*, users do not only consume existing contents, but enrich them with their knowledge and contribute to their organization. As explained in *Relationships in the domain* in Section 6.1.2, in fact, new relationships among domain elements can be established by users either explicitly or implicitly, according to their social actions and browsing behaviour. A new relationship can be created explicitly if, for example, several users comment on a product (e.g., a goat cheese), suggesting to match it with a certain type of red wine. A new relationship can be created implicitly if several users visualize the detail file of two elements one after the other, for example in the case of two recipes which can be served together as part of a traditional meal (e.g., *Agnolotti del Plin*, a first course, and *Brasato al Barolo*, a second course).

Thus, knowledge organization in *Wantet Video* does not only base on static properties of the domain elements, but dynamically evolves according to user activities. User-generated relationships determine new possible exploration paths which reflect trends and common preferences among *Wanteat Video* users, thus representing useful shortcuts to potentially interesting contents for new users.

6.1.4 Evaluation

A large-scale user evaluation was performed in October 2010, during the International Food Fair "Salone del Gusto" in Turin, Italy, in order to experimentally assess the wheel interaction model, the robustness of *Wanteat* system, and the consistency of the available data. On that occasion, *Wanteat Mobile*, an application for Apple iPhone which offers no videos, but features the same wheel-like interaction model and social actions as *Wanteat Video* was used. The evaluation involved more than 600 users and very positive results were obtained as far as the comprehensibility, aesthetic pleasantness, usefulness and ease of use of the application were concerned. Although *WantEat Video* adds a significant part of interaction with videos, we are inclined to believe that these positive proprieties are not lost with the extension. The basic mechanisms of interaction with videos are similar to the standards used by popular applications (for example, landscape mode of Youtube[3] or Joost Application[4] or web 2.0 actions of the videos in DailyMotion Application[5] or Facebook site[6]) and they should therefore not cause difficulties to the users. However, a specific user evaluation is planned with the aim of assessing the whole user experience with *Wanteat Video*.

6.2 Telleat

The *Telleat* application has been developed as an add-on to the *Wanteat system* and its goal is to provide users with an additional way to share in-

[3]Youtube `http://www.youtube.com`
[4]Joost `http://itunes.apple.com/it/app/id295977505?mt=8`
[5]DailyMotion `http://itunes.apple.com/it/app/dailymotion/id336978041?mt=8`
[6]Facebook `http://www.facebook.com`

formation with the system and other users. *Telleat* enables users to tell to the system what they did with a certain object. For example, they may have sipped a wine, brought it to a dinner party, or used it in a recipe. As a "reward" for telling something to *Telleat*, users get back a list of facts told by other people that the system reckons to be pertinent to the fact inserted by the user. This means that users can tell facts with the intent of querying the system for similar things happening to their friends.

Conceptually, *Telleat* is composed of the following modules (a more detailed description of the architecture and its implementation is given in 6.2.2).

- A **client app** for Apple iPhone™, that allows users to interact with the system and share their contributions in a playful way.

- A **fact repository**: as described in [37], we represent facts in OWL, as instances in an ontology of verbs. Conceptually, each fact is a pair (p, F) where p is a chosen verb, representing the action, and F is a set of pairs (r_i, f_i), representing the actors and their roles in the action. For each of these pairs, r_i is a role label chosen among who, what, where, when, how, why; f_i is the role filler which can either be an entity in the domain ontology of *Wanteat* (person, thing, place, etc.) or a custom label defined by the user.

- A **pertinence module**, which evaluates the pertinence between a newly inserted fact and those existing in the repository. The measure of pertinence we use is introduced in [37] and is based on the friendship between involved people, on the colocation between facts (co-occurrence in space and time), and on semantic similarity between the mentioned entities, computed using the distance based approach proposed in Chapter 4.

