Personalized Information Retrieval in Context by Exploiting Semantic Knowledge and Implicit User Feedback

David Jordi Vallet Weadon

Thesis advisor

Pablo Castells Azpilicueta

Department of Computer Science
Escuela Politécnica Superior
Universidad Autónoma de Madrid

Submitted for the Degree of Doctor of Philosophy
in the subject of
Computer Science
at the Universidad Autónoma de Madrid
May 2008
Abstract

Personalization in information retrieval aims at improving the user’s experience by incorporating the user subjectivity into the retrieval methods and models. The exploitation of implicit user interests and preferences has been identified as an important direction to enhance current mainstream retrieval technologies and anticipate future limitations as worldwide content keeps growing, and user expectations keep rising. Without requiring further efforts from users, personalization aims to compensate the limitations of user need representation formalisms (such as the dominant keyword-based or document-based) and help handle the scale of search spaces and answer sets, under which a user query alone is often not enough information for the system to provide effective results. However, the general set of user interests that a retrieval system can learn over a period of time, and bring to bear in a specific retrieval session, can be fairly vast, diverse, and to a large extent unrelated to a particular user search in process. This means that even on the basis of correctly learned user preferences, the system could make wrong guesses or get intrusive. Rather than introducing all user preferences en bloc, an optimum search adaptation could be achieved if the personalization system was able to select only those preferences which are pertinent to the ongoing user actions. In other words, although personalization alone is a key aspect of modern retrieval systems, it is the application of context awareness into personalization what can really produce a step forward in future retrieval applications.

Context modeling has been long acknowledged as a key aspect in a wide variety of problem domains, among which Information Retrieval is a prominent one. In this work, we focus on the representation of live retrieval user contexts, based on implicit feedback techniques. The particular notion of context considered in this thesis is defined as the set of themes under which retrieval user activities occur within a unit of time.

Our proposal of contextualized personalization is based on the semantic relation between the user profile and the user context. Only those preferences related to the current context should be used, disregarding those that are out of context. The use of semantic-driven representations of the domain of discourse, as a common, enriched representational ground for content meaning, user interests, and contextual conditions, is proposed as a key enabler of effective means for a) a rich user model representation, b) context acquisition at runtime and, most importantly, c) the discovery of semantic connections between the context and concepts of user interest, in order to filter those preferences that have chances to be intrusive within the current course of user activities.
Acknowledgements

I would like to thank to all the persons that have collaborated on the consecution of this PhD over these four years. I feel lucky to have had the chance to make this research with such a good team, which surely allowed me to progress on my research at a solid pace. Pablo, thank you for your hard work, for giving me the opportunity to work with this amazing team, and thank you for your mentorship over these years. I would like to thank current and past members of this team, especially to Miguel Corella, Chema Fuentes, Alvaro Valera, Iván Cantador, Alejandro Bellogín and Miriam Fernández. Thank you all for your company and guidance, and for making these years so enjoyable. I would also like to thank reviewers Yannis Avrithis, Phivos Mylonas and Daniel Olmedilla, whose detail reading and feedback helped on the improvement of the final version of this PhD.

During my PhD period, I had the opportunity to visit three different research labs, in where I received a very important part of my learning process. I want to thank the members of the Ebiquity group, at UMBC, the MIR Group, at the University of Glasgow and Yahoo! Research, at Barcelona. Thank you for your warm welcome, and specially to Joemon Jose and Hugo Zaragoza for the opportunity they gave me to feel part of their group.

Lastly, I want to thank all those who have supported me with their sincere friendship during these years. Thank you for not losing touch over these years, and for being patient with me, even with my continuous trips or stressful periods. Special thanks to Oscar, for being my restless and tolerant undergraduate partner, to my brother, and to Maria and my father, my two basic pillars, for putting up with all this, and letting me lean on them all these years.

To my mother
# Contents

Abstract ...................................................................................................................... i

Acknowledgements ................................................................................................. ii

Contents .................................................................................................................... iii

List of Figures ............................................................................................................ vii

List of Tables .............................................................................................................. ix

1 Introduction ............................................................................................................ 1

1.1 Motivation .......................................................................................................... 1

1.2 Personalization in Context .............................................................................. 4

1.2.1 Semantics and Personalization .................................................................. 4

1.2.2 User Context Modeling and Exploitation .............................................. 6

1.3 Contributions ................................................................................................... 9

1.4 Outline ............................................................................................................. 10

2 State of the Art ................................................................................................... 13

2.1 Personalized Information Retrieval ............................................................ 13

2.1.1 Modeling Users ...................................................................................... 16

2.1.2 User Profile Representation ................................................................. 17

2.1.3 User Profile Learning ........................................................................... 23

2.1.4 User Profile Exploitation ..................................................................... 28

2.1.5 Personalization in Working Applications ............................................ 38

2.2 Context Modeling for Information Retrieval .............................................. 40

2.2.1 Concept of Context ............................................................................... 43

2.2.2 Context Acquisition and Representation ........................................... 45

2.2.3 Context Exploitation ............................................................................ 46

3 A Personalized Information Retrieval Model Based on Semantic Knowledge .... 49
3.1 Ontology-based User Profile Representation .............................................................. 50
3.2 A Semantic Approach for User Profile Exploitation .................................................. 53
4 Personalization in Context ................................................................................................ 59
4.1 Notation ....................................................................................................................... 61
4.2 Preliminaries ............................................................................................................... 62
4.3 Semantic Context for Personalized Content Retrieval ................................................ 64
4.4 Capturing the Context ................................................................................................. 66
4.5 Semantic Expansion of Context and Preferences ........................................................ 67
  4.5.1 Spreading Activation Algorithm ......................................................................... 68
  4.5.2 Comparison to Classic CSA ................................................................................ 74
  4.5.3 Semantic Preference Expansion .......................................................................... 75
4.6 Contextual Activation of Preferences .......................................................................... 76
4.7 Contextualization of Preferences ................................................................................. 77
4.8 Long and Short Term Interests in the State of the Art ................................................ 78
4.9 An Example Use Case ................................................................................................. 80
5 Experimental work ............................................................................................................ 87
  5.1 Evaluation of Interactive IR Systems: An Overview .................................................. 87
    5.1.1 User Centered Evaluation .................................................................................... 88
    5.1.2 Data Driven Evaluation ....................................................................................... 91
    5.1.3 Evaluation Metrics .............................................................................................. 92
    5.1.4 Evaluation Corpus ............................................................................................... 93
5.2 Experimental Setup ..................................................................................................... 94
5.3 Scenario Based Testing: a Data Driven Evaluation .................................................... 97
    5.3.1 Evaluation Tool ................................................................................................... 97
    5.3.2 Methodology ..................................................................................................... 102
    5.3.3 Results ............................................................................................................... 103
5.4 User Centered Evaluation .......................................................................................... 105
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4.1</td>
<td>Evaluation Tool</td>
<td>105</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Methodology</td>
<td>107</td>
</tr>
<tr>
<td>5.4.3</td>
<td>Results</td>
<td>111</td>
</tr>
<tr>
<td>6</td>
<td>Conclusions</td>
<td>113</td>
</tr>
<tr>
<td>6.1</td>
<td>Summary and Achieved Contributions</td>
<td>113</td>
</tr>
<tr>
<td>6.1.1</td>
<td>Personalization Framework Based on Semantic Knowledge</td>
<td>113</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Personalization in Context</td>
<td>114</td>
</tr>
<tr>
<td>6.1.3</td>
<td>User and Context Awareness Evaluation</td>
<td>115</td>
</tr>
<tr>
<td>6.2</td>
<td>Discussion and Future Work</td>
<td>116</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Context modeling</td>
<td>117</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Semantic resources</td>
<td>119</td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>121</td>
</tr>
<tr>
<td>Published Work</td>
<td></td>
<td>131</td>
</tr>
<tr>
<td>Appendices</td>
<td></td>
<td>133</td>
</tr>
<tr>
<td>A.</td>
<td>Detailed Results for the Scenario Based Experiments</td>
<td>134</td>
</tr>
<tr>
<td>B.</td>
<td>User Centered Evaluation Task Descriptions</td>
<td>138</td>
</tr>
</tbody>
</table>
List of Figures