6.2.1 The Client App: Interaction Model and User Interface

In order to build a fact (p, F) users need to provide the verb p and as many role fillers as they like for each of the six available roles. The choice of the verb is restricted to the verbs present in our ontology; role fillers can be either entities (objects, places, people) that are present in the *Wanteat* social network, or custom entries made of a label, a short description and an optional image.

For our interaction model we decided to use the metaphor of a letter: the fact is represented by an envelop, containing paper sheets for the different roles, in different colors, and with a stamp showing the chosen verb.

The user starts by selecting a verb, and then moves on to provide fillers for the roles; it is always possible to go back at any time if needed.

(a) (b) (c)

Figure 6.5: The different view of prototype: (a) Verb categories (first level). (b) Key verbs (second level). (c) Verb variations (third level).

Figure 6.5 shows the phases of the verb selection process. Rather than showing the ontology structure, which would make it difficult for the user to find the desired verbs, the user interface shows verbs organized according to a three-level folder structure: the first level (Figure 6.5(a)) contains broad categories (e.g. *speech actions*); the second level (Figure 6.5(b)) contains key verbs for the category (e.g. *speak* or *talk*); the third level (Figure 6.5(c)) contains subtler variations of the verbs from the second level (e.g. *whisper*). A verb is selected by dragging the corresponding stamp on the envelop.

Once the verb has been chosen, the role filler selection phase takes place, as shown in Figure 6.6. Six paper sheets in different colors, one for each role (Figure 6.6(a)), appear from behind the envelop. This view serves as an overview of the given fact, where the user can see the verb and its roles at a glance; the sheets can be moved on the screen to better explore their contents. This view is also used to show to a user someone else's facts.

In order to *edit* a sheet's content, the user has to tap on it. The main screen for the role filler selection process is represented in Figure 6.6(b). Role fillers are represented as stickers (containing a picture or a label). A set of suggested role fillers is placed in the envelop (exploiting the recommendation service provided by *Wanteat*); the user can either pick one of them, or search something in the *Wanteat* domain. The search window provides also the option of inserting a custom label and/or picture and/or a short description (see Figure 6.6(c)) in case the desired object is not present in *Wanteat* system. However, in this case the object is not tied to the ontology and will not be interpreted by the system.

When the user has finished editing the fact, he can submit it by clicking on the send button at the top right of the screen. He will then get back a list of pertinent facts, which he can view one by one.

(a) (b) (c)

Figure 6.6: The different view of prototype: (a) An empty phrase, (b) Suggestions in the envelop, (c) Creation of a new object

6.2.2 Server-Side Architecture

In order to achieve its goals, *Telleat* exploits both services of the *Wanteat* system and its own modules.

Figure 6.7 shows the server-side architecture, where services and modules that are Telleat-specific are distinguished by a thick black border. The figure also distinguishes three different interaction threads, numbered 1, 2 and 3, between the client application and the server.

Interaction 1 happens when the client application needs the available verbs. Since these are stored as an added part of the domain ontology, *Telleat*'s *Coordination Manager* dispatches the query to the Fact Repository Module, that in turn queries the *Ontology & DB Manager* for the list of available verbs, and maps them to the three-level hierarchy in the client.

Interaction 2 happens when the client has to suggest role fillers to the

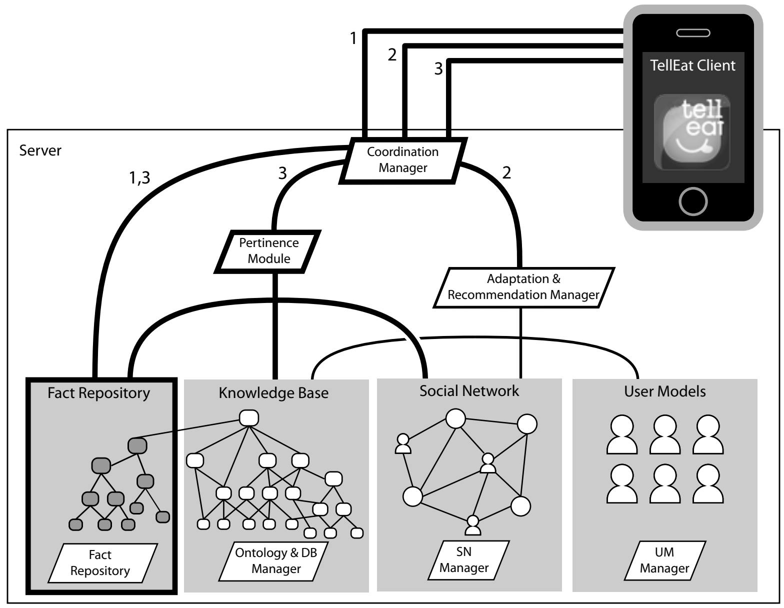

Figure 6.7: Server-side architecture of Telleat

user. This is achieved by using the support from the *Adaptation & Recommendation Manager* in *Wanteat*, as well as by taking into account other things, such as a role type, other facts inserted by the user, etc.

Interaction 3 happens when the client submits a newly inserted fact. In this case, the fact is sent for storage to the *Fact Repository* module. After the fact has been inserted, its pertinence with other stored facts is computed by the *Pertinence Module*. Facts with a pertinence value above a certain threshold are then sent back to the client, to be shown to the user.

Fact Repository.

Facts in *Telleat* are represented both as instances of a verb in the OWL verbs ontology and as records in a relational MySql database, storing other information as creation date, ownership, notes, custom labels and pictures. The Fact Repository provides a homogeneous access interface to the facts synthesizing a uniform representation from these two heterogeneous data sources. In particular the D2RQ tool is used to view a part of a relational database as a set of OWL individuals [15] to provide a uniform virtual data source that can be directly queried in SPARQL.

Pertinence Module.

The pertinence module computes the pertinence between facts according to the measure in Subsection 3.3:

$$\text{PERT}(f,g) = \alpha_0\text{SP} + \Sigma_{i=1}^{m}\alpha_i\text{SRF}_i + \beta\text{COLOC} \tag{6.1}$$

To reduce redundancy, when calculating the similarity of role fillers, we consider only the pairs with maximum similarity values for each role filler. This allows to take into account only meaningful conceptual distances, not any type of vague resemblance between role fillers.

6.2.3 Evaluation

Goals of the experiment

In this section we describe a simple preliminary evaluation which we conducted in order to evaluate the performance of our pertinence measure. Our main goal was to find out if the facts retained pertinent to a given fact by the system, are also viewed as pertinent by the users, since this would mean that the suggestions of our system could be interesting for users.

Description of the experiment

We recruited a total of twenty subjects among our contacts and colleagues, according to an availability sampling strategy.[7] All subjects were native

[7] Even though non-random samples are not statistically representative, they are often used in psychology research and usability testing, during early evaluation phases.

Italian speakers.

The test consisted of 5 identical steps. In each step, the subject of the test was given one primary fact and a list of seven secondary facts. We chose the secondary facts from a range of facts having different values of pertinence with the primary fact according to our system. For example, for the primary fact

"*At the party, Sonia and Lea shared a piece of cake made with Fuji Apples.*",

the secondary facts could have been

"*Sonia brought a cake made with Fuji Apples to a party.*",

as well as

"*Dan tastes a cheese in his local store.*" or

"*Fred buys a book for his girlfriend's birthday.*".

Also, the secondary facts were always presented in the random order. For each of the secondary facts, the subject was asked to assign the values on the 4-point scale from 0 to 3 (0 meaning not pertinent at all, 3 meaning very pertinent) depending on their perception of the pertinence of the secondary fact with the primary fact. Hence, each subject had to evaluate a total of 35 facts.

Results and discussion

Given the subjective nature of pertinence evaluation by the users (context awareness, subjective importance of different roles, etc.) in a social context, the correct detection of the pertinent facts by the system is more important and interesting than the classification of non-pertinent facts. In a social system users expect to see positive results and the links between them.