Figure 2.1. Query operation example for a two dimension projection. ............................. 29
Figure 2.2. Example of user profile based on logic term operators................................. 31
Figure 2.3. Typical schema of document scoring on personalized retrieval systems....... 34
Figure 3.1. User preferences as concepts in an ontology.................................................. 52
Figure 3.2. Links between user preferences and search space........................................... 54
Figure 3.3. Visual representation of metadata and preference's vector similarity.......... 55
Figure 3.4. Construction of the user and the content concept-weighted vectors............. 56
Figure 4.1. Simple version of the spreading activation algorithm........................................... 69
Figure 4.2. Example of preference expansion with the CSA algorithm............................ 70
Figure 4.3. Priority queue variation of the spreading activation algorithm......................... 71
Figure 4.4. Parameter optimized variation with priority queue of the spreading activation algorithm........................................................................................................... 73
Figure 4.5. Semantic intersection between preferences and context................................. 77
Figure 4.6. Characterization of concept drift in the contextualization algorithm.............. 78
Figure 4.7. A subset of domain ontology concepts involved in the use case....................... 81
Figure 4.8. Visual representation of the preference contextualization................................. 84
Figure 5.1. Example of a simulated situation, from (Borlund 2003).................................. 89
Figure 5.2. Different areas of performance for a precision and recall curve......................... 96
Figure 5.3. Main window of the scenario based evaluation tool......................................... 97
Figure 5.4. UI for editing complex relational query conditions........................................... 99
Figure 5.5. Interactive dialog for semantic query construction............................................. 99
Figure 5.6. Concept profile editor .................................................................................... 100
Figure 5.7. Examples of document snippets and text highlighting..................................... 101
Figure 5.8. Contextual and personalization information window ....................................... 101
Figure 5.9. Example of user centered task description. .............................................................. 103

Figure 5.10. Comparative performance of personalized search with and without contextualization. ................................................................................................................. 104

Figure 5.11. Comparative mean average precision histogram of personalized search with and without contextualization .............................................................................. 105

Figure 5.12. User centered evaluation tool. ............................................................................. 106

Figure 5.13. User preference edition ......................................................................................... 107

Figure 5.14. Task description for task 1: News about agreements between companies. 108

Figure 5.15. Relevance assessment UI. .................................................................................... 110

Figure 5.16. Comparative performance of personalized search with and without contextualization. ................................................................................................................. 112
### List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Overview of personalized information retrieval systems.</td>
<td>16</td>
</tr>
<tr>
<td>2.2</td>
<td>Overview of term-based user profile representation systems.</td>
<td>19</td>
</tr>
<tr>
<td>2.3</td>
<td>Concept-based user profile representation in personalized retrieval systems.</td>
<td>20</td>
</tr>
<tr>
<td>2.4</td>
<td>Overview of usage history based user profile representation in personalized retrieval systems.</td>
<td>22</td>
</tr>
<tr>
<td>2.5</td>
<td>Overview of explicit feedback learning in personalized systems.</td>
<td>25</td>
</tr>
<tr>
<td>2.6</td>
<td>Overview of implicit feedback learning in personalized systems.</td>
<td>27</td>
</tr>
<tr>
<td>2.7</td>
<td>Overview of hybrid feedback learning in personalized systems.</td>
<td>28</td>
</tr>
<tr>
<td>2.8</td>
<td>Classification of document scoring exploitation in personalized systems.</td>
<td>35</td>
</tr>
<tr>
<td>2.9</td>
<td>Overview of context-aware retrieval systems.</td>
<td>43</td>
</tr>
<tr>
<td>4.1</td>
<td>Spreading activation algorithm optimization parameters.</td>
<td>74</td>
</tr>
<tr>
<td>4.2</td>
<td>Example of user Profile: Sue's preferences.</td>
<td>81</td>
</tr>
<tr>
<td>4.3</td>
<td>Example of propagation of weights through semantic relations.</td>
<td>82</td>
</tr>
<tr>
<td>4.4</td>
<td>Example of Context vector.</td>
<td>82</td>
</tr>
<tr>
<td>4.5</td>
<td>Example of expanded context vector.</td>
<td>83</td>
</tr>
<tr>
<td>4.6</td>
<td>Example of extended user preferences.</td>
<td>83</td>
</tr>
<tr>
<td>4.7</td>
<td>Example of contextualized user preferences.</td>
<td>84</td>
</tr>
<tr>
<td>5.1</td>
<td>Summary of complete evaluation systems.</td>
<td>88</td>
</tr>
<tr>
<td>5.2</td>
<td>Results on Mean Average Precision (MAP) for each of the three evaluated retrieval models.</td>
<td>112</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Motivation

The size and the pace of growth of the world-wide body of available information in digital format (text and audiovisual) constitute a permanent challenge for content retrieval technologies. People have instant access to unprecedented inventories of multimedia content world-wide, readily available from their office, their living room, or the palm of their hand. In such environments, users would be helpless without the assistance of powerful searching and browsing tools to find their way through. In environments lacking a strong global organization (such as the open WWW), with decentralized content provision, dynamic networks, etc., query-based and browsing technologies often find their limits.

Take as an example a user who enters the query “search library” into a typical Web search engine, such as Google, Yahoo! or MSN search. Taking the query alone, we may think the user is looking for an online service for book location in e.g. some local library, bookstores, or digital libraries. But the intention of this query could also be related, for instance, to finding computer programming libraries supporting content search and retrieval functionalities. Such an ambiguous query, which by itself alone does not provide enough information to properly grasp the user’s information need, is an example where personalization capabilities show their usefulness. While mainstream Web search engines return the same results to all users\(^1\), a personalized system adapts the search results to the users’ interests. In the example, the second interpretation (programming library) might seem more likely, and the first (book search) a bit far-fetched. Interestingly though, testing the example in Google, the results happen to be more related to the first meaning of the query: Web sites like wordcat (a book and local library locator) or the Google book search service appear at the top of the ranking.

\(^1\) Nowadays there are some incipient exceptions. For instance, Google is currently applying a subtle personalization approach, which, according to the description given in corresponding US patent applications, uses the past usage history of the user in order to promote previously clicked results in similar past queries. The user’s country and language are also used for some simple adaptations.
Let’s now suppose there are two users with different interests using the Web search engine: one has an interest for computer programming and the other has an interest for science fiction literature. With this information at hand, it should be possible for a personalized search engine to disambiguate the original query “search library”. The first user should receive e.g. the Lucene and Terrier Java libraries (which support indexing and searching functionalities) in the top results. The second user should receive results about e.g. catalog search services for local and online libraries specialized in science fiction literature.

Now what if a user happens to share these two interests, e.g. a computer programmer who likes science fiction literature? If the personalization system applied all the preferences together, it may happen that the results neither fully satisfy one interest nor the other. Results based on both preferences may include for instance two average-quality science fiction online catalogs, written in java, and an Amazon.com page about “Java programming” under the “Science Fiction & Fantasy” category. These results are relevant to all the interests of the user in a too literal way, but it is unlikely the user will find them subjectively interesting in a particular realistic situation. The problem here is that user preferences, taken as a whole, are also ambiguous for the query at hand. The question then is whether and where it is possible to find further information to clarify the actual user’s intent. The solution explored in this thesis is to seek for such cues in the closer context of the current user situation (e.g. the task at hand).

This thesis thus elaborates on the research hypothesis that context applied to personalized retrieval can be exploited to discard interests that are not related to the current context of the user. For instance, if the user is at work, the preference for computer programming is more likely to be relevant, whereas the preference for science fiction literature could be more safely discarded and not used in the personalization process (i.e. this can be expected to be a good decision in most cases, that is, on average).

Another example of context source, which is explored in this work, is the implicit feedback information from the user, i.e. the contextual information implicitly provided by previous interactions of the user with the retrieval system, within the same search session. As an

---

2 In a simplification of the personalization process, we may assume for the sake of this example that the personalization system disambiguates the query by adding automatically extra terms. For instance, it would change the query to “java search library” or to “search science fiction library” for each user respectively. This example was elaborated based on results from the Google search engine. Note that results may vary with time.

3 http://lucene.apache.org/

4 http://ir.dcs.gla.ac.uk/terrier/

5 The personalized example can be simulated by the query “java search science fiction library”.
example, suppose that before the user issued the query “search library”, she opened a document related to science fiction, or input a query about the “Ender’s game” science fiction book. With this background information the system may infer that the relevant preferences in this particular situation are the ones related to science fiction literature, whereas the preference for computer programming has no clear relation with the current user focus, and can thus be discarded from the personalized processing step in this particular case. In both situations, the system is able at the same time to tackle the ambiguity of the search, and to select which background user interests matter in the current situation, achieving results that are relevant to both the user and her situation.

Personalized content access aims to alleviate the information overload and information need ambiguity problem with an improved Information Retrieval (IR) process, by using implicit user preferences to complement explicit user requests, to better meet individual user needs (Gauch et al. 2003; Haveliwala 2002; Jose and Urban 2006; Kobsa 2001). As illustrated in the previous example, the main motivation of personalized retrieval systems is that users often fail to accurately represent their information need, using no more than 3 keywords (Jansen et al. 1998), which often lead to ambiguous queries (Krovetz and Croft 1992). Needless to say, user queries rarely include the implicit interests of the user.

Personalization is being currently envisioned as a major research trend in the upcoming revolution of IR technologies, since classic IR tends to select the same content for different users on the same query, many of which are barely related to the user’s wish (Chen and Kuo 2000). Since the early days up to the latest progress in this area, personalization has been applied to different retrieval aspects, such as content filtering (Micarelli and Sciarrone 2004) and recommendation (Sheth and Maes 1993), content search (Jeh and Widom 2003), navigation (Lieberman 1995), or content presentation (Sakagami and Kamba 1997). Personalization also is relevant to many other research areas, such as education (Brusilovsky et al. 1998), digital libraries (Smeaton and Callan 2001), TV media (Aroyo et al. 2007), or tourism (Fink and Kobsa 2002), to name a few. Nowadays, major online services such as Google (Badros and Lawrence 2005; Zamir et al. 2005), Amazon.com (Smith et al. 2005) or Yahoo! (Kraft et al. 2005) are researching on personalization, in particular to improve their content retrieval systems.

One of the lessons learnt over the years, in particular with the practical initiatives, is that it is very difficult to achieve effective generic personalization solutions, without having considerable knowledge about the particular problem being addressed. Personalization approaches seemed to result in either a very specialized or a rather generic solution that provided very limited personalization capabilities. In order to address some of the limitations of these personalization
systems, researchers have looked to the new emerging area defined by the so-called context-aware applications and systems (Abowd et al. 1997).

Context-awareness has been long researched and successfully applied in a wide variety of fields, including mobile and pervasive computing (Chalmers 2004), image analysis (Thanos et al. 2007), computational linguistics (Finkelstein et al. 2002), or information retrieval (Bharat 2000; Kim and Chan 2003; White and Kelly 2006). Context in IR has also been subject to a wide range of interpretations and applications, ranging from desktop information (Dumais et al. 2003) to physical user location (Melucci 2005), recently visited Web pages (Sugiyama et al. 2004), or session interaction data (Shen et al. 2005b).

The research undertaken here lies at the confluence of context-awareness and personalization, and aims to develop personalized IR solutions that draw and combine advantage from the two areas. A personalization approach that is context-aware, i.e. a personalization in context approach, should be able to apply personalization in the different areas and retrieval aspects mentioned previously. And, at the same time, it should be aware of the context the user is in when performing a retrieval task. It should be able to “adapt the adaptation process” in order to provide a more effective and precise personalization. In this setting, this thesis focuses on three main areas: a) exploitation of domain knowledge, represented in a rich and accurate form, to enhance the capabilities and performance of personalization, by improving the representation of user preferences; b) acknowledge and cope with the dynamic aspects of implicit user preferences, which though stable, do not come into play in a monolithic way in practice, but relative to the user goals, state, ongoing actions, etc.; c) define a modular context modeling framework, on top of a personalization system, which captures the relative essence of user interests in a workable yet effective way, improving the performance and reliability of the base personalization system, and in particular, reducing the well-known potential intrusiveness of personalization techniques; d) test, evaluate and measure the improvement achieved by the personalization techniques and their context-based enhancement.

1.2 Personalization in Context

1.2.1 Semantics and Personalization

Three main areas or problems commonly need to be addressed in a personalization approach: the representation, acquisition, and exploitation of user profiles.

The user profile can be automatically acquired (or enriched) by monitoring of user’s interaction with the system, as long as the monitoring period is sufficient and representative of the user’s
preferences. User profile learning alone is a wide and complex area of research (Gauch et al. 2007), out of the scope and complementary to the problems addressed in this thesis, which focuses on the areas of user profile representation and exploitation.

Representing and exploiting user preferences in a formal way is not an easy task. User preferences are often vague (e.g. “I like sports”, “I like travelling”, “I like animals”), complex (e.g. “I like swimming, but only when it’s really hot”, “On rainy days, there’s nothing like going to the cinema”, “I like traveling to Africa, but only to countries with stable governments”), or even contradictory (e.g. “I don’t like sensationalist tabloids, but when I’m waiting for my doctor’s appointment, I like to take a peek at them for a while…”, “I like animals, but I cannot stand anything that resembles a rat”).

Typical solutions for user profile representation are based on statistical methods, where a user profile is represented as a bag of terms (Liu et al. 2004; Sakagami and Kamba 1997; Teevan et al. 2005). These approaches can be complemented with relations, such as correlation measures (Asnicar and Tasso 1997) or links to topic categories (Liu et al. 2004). However, terms cannot represent all the subtleties of the previous examples: 1) they are ambiguous, for instance, “Jaguar” can be related to an animal, a car brand, or to an Operative System, 2) their semantics are rather limited, for instance, an interest for “birds” in general is difficult to match to a document that is related to the “Woodpecker” without explicitly stating that it is a bird, and 3) they do not allow to represent complex preferences based on relations. For instance, a preference represented as a bag of terms “stable government African country” could be less likely to match interesting documents than an explicit list of countries that fulfill this restriction.

In this thesis, we address this limitation by elaborating on the semantic representation of both user interests and multimedia content. Our goal is to exploit these representations on a personalization approach for content access and retrieval of documents, in which documents are associated to a semantic index, where content is expressed by means of a set of knowledge concepts (Castells et al. 2007; Kiryakov et al. 2004). Among the possible semantic representation formalisms, ontologies bring a number of advantages (Staab and Studer 2004), as they provide a formal framework for supporting explicit, machine-processable semantics definitions, and support the inference and derivation of new knowledge based on existing one. Our approach adopts, but is not restricted to, an ontology based grounding for the representation of user profile and content descriptions. Our current personalization approach thus aims to draw advantage from the exploitation of concepts, and relations among them, for personalized and context-aware systems. The advantages that we draw from this representation can be summarized as:
O1) **Rich user profile representations:** Concept-based preferences are more precise and convey more semantics than simple keyword terms. Concepts are unambiguously identified to a piece of content or to a user profile. For instance, the concept “*WildAnimal:*Jaguar” is uniquely identified as “Jaguar, the animal species”. Furthermore, concepts can enrich their semantics by means of semantic properties. For instance the concept “*Woodpecker*” could be related to the “*Bird*” concept, through the relation “is a subspecies of”.

O2) **A formal ontological representation allows the expression of complex preferences:** The formal representation of ontologies allows the selection of a set of concepts by means of complex queries or relations. Previous mentioned complex preferences such as "I like traveling to Africa, but only to countries with stable governments” can be represented in an ontological, formal way.

### 1.2.2 User Context Modeling and Exploitation

Similarly to personalization, approaches aiming to achieve context-aware enhancements need to address issues of context representation, acquisition and exploitation.

As in user profile representation, context aware systems face difficulties regarding the representation of the user’s current contextual situation. This representation depends largely on the notion of context the system is considering. Context can be interpreted as the physical location of the user, the open applications in the user’s desktop, or the content the user has previously interacted with, to name a few. We will henceforth use the term “context” to denote a particular interpretation of context which is proposed and researched here, which is defined as the set of themes under which retrieval user activities occur within a retrieval session. Following our interpretation, context descriptions such as “I’m researching on tropical birds”, “tomorrow I’m travelling to Zurich” or “today I want to go to the cinema” are difficult to represent in a formal way. Similarly to user profiles, context has been commonly obtained and modeled using term related statistical approaches (Dumais et al. 2003; Rocchio and Salton 1971; Shen et al. 2005b). This has similar limitations as the ones pointed out for user preference representation. Thus, in our approach we explore a more expressive, concept-based semantic representation of the user context, in such a way that we have the same representation richness and enhanced semantics beyond (or complementarily to) pure statistical approaches.

Context acquisition is also tightly related to the particular interpretation of context, which makes it a difficult notion to capture and grasp in a software system. In general, sources of contextual information are implicit, i.e. they are not directly represented as a characterization of
the relevant aspects of the user and her situation. This implicit nature of context makes it difficult to acquire, as the same as for user profile learning approaches, the user can explicitly provide contextual information to the system, but it is useful to automate this input as far as possible, to relieve the user from an extra work. Context acquisition approaches based on a manual, explicit cooperation of the user are mostly based on Relevance Feedback techniques (RF), in which the user indicates which pieces of content are relevant in the current situation. However, even to a higher degree than manual user profiling, users are often reluctant to provide such information (Shen et al. 2005b). The main cause is that users have to provide this information in every interactive session, as the recorded short-term feedback is discarded once the session ends.