To this aim, we decided to set a threshold value of PERT = 2.0 above which the secondary facts are considered pertinent, hence interesting, for users. This meant that for different primary facts the system offered between 1 and 4 pertinent secondary facts. These facts were considered pertinent by the users in 75% of the cases. When we raised the threshold to PERT = 2.25, the percentage of the facts perceived pertinent by the users was 76%.

This shows a satisfactory level of performance of our pertinence measure. On the other hand, while performing the evaluation, we learned a whole lot about human behavior in assessing pertinence between events. Some of the users were looking for cause-effect relationships, some were giving higher importance to friendship relationships between protagonists, some were valuing more the events etc. These findings provided us with valuable directions for future research and for fine tuning our application.

7

Conclusions

This thesis is a contribution toward the goal of enhancing real, daily life objects with the capability to interact with users, and with each other, thereby exhibiting intelligent and social abilities (*SWIT* or Social Web of Intelligent Things, see Chapter 1).

According to *SWIT principles*, things can interact with people in a natural, personalized and bidirectional way. They know how to manage and exchange information with people and others things and, last but not least, they may have social abilities, such as the capability to befriend other things.

In a *SWIT*, a user on the move can interact with real things in a natural and playful way with his mobile device. He can perform several actions on them, which produce user-generated contents. The history of interaction and the generated contents are then visible in his virtual space (*i.e.* on the web) maintaining a *continuum of experience* of interaction between the user and *SWIT*-things.

Wanteat Video (see Section 6.1) proposes this type of experience, allowing users on the move to discover interesting content related to those things that surround him. He can view videos related to a particular thing and the opinions of other users about it. Moreover, he may explore other contents by means of the "wheel", which are connected to his current focus by social or ontological relations. The user has the possibility to add comments and votes on the avatars of real things that he previously met, to store his experience with them, and share useful information with others.

Some of these user actions are very structured (*i.e.* likes and votes) and machine processable, while others are very unstructured (*i.e.* comments), and therefore difficult to use as information for user profiling, recommendation and content aggregation of presentation.

For this reason in Chapter 3, we have introduced a new type of user-generated content, called *facts*. Facts are simple sentences, created by users according to a structured and guided approach, composed by a predicate and a set of role fillers. Role fillers play a role within the action (*i.e.* *subject* of the action or specification of the *place* where the action takes place) and the user can choose fillers by selecting them from entities in an ontology.

Facts are structured information but at the same time they allow the user a certain freedom of expression. In order to enable facts, we have extended the *Wanteat* iPhone application with *Telleat*, an application that allows users to create a *fact* in a playful and guided way. Users can talk about objects in *Wanteat* ontologies and all facts are stored in the *Wanteat* server.

This allows *Wanteat* to select and rank interesting contents according to the user model and the system knowledge. We propose that it also may show interesting facts previously stored, or also contents described by interesting facts (because facts are their captions or advanced tags). In fact, using *facts* as "advanced tags" to describe and add information to complex contents (*i.e.* videos or images), the system may find correlations among them that simple tagging does not allow.

The system can also aggregate facts according to their pertinence. The pertinence between two facts is given by the weighted sum of *semantic similarity of predicates*, *semantic similarity of role fillers* and colocation, as an expression of spatial-temporal contiguity.

The semantic similarity measure we propose for this task is based on *conceptual specificity*, which measures how much a certain concept is relevant in a given context, and on *conceptual distance*, which introduces different edge lengths in the ontology graph according to specificity. The evaluation of our measure shows an improvement over Leacock and Chodorow's distance [35]. and it has been proposed in the future work on user modeling [21].

A long-term goal of this research is to make our system able not only to create a set of interesting pertinent facts, but also to aggregate them by means of *storytelling mechanisms* and *artificial intelligence strategies* in narratively pertinent clusters. In the literature [37][30] some strategies has been proposed to create associations between facts. Ultimately, the result is a "collective story" formulated by selected users-generated content.