For this reason, implicit feedback has been widely researched as an alternative in context-aware retrieval systems (Kelly and Teevan 2003; White 2004b). Implicit feedback techniques often rely on monitoring the user interaction with the retrieval system, and extract the apparently most representative information related to what the user is aiming at. Again, typical implicit feedback approaches are based on statistical techniques, which, similarly to RF approaches, collect the most important documents that represent the user’s current context, from which a term-based representation is built (Leroy et al. 2003; Shen et al. 2005b; Sugiyama et al. 2004). An example of the implicit feedback model is the ostensive model (Campbell and van Rijsbergen 1996). This model copes with the drifting nature of context, using a time variable and giving more importance to recently occurring items than older ones. However, this model has only been applied to a term-based context representation (White et al. 2005b).

In this thesis we propose a notion of semantic runtime context, as standing for the set of concepts or themes involved in user actions during an ongoing retrieval session. We propose a method to build a dynamic representation of such semantic context model of retrieval user tasks in progress, by using implicit feedback techniques and adapting the ostensive model approach to our semantic representation of context. The goals for our research on context modeling can be summarized as:

O3) **Enhanced representation of the user context:** Similar to the semantic representation of the user profile, we aim to build a semantically rich representation of the user context in order to enable better, more meaningful and accurate representations of the user’s contextual situations.

O4) **Implicit feedback acquisition of live semantic context:** We do not want to burden users with explicitly having to provide their context. By adapting existing implicit
feedback approaches, our goal is to introduce a semantic acquisition approach of user context, taking also into consideration the drift nature of context.

The third issue in context-awareness, namely context exploitation, is also a complex research problem on its own. Once the system has a representation of the user context, how to best exploit it in benefit of the user is not a trivial question. A widely adopted approach is to take this context representation as a short-term interest profile, and exploit it similarly to long-term user profiles in a personalization approach. The main advantages of this approach are that the short-term user profile is usually narrower, more precise and focused on the task, as it has been acquired with the current session information, and wrong system guesses have a much lesser impact on performance, as the potentially incorrect predictions are discarded after the retrieval session. However, this approach does not make a clear, explicit difference between short-term and long-term interest. As a consequence, either the wider perspective of overall user trends, or the ability of the system to focus on temporary user priorities, is often lost. Room for improvement thus remains towards combining the advantages of personalization and context-aware approaches.

Our proposed approach is to use the user context in order to reduce potential inaccuracies of personalization systems, which typically apply their personalization algorithms out of context. In other words, although users may have stable and recurrent overall preferences, not all of their interests are relevant all the time. Instead, usually only a subset is active in the user’s mind during an outgoing task, and the rest can be considered as “noise” preferences. Our proposal is to provide a method for the combination of long-term (i.e. user profile) and short-term user interests (i.e. user context) that takes place in a personalized interaction, bringing to bear the differential aspects of individual users while avoiding distracting them away from their current specific goals. Many personalized systems do not distinguish the differences between long-term and short-term preferences, either applying the first or the latter, or treating both as the same. What we propose in this work is to have a clear distinction between these, and to model how both long-term interests (i.e. user preferences) and short-term interests (i.e. user context) can complement each other in order to maximize the performance of search results by the incorporation of context-awareness to personalization.

Our approach is based on the exploitation of the semantic representation of context in order to discard those preferences that are out of context in a current situation. This sort of contextual activation of preferences is based on the computation of the semantic distance between each user preference and the set of concepts in the current context. This distance is computed by
exploiting the semantic paths linking preferences to context, across the semantic network defined by a semantic Knowledge Base (KB). This approach aims to the following objective:

**O5) Complement personalization with context awareness:** We aim at a definition of user context and preferences which allows the combination of both techniques in a single retrieval system. To this end, the semantic representation of both user preferences and context enable finding non-explicit relations between context and user interests. For instance, if the context is related to Sports, preferences for Soccer or “Real Madrid” would be likely to be activated.

1.3 Contributions

The main original contributions of the research presented in this thesis, which will be described in the sections, include the following:

- **A semantic-based personalization framework for information retrieval.**
  
  A personalization model based on an enhanced semantic representation of user preferences and content is developed. Explicit domain concepts and relations are exploited to achieve performance improvements in personalized IR.

- **A semantic IR context modeling approach.**

  Context is a broad notion in many ways. One of the aims of the research undertaken in this thesis is to identify and synthesize a particular subset out of the full potential scope and variability of the term, concise enough to be approximated (represented, obtained, and applied), but powerful enough to enable specific improvements in IR performance. Similarly to the personalization framework, we propose a semantic-oriented model for context representation, based on explicit domain concepts defined upon an ontological grounding. On top of this, a context acquisition model is defined, based on implicit feedback techniques and ostensive models.

- **A user preference contextualization approach.**

  An approach to the contextualization of user preferences is proposed, based on a combination of long-term and short-term user interests. The proposed strategy consists of a semantic expansion technique, defined as a form of Constraint Spreading Activation (CSA), exploiting semantic relations in order to find the preferences that are (semantically) related to the live user context, and thus relevant for the retrieval task at hand.
• **Research of experimental evaluation methods for personalized and contextual IR.**

In order to evaluate the proposed contextual personalization approach, two complementary evaluation methodologies are followed. The aim of the proposed experimental methodology is to achieve a fair balance between a fine grained and reproducible scenario based evaluation, and an objective and more general user centered evaluation.

This thesis includes a strong evaluation component of the proposed approach. The evaluation of both personalized (Yang and Padmanabhan 2005) and interactive IR systems (Yang and Padmanabhan 2005) is known to be a difficult and expensive task. On top of that, a formal evaluation of a contextualization technique may require a significant amount of extra feedback from users in order to measure how much better a retrieval system can perform with the proposed techniques than without them. Our evaluation methodology tackles this evaluation complexity by combining real user and scenario-based methodologies.

### 1.4 Outline

This thesis is structured in five main Chapters.

In **Chapter 2** we overview the context of our work. We survey related work on the State of the Art of personalized and context-aware retrieval systems. This survey includes a comprehensive categorization of previous related work, in which we highlight the main characteristics on the conceptualization of user interests and/or context of the surveyed proposals.

In **Chapter 3** we describe our personalization framework, based on a conceptual representation of user interests. The main characteristic of this personalization framework is a concept-based representation of user interests, in which user profiles are represented as a set of weighted concept vectors. Adopting a probabilistic approach, the concept weights correspond to the intensity of user interest (or user dislike, in case of negative values) for each concept of the ontology. A Personal Relevance Measure (PRM) score computation technique for content items is introduced. This approach is based on the concept-vector similarity between the user profile and the concept vector representing the content item, obtained from the semantic index.

In **Chapter 4** we introduce the core part of this thesis: the application of context into our personalization framework. Firstly, the model for the semantic based representation of the user context is presented. This representation model, as well as the user preference model, is based on a weighted concept vector, where each weight value represents the probability that the concept in the ontology is related to the current context. Secondly, we introduce our approach for live semantic user context acquisition. This approach is based on an adaptation of the
ostensive model (Campbell and van Rijsbergen 1996) to a semantic index. The acquisition technique monitors user interactions with the retrieval system during the current session (e.g. user queries and opened content), extracting for each interaction step the concepts related to each action. Finally, our approach for the contextualization of preferences is described. This approach consists of a sort of fuzzy intersection between user preferences and context, by exploiting the semantic relations of the KB with a probabilistic model.

In Chapter 5 we evaluate the performance of our proposals. We survey the most important evaluation methodologies regarding adaptive and interactive retrieval systems in order to provide reasoning for our own evaluation methodology. Our evaluation methodology is based on the extension of simulated task situations (Borlund 2003), by including a set of user preferences and a hypothetical contextual simulation. We present an evaluation approach comprising two evaluation strategies. A first scenario-based methodology, in which user preferences and the interaction model are simulated, and a second user centered approach, in which user preferences are provided manually by users and users interact freely with our experimental retrieval system.

In Chapter 6 we provide conclusions of the thesis, together with further discussion and future potential lines of research as a continuation of the achievements presented herein.
Chapter 2

State of the Art

The aim of this section is to gather and evaluate existing techniques, approaches, ideas, and standards from the field of user modeling, personalization, and context aware systems. However, we will only focus on content-based systems, which exploit a user-content similarity computation, based on the similarity between the user interests and a document. We will thus exclude from this study systems that make use of other similarity functions, such as user-user similarity, exploited, for instance, in item based collaborative recommendation systems (Schafer et al. 2007). We have also added a selection of content-based recommendation systems, which although cannot be considered personalized retrieval approaches, follow the same model of computing a user-document similarity score.

2.1 Personalized Information Retrieval

Due to the massive amount of information that is nowadays available, the process of information retrieval tends to select numerous and heterogeneous documents as result of a single query; this is known as information overload. The reason is that the system cannot acquire adequate information concerning the user's wish. Traditionally, Information Retrieval Systems (IRSs) allow the users to provide a small set of keywords describing their wishes, and attempt to select the documents that best match these keywords. The majority of these queries are short (85% of users search with no more than 3 keywords (Jansen et al. 1998)) and ambiguous (Krovetz and Croft 1992), and often fail to represent the information need, nevertheless to say to represent also the implicit interests of the user. Although the information contained in these keywords rarely suffices for the exact determination of user wishes, this is a simple way of interfacing that users are accustomed to; therefore, there is a need to investigate ways to enhance information retrieval, without altering the way they specify their request. Consequently, information about the user wishes needs to be found in other sources.

The earliest work in the field of user modeling and adaptive systems can be traced back to the late 70’s (see e.g. (Perrault et al. 1978; Rich 1998)). Personalization technologies gained significance in the 90’s, with the boost of large-scale computing networks which enabled the deployment of services to massive, heterogeneous, and less predictable end-consumer audiences (Hirsh et al. 2000). One of the main boost on personalization approaches came in the mid-late
90’s with the appearance of personalized news access systems (Bharat et al. 1998; Lang 1995; Sakagami and Kamba 1997) and personalized information agents (Chen and Sycara 1998; Lieberman 1995; Widyantoro et al. 1997). Significant work has been produced since the early times in terms of both academic achievements and commercial products (see (Brusilovsky et al. 1998; Fink et al. 1997; Kobsa 2001; Montaner et al. 2003) for recent reviews).

Aspects of software that have been subject to personalization include, among others, content filtering (Micarelli and Sciarrone 2004), sequencing (Brusilovsky et al. 1998), content presentation (De Bra et al. 1998), content recommendation (Sheth and Maes 1993), content retrieval (Jeh and Widom 2003; Liu et al. 2004), user interfaces (Eisenstein et al. 2000; Hanumansetty 2004; Mitrovic and Mena 2002), task sequencing (Vassileva 1997), or online help (Encarnação 1997). Typical application domains for user modeling and adaptive systems include education (Brusilovsky et al. 1998; De Bra et al. 1998; Terveen and Hill 2001; Vassileva 1997), e-commerce (Ardissono and Goy 2000; Fink and Kobsa 2000), news (Bharat et al. 1998; Sheth and Maes 1993; Widyantoro et al. 1999), digital libraries (Callan et al. 2003; Smeaton and Callan 2001), cultural heritage (Ardissono et al. 2003), tourism (Fink and Kobsa 2002), etc. The field of user modeling and personalization is considerably broad.

The goal of these and other personalization systems is to gain the capability to change (adapt) any aspect of their functionality and/or appearance to the particularities of users, to better suit their needs. To do so, the system must have an internal representation (model) of the user. It is common in the user modeling discipline to distinguish between user model representation, user model learning/update, and adaptation effects or user model exploitation. Personalization of retrieval is the approach that exploits the user profiles, additionally to the query, in order to estimate the user’s wishes and select the set of relevant documents (Chen and Kuo 2000). In this process, the query describes the user’s current search, which is the local interest (Barry 1994), while the user profile describes the user’s preferences over a long period of time; we refer to the latter as global interest. The method for preference representation and extraction, as well as the estimation of the degree to which local or global interests should dominate in the selection of the set of relevant documents, are still open research issues (Wallace and Stamou 2002).

The aim of this section is not to provide a full overview of the field, but to report the state of the art on the area related to this work, i.e. personalized content-based retrieval, recommendation and filtering. Table 2.1 is a classification of the most important studied proposals. The representation column shows the representation approach of the user profile. The learning column classifies the adopted user profile learning approach. The last column, exploitation, shows which technique is used on the personalization phase. In the next sections we provide an
overview for each classification (i.e. user profile representation, learning and exploitation). Other classifications of personalization systems can be found at (Adomavicius and Tuzhilin 2005; Micarelli et al. 2007; Montaner et al. 2003).

<table>
<thead>
<tr>
<th>REFERENCE</th>
<th>REPRESENTATION</th>
<th>LEARNING</th>
<th>EXPLOITATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Ahn et al. 2007)</td>
<td>Terms</td>
<td>Hybrid</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Aroyo et al. 2007)</td>
<td>Concepts</td>
<td>None</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Asnicar and Tasso 1997)</td>
<td>Terms</td>
<td>Explicit</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Billsus and Pazzani 2000)</td>
<td>Terms</td>
<td>Hybrid</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Chakrabarti et al. 1999)</td>
<td>Concepts</td>
<td>Explicit</td>
<td>Link-based</td>
</tr>
<tr>
<td>(Chen et al. 2002)</td>
<td>Concepts</td>
<td>Implicit</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Chen and Kuo 2000)</td>
<td>Terms</td>
<td>Implicit</td>
<td>Query operations</td>
</tr>
<tr>
<td>(Chen and Sycara 1998)</td>
<td>Terms</td>
<td>Explicit</td>
<td>Query operations</td>
</tr>
<tr>
<td>(Chirita et al. 2006)</td>
<td>Terms</td>
<td>Implicit</td>
<td>Query expansion</td>
</tr>
<tr>
<td>(Dou et al. 2007)</td>
<td>Concepts</td>
<td>Implicit</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Gauch et al. 2003)</td>
<td>Concepts</td>
<td>Hybrid</td>
<td>Document weighing</td>
</tr>
<tr>
<td>(Haveliwala 2002)</td>
<td>Usage History</td>
<td>Implicit</td>
<td>Link-based</td>
</tr>
<tr>
<td>(Jeh and Widom 2003)</td>
<td>Documents</td>
<td>Implicit</td>
<td>Link-based</td>
</tr>
<tr>
<td>(Kerschberg et al. 2001)</td>
<td>Concepts</td>
<td>Explicit</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Koutrika and Ioannidis 2005)</td>
<td>Terms</td>
<td>Explicit</td>
<td>Query operations</td>
</tr>
<tr>
<td>(Krulwich and Burkey 1997)</td>
<td>Terms</td>
<td>Explicit</td>
<td>Query operations</td>
</tr>
<tr>
<td>(Lang 1995)</td>
<td>Terms</td>
<td>Explicit</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Lieberman 1995)</td>
<td>Documents</td>
<td>Implicit</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Liu et al. 2004)</td>
<td>Terms</td>
<td>Implicit</td>
<td>Query operations</td>
</tr>
<tr>
<td>(Ma et al. 2007)</td>
<td>Concepts</td>
<td>None</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Martin and Jose 2004)</td>
<td>Documents</td>
<td>Explicit</td>
<td>Query operations</td>
</tr>
<tr>
<td>(Micarelli and Sciarrone 2004)</td>
<td>Stereotypes</td>
<td>Explicit</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Middleton et al. 2003)</td>
<td>Concepts</td>
<td>Explicit</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Noll and Meinel 2007b)</td>
<td>Concepts</td>
<td>Implicit</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Pitkow et al. 2002)</td>
<td>Concepts</td>
<td>Implicit</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Sakagami and Kamba 1997)</td>
<td>Terms</td>
<td>Implicit</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Seo and Zhang 2001)</td>
<td>Terms</td>
<td>Hybrid</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Shen et al. 2005b)</td>
<td>Terms</td>
<td>Implicit</td>
<td>Query operations</td>
</tr>
</tbody>
</table>
Table 2.1. Overview of personalized information retrieval systems.

<table>
<thead>
<tr>
<th>Model</th>
<th>Feature</th>
<th>Type</th>
<th>Scoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Sieg et al. 2007b)</td>
<td>Concepts</td>
<td>Explicit</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Speretta and Gauch 2005)</td>
<td>Concepts</td>
<td>Implicit</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Sun et al. 2005)</td>
<td>Other</td>
<td>Implicit</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Sugiyama et al. 2004)</td>
<td>Terms</td>
<td>Implicit</td>
<td>Query operations</td>
</tr>
<tr>
<td>(Tan et al. 2006)</td>
<td>Usage History</td>
<td>Implicit</td>
<td>Query operations</td>
</tr>
<tr>
<td>(Tanudjaja and Mui 2002)</td>
<td>Concepts</td>
<td>Explicit</td>
<td>Link-based</td>
</tr>
<tr>
<td>(Teevan et al. 2005)</td>
<td>Terms</td>
<td>Implicit</td>
<td>Query operations</td>
</tr>
<tr>
<td>(Widyantoro et al. 1997)</td>
<td>Terms</td>
<td>Explicit</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Yuen et al. 2004)</td>
<td>Usage History</td>
<td>Implicit</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Zigoris and Zhang 2006)</td>
<td>Documents</td>
<td>None</td>
<td>Document scoring</td>
</tr>
</tbody>
</table>

2.1.1 Modeling Users

In the research area of user modeling, there are many different aspects of the user that have been taken into consideration. These aspects include user’s preferences, interests, goals/tasks, knowledge, background, experience with technology, demographic information (e.g. age, gender, language, residence, nationality, education, job, etc.), environment information (e.g. spatial location), personality traits, physical capabilities (e.g. typing skills, visual capacity, impairments, etc.), and affective state, etc. (Brusilovsky 2001).