We found that an important factor to take in account for this task is the spatial-temporal content where actions take place. We therefore introduced in our ontology the notion of *STOB* (spatial-temporal object, see Chapter 3) that aims at capturing this concept. Several *STOBs* where actions take place are actually "events" (either public, such as concerts or fairs, or private, such as dinner, parties or day trips). Adding this notion to the *Wanteat* ontologies led us to investigate what is the difference between the recommendation of events with respect to other content types. In particular, we inquired which additional factors may influence users when selecting an event to join.

We investigated several score factors that may influence the events recommendation: (a) *thematic and type interest*, a factor calculate by combining the basic information of an event (the *content features*) and the user model, (b) *reachability* which measures the feasibility of user participation (*i.e.* propensity to move from user position) taking in account the *context features* of event (temporal and spatial properties), (c) *average rating* and *friend participation* that consider the *social features* of the event.

We then discovered that themes and type play a crucial role, but not in equally way. In particular, themes carry more information and influence users more. Moreover, the additional information about event (reachability by the user, friends' participation and event rating) improves the performance of recommendation in different ways depending to the importance of factors. The importance of factors may not be declared by users, because they do not correspond to their real rating and attendance behavior. For this reason, we studied fine-tuning weights and the best event recommendation was obtain by dynamic weights, based on values of themes and type scores.

Our long-temporal goal aims to offer users a novel approach to content navigation, which is narrative-based.

The *Wanteat* framework supports mainly two types of navigation with the aim of allowing users to discover potentially interesting things. The most important in the framework is "the wheel" that valorizes the different relationships between things (*i.e.* food-related objects or geographic places) and people. A second kind of navigation, often used in popular social networks[1], uses a user-generated content, the *tag* (or its evolution the *hashtag*). In this case, the user may discover new information passing from one thing to another through a common tag.

Facts, on the other hand, combine things and people without using ontological properties or social relations. Each fact is an aggregator of ontological entities, people and events with a new kind of relation, the "action", which is essentially narrative. The aggregation of many facts similar to each other, that contain the same elements, can create a new exploring path for user, which is semantically and narratively more meaningful than a tag-based one. Moreover, facts relate ontological entities of different ontologies thanks to users' contributions (a successful strategy for some social platforms, such as Wikipedia[2]). These links between ontological entities allow to make inferences and associations for creating a "knowledge map" above the descriptions offered by ontologies. For example, the *Knowledge Graph* of Google[3] consid-

[1]Instagram `http://instagram.com/`, Flickr `http://www.flickr.com/`, Twitter `https://twitter.com/`

[2]Wikipedia `http://it.wikipedia.org/`

[3]Official Blog of Google `http://googleblog.blogspot.co.uk/2012/05/introducing-knowledge-graph-things-not.html`

ers strings as things, connected to others through their semantic meaning. This allows to disambiguate, for example, a word in a user query by the context in which it is expressed.

Last but not least, collecting facts provided by people on specific things and locations can create a history of that thing or place that is at the same time collective and personal. The suggestion we propose here is that, by making real, physical objects totemic repositories of content, and by shaping such content in a narrative form, we may one day physically walk among our stories.

8 Acknowledgements

The project has been partially supported by Regione Piemonte, grant ICT Converging Technologies 2007.

9

References

Papers by this author

[1] Elisa Chiabrando, Fabrizio Franceschi, Roberto Furnari, and Fabiana Vernero. Questing for videos: a user-centric model for exploring the gastronomy domain. In *Proc. of NEM Summit 2011, Turin, Italy*, 2011.

[2] Elisa Chiabrando, Roberto Furnari, Pierluigi Grillo, Silvia Likavec, and Ilaria Lombardi. Dynamic interface reconfiguration based on different ontological relations. In *Proc. of the 14th International Conference on Human-Computer Interaction, HCII '11*, LNCS, pages 538–547. Springer, 2011.