Some model features are more useful in specific domains. For instance, subject-matter knowledge and learning style are of primary importance to personalise educational applications, but may be secondary or irrelevant for an on-line music store. Some systems create different subprofiles for a single user in different situations (e.g. work, leisure, learning) or behavioral states (e.g. looking for something, objective found, abandoned interest) (Hirsh et al. 2000).

These different modeling aspects can be classified into: domain-independent features, social profiling and semantic features. Example of domain-independent features are demographic data (De Bra et al. 1998) or physical abilities or disabilities (Fink et al. 1997). Social profiling follows the trend and the growing success of social networks, in which your connection’s attributes can be exploited in your own adaptation process (Agosto et al. 2005). Semantic features models create relations between the user and application objects such as products, contents, or processes. They refer to product attributes such as price, date, source, classification, i.e. to the meanings (semantics) held in the content.
In the IR context, systems normally focus on modeling a particular semantic feature: the user’s preferences. This is done in a way that the various forms of content access (search, browsing, navigation, etc.) can be adapted to the user’s wishes. User interests can be broadly classified into qualitative and quantitative (Chomicki 2003). Qualitative interests indicate a relation of preference, i.e. the user expresses a preference for a concept over a different one (e.g. “I prefer Autumn to Spring”). Quantitative interests express a degree of preference for a single concept (e.g. “I like Autumn”, “I don’t like Spring”). Qualitative interests make more sense on multi facet retrieval systems, such as database search (Chomicki 2003), in which several conditions can be expressed at once (e.g. “In the event of searching for a specific book, I prefer buying the cheapest one, regardless of the supplier”). Content based retrieval system, which are the focus of this background chapter, express, in general, user interests quantitatively, as normally this kind of conditions are difficult to exploit in unstructured content.

2.1.2 User Profile Representation

Any personalization system has some sort of internal representation of each user’s preferences. Broadly speaking, the user profile represents which general or global interests the user has that can be exploited by the system in order to adapt the information retrieval mechanism. The ways in which this profile is exploited are varied, e.g. the system can use the user preferences to refine the user’s query, to adapt the user’s navigation process or to adapt how the content is presented to the user. More details on the exploitation of the user profile will be introduced in a following section (section 2.1.4).

In the IR context, the most common approach of user profile representation is the bag of terms, where user interests are represented as a set of terms. Other systems add more semantics to this representation, by representing the user profile with a set of concepts. These concepts have some kind of background knowledge, which usually adds new relations between concepts. Other approaches are item-based, i.e. the user profile is represented as a set of documents that the user has interest in (e.g. a set of bookmarks or documents). The personalization system will exploit the document’s content or their intra-document relations in order to find other interesting documents (see section 2.1.3). Another important approach is collecting past interaction information of the user with the retrieval system, such as past queries and visited documents, which could be interpreted as interests of the user.

Terms

Bag of terms is the most common way of representing a user profile, probably because fits better the classic IR paradigm (Salton and McGill 1986), where both documents and users’ need
of information (i.e. queries) are expressed as a weighted vector of terms. The user profile in this case is thus represented in a similar way, by expressing user profiles as a set of weighted terms. Table 2.2 classifies these term-based approaches, which are a significant percentage over the studied systems, and in which systems based on weighted term vectors are majority.

Personalization systems that make use of simple, non-weighted terms for the user profile representation usually complement this approach by adding some semantic relations to the representation. This is the case of the ifWeb system (Asnicar and Tasso 1997), where the terms of the profile are linked by document correlation. Similarly to the ifWeb system, Liu et al (2004) link terms by correlation, but in this case the correlation is based on co-occurrence on a predefined set of categories, obtained from the Open Directory Project (ODP)6.

Systems that do use a weighted term representation of the user profile can also enrich the profile with further relations. For instance, Chirita et al. (2006) cluster the terms extracted from the documents of the user’s desktop environment. Terms are only weighted by term frequency; as the number of times that the term appears in all documents. The authors claim that in this case the “rareness” factor of a term, i.e. the idf (inverse document frequency, calculated as the number of documents that contain the term) should not be used, as a term can be very common on the user’s desktop, whereas being very rare in other corpora (e.g. the WWW). Chen and Sycara (1998) also cluster the term vectors, by having N different term vectors; each one intended for a different thematic-based profile. Their clustering technique is more basic: calculating the cosine similarity between each domain vector profile and the vector representation of the document to be added. Koutrika and Ioannidis (2005) link terms with logical operators, which indicate operations of negation, addition or substitution in relation with other terms. For instance, if the user is interested in technology, the user profile could have the term ‘Apple’ linked through the addition operator with the term ‘computer’, which will in some way “categorize” that concept. Somehow similar to this work, the InfoFinder agent (Krulwich and Burkey 1997) processes the interesting documents for the user with an ID3 learning algorithm and constructs a decision tree with the most important terms as nodes. The tree can be exploited to create on the fly Boolean personalized queries. The AIS system (Billsus and Pazzani 2000) also applies machine learning techniques, using the extracted terms from the visited documents, weighted by frequency of appearance, as a feature of a Bayesian network classifier.

---

6 Open Directory Project (ODP) is a public collaborative taxonomy of the WWW: http://dmoz.org/
In section 2.1.4 we present different ways of exploiting a profile based on term vectors, although, following the classic vector space model, it is very common to use a cosine similarity measure in order to compute the similarity of a document to the user profile, similarly to how the similarity is computed given a query and a document (Salton and McGill 1986).

<table>
<thead>
<tr>
<th>REFERENCE</th>
<th>REPRESENTATION</th>
<th>ADDED SEMANTIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Ahn et al. 2007)</td>
<td>Weighted terms</td>
<td>None</td>
</tr>
<tr>
<td>(Asnicar and Tasso 1997)</td>
<td>Terms</td>
<td>Term-Document correlation</td>
</tr>
<tr>
<td>(Billsus and Pazzani 2000)</td>
<td>Weighted terms</td>
<td>Bayesian network</td>
</tr>
<tr>
<td>(Chen and Kuo 2000)</td>
<td>Terms</td>
<td>None</td>
</tr>
<tr>
<td>(Chirita et al. 2006)</td>
<td>Weighted terms</td>
<td>Clusters</td>
</tr>
<tr>
<td>(Chen and Sycara 1998)</td>
<td>Weighted terms</td>
<td>Clusters</td>
</tr>
<tr>
<td>(Koutrika and Ioannidis 2005)</td>
<td>Terms</td>
<td>Logical operators</td>
</tr>
<tr>
<td>(Krulwich and Burkey 1997)</td>
<td>Terms</td>
<td>Decision Tree</td>
</tr>
<tr>
<td>(Lang 1995)</td>
<td>Weighted terms</td>
<td>None</td>
</tr>
<tr>
<td>(Liu et al. 2004)</td>
<td>Terms</td>
<td>Term-Category correlation</td>
</tr>
<tr>
<td>(Sakagami and Kamba 1997)</td>
<td>Weighted terms</td>
<td>None</td>
</tr>
<tr>
<td>(Seo and Zhang 2001)</td>
<td>Weighted terms</td>
<td>None</td>
</tr>
<tr>
<td>(Shen et al. 2005b)</td>
<td>Weighted terms</td>
<td>None</td>
</tr>
<tr>
<td>(Sugiyama et al. 2004)</td>
<td>Weighted terms</td>
<td>None</td>
</tr>
<tr>
<td>(Teevan et al. 2005)</td>
<td>Weighted terms</td>
<td>None</td>
</tr>
<tr>
<td>(Widyantoro et al. 1997)</td>
<td>Weighted terms</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 2.2. Overview of term-based user profile representation systems.

Concepts

A concept is “an abstract or generic idea generalized from particular instances”\(^7\). Concepts obtain their meaning (their semantic meaning) through relations to other concepts or other types of entities, such as documents or literals (e.g. terms). For instance, a category on Yahoo! Directory\(^8\) or in the ODP can be considered a concept that is defined by 1) the label of the concept, e.g. ‘Microsoft Windows’, 2) the Web documents related to the category, e.g. Microsoft’s homepage and 3) the parent concepts, e.g. ‘Operating Systems’, ‘Computers’, and

---

\(^7\) Second entry, Merriam Webster dictionary

\(^8\) http://dir.yahoo.com/
the children concepts, e.g. ‘Windows XP’, ‘Windows Vista’. In this work we want to stress the difference between taxonomies and ontologies. A taxonomy is a subset of an ontology, and represents a collection of concepts that are ordered in a hierarchical way, an example of taxonomy is ODP. Ontologies, among other formal specifications, and using a loosely interpretation, allow the definition of non-taxonomical relations, which are able to relate concepts by relations not limited to supertype-subtype semantics. For instance, in the previous example a not-taxonomical relation could be ‘Windows Vista’ \(\xrightarrow{competitor}\) ‘Mac OS X’.

Table 2.3 shows the classification of personalized retrieval systems with a concept-based user profile representation. Note that although some approaches claim to use ontological concepts, we have only classified as such those systems that make use of non-taxonomical relations. The definitions of these concepts, along with their semantic relations, are usually contained in a KB, which is usually static or semi-static. Example of KB are ODP, any ad-hoc created ontology or taxonomy and even folksonomies (Mathes 2004), created by the collaborative tagging process of users over a document corpus.

<table>
<thead>
<tr>
<th>REFERENCE</th>
<th>CONCEPTS</th>
<th>KNOWLEDGE BASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Aroyo et al. 2007)</td>
<td>Ontology</td>
<td>Ad-hoc</td>
</tr>
<tr>
<td>(Chakrabarti et al. 1999)</td>
<td>Taxonomy</td>
<td>Ad-hoc</td>
</tr>
<tr>
<td>(Chen et al. 2002)</td>
<td>Weighted Taxonomy</td>
<td>Ad-hoc</td>
</tr>
<tr>
<td>(Chirita et al. 2005)</td>
<td>Taxonomy</td>
<td>ODP, top 16</td>
</tr>
<tr>
<td>(Dou et al. 2007)</td>
<td>Weighted Taxonomy</td>
<td>Ad-hoc</td>
</tr>
<tr>
<td>(Gauch et al. 2003)</td>
<td>Weighted Taxonomy</td>
<td>Yahoo! directory, ODP</td>
</tr>
<tr>
<td>(Kerschberg et al. 2001)</td>
<td>Taxonomy</td>
<td>Ad-hoc</td>
</tr>
<tr>
<td>(Ma et al. 2007)</td>
<td>Taxonomy</td>
<td>ODP</td>
</tr>
<tr>
<td>(Middleton et al. 2003)</td>
<td>Taxonomy</td>
<td>Digital Library</td>
</tr>
<tr>
<td>(Noll and Meinel 2007a)</td>
<td>Folksonomy</td>
<td>Collaborative</td>
</tr>
<tr>
<td>(Pitkow et al. 2002)</td>
<td>Weighted Taxonomy</td>
<td>ODP, top 1K</td>
</tr>
<tr>
<td>(Sieg et al. 2007a)</td>
<td>Weighted Taxonomy</td>
<td>ODP</td>
</tr>
<tr>
<td>(Speretta and Gauch 2005)</td>
<td>Weighted Taxonomy</td>
<td>ODP</td>
</tr>
<tr>
<td>(Tanudjaja and Mui 2002)</td>
<td>Weighted Taxonomy</td>
<td>ODP</td>
</tr>
</tbody>
</table>

**Table 2.3.** Concept-based user profile representation in personalized retrieval systems.
One of the most exploited taxonomy KB in personalized systems is ODP, mainly because it is a widely adopted taxonomy with thousands of contributors, has a rich number of concepts and has a mass amount of Web documents related to concepts, which eases its understanding and exploitation by personalization systems. Gauch and Speretta represent the user profile by weighted topics belonging to the ODP and Yahoo! directory taxonomies in two related personalization approaches (Gauch et al. 2003; Speretta and Gauch 2005). Pitkow et al. (2002) and Sieg et al. (2007) also used weighted topics from ODP, but they limited the profile to the top 1K categories. Chirita et al. (2005) had to also limit the amount of top used topics to 16, as they computed a variation of the PageRank algorithm (Brin and Page 1998) for every topic, which was computationally expensive. Ma et al. (2002) represented the user profile as a set of topics, although this profile was obtained by mapping a term-based profile to these topics. They follow different heuristics for this mapping, checking if the term matches to any category name and, otherwise, checking the similarity of the term, or similar terms, to each category textual description. Tanujada and Mui (2002) did not weight the ODP topics in the profile, but they did allow indicating a negative or positive preference for each topic. A digital library retrieval system introduced by Middleton et al. (2003) made use of the existent digital library taxonomies to construct the user profile. Other systems build an ad-hoc taxonomy. This is the case of the Personal View Agent system (Chen et al. 2002) and the Focused Crawler system (Chakrabarti et al. 1999) which represent the user profile as a set of weighted topics from a predefined taxonomy. The system introduced by Dou et al. (2007) exploits its own classification scheme of 67 categories, which are the output of their automatic classification technique. In the WebSifter II system (Kerschberg et al. 2001) the users are able to create their own topics on their own personal taxonomy.

Going beyond taxonomies, Aroyo et al (2007) present a personalized system, in the TV personalized access domain, which exploits non-taxonomic relations over an ontological KB. In this case the user profile is constructed by concepts such as time of day, genre, or location. The genre concepts belong to a taxonomy, similarly to the taxonomy-based systems. However, the system also uses geographical and time ontologies in order to reason, for instance, which time of day is ‘Friday afternoon’ referred to or which documentaries are shot in a specific region of ‘England’. To the best of our knowledge, and at time of writing, there are no more examples of personalized system that truly exploits an ontology-based KB in a personalization system.

Folksonomies are a type of KB created collaboratively by the tagging actions of users, over a document corpus (Mathes 2004). Noll and Meinell (2007) demonstrate that user profiles based on these tag corpora can be useful for personalization systems. Their user profile is modeled as
a set of weighted tags from the folksonomy, where the weight is given by the frequency of the tag in the user’s tag set, following the hypothesis that the user’s frequent used tags are more representative of her interests.

Usage History

User profiles based on usage history represent previous interactions of the user with the system. The hypothesis of these approaches is that this interaction data can provide useful information in order to extract interests from the user. This hypothesis is shared among many systems, which exploit this kind of implicit information in order to construct the user profile (see section 2.1.3, implicit information). However, the systems here directly model the user profile as usage data, whereas other systems only consider it as one more step of the user profile learning process.

Usage history in retrieval systems is often limited to the clickthrough data. The clickthrough data is normally modeled as the query that the user executed into the system, the returned (multimedia) documents, and the subsequent documents that the user opened to view. In such a way that this interaction can lead to what documents could have been important to the user given the query. One simplification of the clickthrough data is taking only into consideration the user queries, with no interaction information whatsoever. Although the user profiles are represented as this usage history information, the personalization system usually applies some kind of preprocessing over this profiles, during the exploitation process. Table 2.4 shows the type of usage history and preprocessing procedure of these systems.

<table>
<thead>
<tr>
<th>REFERENCE</th>
<th>TYPE OF USAGE</th>
<th>PREPROCESSING</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Haveliwala 2002)</td>
<td>Queries</td>
<td>Classification</td>
</tr>
<tr>
<td>(Tan et al. 2006)</td>
<td>Clickthrough</td>
<td>Language Model</td>
</tr>
<tr>
<td>(Yuen et al. 2004)</td>
<td>Queries</td>
<td>Bayesian</td>
</tr>
</tbody>
</table>

Table 2.4. Overview of usage history based user profile representation in personalized retrieval systems.

Haveliwala (2002) suggests using past queries in order to bias the actual query of the user towards a given topic or other. Yuen et al. (2004) take a different approach by using the past queries as the test set of a Bayesian network that classifies terms into relevant or not to the user. Finally, Tan et al. (2006) compute a language model over the historical user profile in such a way that it can be combined with the language model of the actual query in the personalization step.
Documents

User profiles based on documents contain those items that are relevant to the user. The main application of this type of profiles is to find documents that are similar to those stored in the user profile. The Letizia system (Lieberman 1995) uses this approach by computing a term vector similarity between documents in the profile and those in the retrieval space. Jeh and Widom (2003) interpret the bookmarks of the user as a document-based user profile, exploiting the link structure of the Web to find similar documents. Zigoris and Zhang (2006) represent this profile as a graph, where users are nodes that link to the preferred documents. This graph is used as input of different machine learning approaches. Martin and Jose (2004) define a workspace where the user can store their interesting documents. In order to establish manual relations between documents, they define the concept of bundle, similar to the concept of folder in an operating system, where the user can store related documents.