[3] Elisa Chiabrando, Silvia Likavec, Ilaria Lombardi, Claudia Picardi, and Daniele Daniele Theseider Dupré. Semantic similarity in heterogeneous ontologies. In *Proc. of the 22nd ACM Conference on Hypertext and Hypermedia, Hypertext '11*, pages 153–160. ACM, 2011.

[4] PIEMONTE Team. Interacting with a social web of smart objects for enhancing tourist experiences. In Matthias Fuchs, Francesco Ricci, and Lorenzo Cantoni, editors, *Information and Communication Technologies in Tourism 2012*. Springer Vienna, 2012.

[5] PIEMONTE Team. Wheeling around with wanteat: exploring mixed social networks in the gastronomy domain. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces, IUI '12*, New York, NY, USA, 2012. ACM.

[6] PIEMONTE TEAM. Wanteat: interacting with social networks of smart objects for sharing and promoting cultural heritage. In *UMAP Workshops*, volume 872 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012.

[7] PIEMONTE Team. WantEat: interacting with social networks of intelligent things and people in the world of enogastronomy. In *Interacting with Smart Objects Workshop 2011*, 2011.

[8] PIEMONTE TEAM. Wanteat: interacting with social networks of smart objects for sharing cultural heritage and supporting sustainability. In *ECAI'12*, pages 1005–1006, 2012.

[9] PIEMONTE TEAM. Interacting with social networks of intelligent things and people in the world of gastronomy. *ACM Transaction on Interactive Intelligent Sytems (TIIS)*, in press.

References

[10] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734 – 749, june 2005.

[11] 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, January 2005.

[12] Ofer Arazy, Nanda Kumar, and Bracha Shapira. A theory-driven design framework for social recommender systems. *Journal of the Association for Information System, AIS*, 11(9), 2010.

[13] Sidahmed Benabderrahmane, Malika Smail-Tabbone, Olivier Poch, Amedeo Napoli, and Marie-Dominique Devignes. Intelligo: a new vector-based semantic similarity measure including annotation origin. *BMC Bioinformatics*, 11:1–16, 2010.

[14] Giulia Biamino, Pierluigi Grillo, Ilaria Lombardi, Alessandro Marcengo, Amon Rapp, Rossana Simeoni, and Fabiana Vernero. "The Wheel": an innovative visual model for interacting with a social web of things. In *Visual Interfaces to the Social and Semantic Web, VISSW '11*, 2011.

[15] Christian Bizer and Andy Seaborne. D2RQ - Treating Non-RDF Databases as Virtual RDF Graphs. In *3rd International Semantic Web Conference, ISWC '04 (Poster)*, 2004. http://www4.wiwiss.fu-berlin.de/bizer/d2rq/.

[16] Nicoletta Di Blas, Franca Garzotto, Paolo Paolini, and Amalia Sabiescu. Digital storytelling as a whole-class learning activity: Lessons from a three-years project. In *Proc. of the 2nd Joint International Conference on Interactive Digital Storytelling ICIDS '09*, volume 5915 of *LNCS*, pages 14–25. Springer, 2009.

[17] A. Budanitsky and G. Hirst. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, 32(1):13–47, March 2006.

[18] Yiwei Cao, Ralf Klamma, and Andrea Martini. Collaborative storytelling in the web 2.0. In *Proc. of the 1st International Workshop on Story-Telling and Educational Games, STEG-2008*, 2008.

[19] Francesca Carmagnola, Federica Cena, Luca Console, Omar Cortassa, Cristina Gena, Anna Goy, Ilaria Torre, Andrea Toso, and Fabiana Vernero. Tag-based user modeling for social multi-device adaptive guides. *User Model. User-Adapt. Interact.*, 18(5):497–538, 2008.

[20] Francesca Carmagnola, Federica Cena, Luca Console, Omar Cortassa, Cristina Gena, Anna Goy, Ilaria Torre, Andrea Toso, and Fabiana Vernero. Tag-based user modeling for social multi-device adaptive guides. *User Modeling and User-Adapted Interaction*, 18:497–538, 2008.

[21] Federica Cena, Silvia Likavec, and Francesco Osborne. Property-based interest propagation in ontology-based user model. In Judith Masthoff, Bamshad Mobasher, MichelC. Desmarais, and Roger Nkambou, editors, *User Modeling, Adaptation, and Personalization*, volume 7379 of *Lecture Notes in Computer Science*, pages 38–50. Springer Berlin Heidelberg, 2012.

[22] Rudi L. Cilibrasi and Paul M.B. Vitanyi. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19, 2007.

[23] Luca Console, Ilaria Lombardi, Claudia Picardi, and Rossana Simeoni. Toward a social web of intelligent things. *AI Commun.*, 24(3):265–279, 2011.

[24] Chris Cornelis, Xuetao Guo, Jie Lu, and Guanquang Zhang. A Fuzzy Relational Approach to Event Recommendation. *Artificial Intelligence*, pages 2231–2242, 2007.

[25] Nuno Correia, Luís Alves, Helder Correia, Luis Romero, Carmen Morgado, Luís Soares, José C. Cunha, Teresa Romão, A. Eduardo Dias, and Joaquim A. Jorge. InStory: a System for Mobile Information Access, Storytelling and Gaming Activities in Physical Spaces. In *Proc. of the ACM SIGCHI International Conference on Advances in computer entertainment technology*, ACE '05, pages 102–109. ACM, 2005.

[26] Francisco M. Couto, Mrio J. Silva, and Pedro M. Coutinho. Measuring semantic similarity between gene ontology terms. *Data and Knowledge Engineering*, 61(1):137 – 152, 2007.

[27] Martyn Dade-Robertson. Visual scenario representation in the context of a tool for interactive storytelling. In *Proc. of the 4th International Conference on Virtual Storytelling ICVS '07*, volume 4871 of *LNCS*, pages 3–12. Springer, 2007.

[28] Simon Dooms, Toon De Pessemier, and Luc Martens. A user-centric evaluation of recommender algorithms for an event recommendation system. In Alexander Felfernig, Li Chen, Monika Madl, Martijn Willemsen, Dirk Bollen, and Michael Ekstrand, editors, *Proceedings of the RecSys 2011 : Workshop on Human Decision Making in Recommender Systems (Decisions@RecSys'11) and User-Centric Evaluation of Recommender Systems and Their Interfaces - 2 (UCERSTI 2) affiliated with the 5th ACM Conference on Recommender Systems (RecSys 2011)*, pages 67–73. Ghent University, Department of Information technology, 2011.

[29] Susan T. Dumais. Latent semantic analysis. *Annual Review of Information Science and Technology*, 2004.

[30] Joeli Takala Eero Hyvnen, Tuomas Palonen. Narrative semantic web — case national finnish epic kalevala. In *Poster Papers, the 7th Extended Semantic Web Conference, Heraklion, Greece*, June 2010.

[31] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database.* The MIT Press, Cambridge, MA; London, 1998.

[32] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, pages 1606–1611, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[33] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[34] J.J. Jiang and D.W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of the International Conference on Research in Computational Linguistics*, pages 19–33, 1997.

[35] C. Leacock and M. Chodorow. *Combining local context and WordNet similarity for word sense identification*, pages 305–332. In C. Fellbaum (Ed.), MIT Press, 1998.

[36] Danielle Hyunsook Lee. Pittcult: trust-based cultural event recommender. In *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08, pages 311–314, New York, NY, USA, 2008. ACM.

[37] Silvia Likavec, Ilaria Lombardi, Alberto Nantiat, Claudia Picardi, and Daniele Theseider Dupré. Threading facts into a collective narrative world. In *Proc. of the 3rd Joint International Conference on Interactive Digital Storytelling, ICIDS '10*, volume 6432 of *LNCS*, pages 86–97. Springer-Verlag, 2010.

[38] Dekang Lin. An information-theoretic definition of similarity. In *Proc. of the 15th International Conference on Machine Learning ICML '98*, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[39] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In Ricci et al. [50], pages 73–105.

[40] LeO'Reilly Media, editor. *RESTful Web Services*. The MIT Press, Cambridge, MA; London, 2007.

[41] Einat Minkov, Ben Charrow, Jonathan Ledlie, Seth Teller, and Tommi Jaakkola. Collaborative future event recommendation. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 819–828, New York, NY, USA, 2010. ACM.

[42] Janet Horowitz Murray. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace.* The Free Press, New York, NY, USA, 1997.

[43] Frank Nack, Abdallah El Ali, Philo Van Kemenade, Jan Overgoor, and Bastiaan Van der Weij. A story to go, please. In *Proceedings of the Third joint conference on Interactive digital storytelling*, ICIDS'10, pages 74–85. Springer-Verlag, 2010.

[44] Michael J. Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web: methods and strategies of web pesonalization. Volume 4321 of Lecture Notes in Computer Science*, pages 325–341. Springer-Verlag, 2007.

[45] Ted Pedersen. Information content measures of semantic similarity perform better without sense-tagged text. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 329–332, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[46] Toon De Pessemier, Sam Coppens, Kristof Geebelen, Chris Vleugels, Stijn Bannier, Erik Mannens, Kris Vanhecke, and Luc Martens. Collaborative recommendations with content-based filters for cultural activities via a scalable event distribution platform. *Multimedia Tools and Applications*, 58(1):167–213, 2012.

[47] Roy Rada, Hafedh Mili, Ellen Bicknell, and Maria Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems Management and Cybernetics*, 19(1):17–30, 1989.

[48] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *In Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, 1995.

[49] Philip Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, 1998.

[50] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.

[51] Ray Richardson and Alan F. Smeaton. Using wordnet in a knowledge-based approach to information retrieval. Technical report, 1995.

[52] Alan F. Smeaton and Ian Quigley. Experiments on using semantic distances between words in image caption retrieval. In *Proc. of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR '96*, pages 174–180. ACM, 1996.

[53] Marc Spaniol, Ralf Klamma, Nalin Sharda, and Matthias Jarke. Web-based learning with non-linear multimedia stories. In *Proc. of the 5th International Conference on Advances in Web Based Learning, ICWL '06*, volume 4181 of *LNCS*, pages 249–263, 2006.

[54] Michael Sussna. Word sense disambiguation for free-text indexing using a massive semantic network. In *Proceedings of the second international conference on Information and knowledge management*, CIKM '93, pages 67–74, New York, NY, USA, 1993. ACM.

[55] Carl Taylor. Semantic modeling and navigation in a museum hypermedia system, 1995. PhD Thesis, University of Glamorgan.

[56] Raphaël Troncy, Bartosz Malocha, and André T. S. Fialho. Linking events with media. In *Proceedings of the 6th International Conference on Semantic Systems*, I-SEMANTICS '10. ACM, 2010.

[57] Douglas Tudhope and Carl Taylor. Navigation via similarity: Automatic linking based on semantic closeness. *Information Processing and Management*, 33(2):233–242, 1997.

[58] Patricia Victor, Martine De Cock, and Chris Cornelis. Trust and recommendations. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 645–675. Springer, 2011.

[59] Xiang-jun Wang, Swathi Mamadgi, Atit Thekdi, Aisling Kelliher, and Hari Sundaram. Eventory – an event based media repository. In *Proceedings of the International Conference on Semantic Computing*, ICSC '07, Washington, DC, USA, 2007. IEEE Computer Society.

[60] Utz Westermann and Ramesh Jain. Toward a common event model for multimedia applications. *IEEE MultiMedia*, 2007.

[61] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proc. of the 32nd Annual Meeting on Association for Computational Linguistics*, pages 133–138, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

[62] Gian Piero Zarri. *Representation and Management of Narrative Information: Theoretical Principles and Implementation*. Advanced Information and Knowledge Processing. Springer, 2010.