Stereotypes

Stereotyping users is a well known approach for user modeling. Stereotypes are created either manually or by mining or clustering the information of a group of users. The goal of a stereotype is to represent the interests of a group of users, in such a way that personalizing the system with a given stereotype will maximize the quality of personalization for this group. Micarelli and Sciarrone (2004) follow this approach by having a predefined set of stereotypes (e.g. Java programmer, scientist, machine learning researcher). The system can be seen as a combination of stereotypes and term-based profiles, as each stereotype is further defined by a set of weighted terms.

Others

Sun et al. (2005) process the clickthrough information with a Singular Value Decomposition technique. They adapt a dimension reduction approach, known as Singular Value Decomposition (Furnas et al. 1988), in order to process the clickthrough information, which can be seen as triplets <user,query,document>, and calculate new weighted preference triplets for both new queries and documents. Therefore, the final user profile is a new set of weighted triplets <user,query,document> which will be used in the profile exploitation step.

2.1.3 User Profile Learning

User profile learning can be considered a single research area by itself, which studies how to acquire the user interests (Gauch et al. 2007). One of the main characteristic on user profile learning approaches is the amount of extra interaction needed from the user, i.e. learning
approaches can rely on an explicit user interaction (explicit approaches) or collect this information by monitoring the user’s interaction with the system (implicit approaches). The main advantage of explicit feedback systems is that there is a higher degree of confidence on the collected information, as the proper user is providing the preferences the system is adapting to. The problem with explicit techniques is that users are often reluctant to provide this information to the system as this requires an extra effort from the user (Shen et al. 2005b). On the other hand, implicit feedback techniques (Kelly and Teevan 2003) do not need to burden users with this extra information. Although some studies indicate that implicit indicators can be comparable to explicit (Claypool et al. 2001; Sakagami and Kamba 1997; Thomas and Hawking 2006), other study points that the quality of these depend on the searcher’s experience, the retrieval task at hand and the retrieval system (White et al. 2005a).

**Explicit Feedback User Profile Learning**

A basic mechanism for obtaining the user profile explicitly from the user is by directly letting the user manually edit or modify her profile. The personalized system can present a set of predefined terms/concepts of the user profile, and next allow the user assign interest weights for each or some concepts (Chirita et al. 2005; Micarelli and Sciarrone 2004), or even let the user construct the profile entirely (Kerschberg et al. 2001). However, a recent study concluded that these approaches, which demand an extra effort from the users, did not necessarily resulted on an increase of performance of the personalized system (Ahn et al. 2007).

Another type of explicit information that needs, in general, less interaction from the user, is requesting a set of documents which exemplify the user’s interests. This can be done either by asking for a set of example preferred content (Asnicar and Tasso 1997; Chakrabarti et al. 1999; Krulwich and Burkey 1997; Martin and Jose 2004; Sieg et al. 2007a), or by using relevance feedback techniques (Rocchio and Salton 1971), on which the user indicates a relevant set of documents encountered during a retrieval session, e.g. by indicating which documents returned by a query were subjectively relevant (Chen and Sycara 1998; Koutrika and Ioannidis 2005; Lang 1995; Middleton et al. 2003; Tanudjaja and Mui 2002; Widyantoro et al. 1997). Once the interesting documents are provided, and if the user profile is not based solely on documents, there has to be some type of preprocessing phase to construct the user profile. Typical approaches are extracting relevant terms from these documents and adding them to the user profile, normally using statistical techniques (Asnicar and Tasso 1997; Koutrika and Ioannidis 2005), being in fact able to extract negative interest from document marked explicitly as irrelevant (Widyantoro et al. 1997).
Based on the initial set of preferred documents, systems based on taxonomical profiles normally use classification techniques over the underlying KB, in order to update the user topic-based preferences (Chakrabarti et al. 1999; Middleton et al. 2003; Sieg et al. 2007a; Tanudjaja and Mui 2002). Other systems use these documents as input documents for their machine learning modules (Krulwich and Burkey 1997; Lang 1995). Table 2.5 presents a summary of these explicit techniques.

<table>
<thead>
<tr>
<th>REFERENCE</th>
<th>INTEREST</th>
<th>PREPROCESSING</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Asnicar and Tasso 1997)</td>
<td>Document/Example</td>
<td>Term extraction</td>
</tr>
<tr>
<td>(Chakrabarti et al. 1999)</td>
<td>Document/Example</td>
<td>Classification</td>
</tr>
<tr>
<td>(Chirita et al. 2005)</td>
<td>User Profile</td>
<td>None</td>
</tr>
<tr>
<td>(Chen and Sycara 1998)</td>
<td>Document/Feedback</td>
<td>Term extraction</td>
</tr>
<tr>
<td>(Kerschberg et al. 2001)</td>
<td>User Profile</td>
<td>None</td>
</tr>
<tr>
<td>(Koutrika and Ioannidis 2005)</td>
<td>Document/Feedback</td>
<td>Term extraction</td>
</tr>
<tr>
<td>(Krulwich and Burkey 1997)</td>
<td>Document/Example</td>
<td>Machine learning</td>
</tr>
<tr>
<td>(Martin and Jose 2004)</td>
<td>Document/Example</td>
<td>None</td>
</tr>
<tr>
<td>(Micarelli and Sciarrone 2004)</td>
<td>User Profile</td>
<td>None</td>
</tr>
<tr>
<td>(Middleton et al. 2003)</td>
<td>Document/Feedback</td>
<td>Classification</td>
</tr>
<tr>
<td>(Sieg et al. 2007a)</td>
<td>Document/Example</td>
<td>Classification</td>
</tr>
<tr>
<td>(Tanudjaja and Mui 2002)</td>
<td>Document/Feedback</td>
<td>Classification</td>
</tr>
<tr>
<td>(Widyantoro et al. 1997)</td>
<td>Document/Feedback</td>
<td>Term extraction</td>
</tr>
</tbody>
</table>

**Table 2.5.** Overview of explicit feedback learning in personalized systems.

**Implicit Feedback User Profile Learning**

Implicit feedback personalization systems have to monitor the user interaction with the system. The learning modules of these systems are usually based on clickthrough data: which queries the user has executed previously and which documents the user interacted with. Implicit feedback techniques are based on an implicit relevance indicator, which decides when a document was relevant to the user, without an explicit indication. Table 2.6 shows the classification of the different implicit feedback techniques used by these systems.

One approach is to monitor the browsing activity of the user (Chen et al. 2002; Sakagami and Kamba 1997; Sugiyama et al. 2004; Yuen et al. 2004). The early system Anatagonomy (Sakagami and Kamba 1997) applied a document classification technique over the user’s history
of opened documents, in order to update the taxonomy-based user profile. They used two implicit indicators of relevancy, the action of scrolling an article, and the action of ‘enlarging it’ (i.e. opening the document in a single window). Chen et al. (2002) also applied classification techniques, but used another type of relevancy indicator: a threshold of 2 minutes of viewing time. Sugiyama et al (2004) also used a viewing threshold indicator, although theirs was normalized by the number of terms on the document (they estimated 0.371 seconds of viewing time per term). They used term extraction techniques in order to add new weighted terms from the inferred interesting documents into the user profile. Yuen et al. (2004) solely use the opened document action, with no relevancy indicators, and they didn’t have to process this information, as they used the documents as input of their Bayesian network, used for the profile exploitation. The Letizia Web system, Lieberman (2005) had a slightly different implicit indicator: apart from considering an opened document (by following a link) as an indicator of interest, the fact that a user clicked a link is also considered an interest for the current document.

Clickthrough data complements browsing history with the information of the past queries that produced the result sets of documents. This is one of the most common sources of implicit feedback. However, Haveliwala (2002) suggested that past query themselves could be enough to learn the user profile model, in this case based on ODP categories and updated by means of classification techniques. It is important to differentiate those systems that exploit the whole interaction information (marked with an * on Table 2.6), which make use of the relations ‘launched query’ → ‘interacted document’, to those systems which treat queries and documents as different sources of implicit information. Chen and Kuo (2000) exploit the relation query-document by using a correlation matrix between the issued queries and the opened documents. Dou et al. (2007) directly use the obtained clickthrough data in order to search for similar data from previous users in a collaborative recommendation approach, combining the probability of previous user clicks with other ones of other users, and also adding a topic similarity between documents. Sun et al. (2005) adapted the Singular Value Decomposition (SVD) technique in order to process the clickthrough information, which can be seen as triplets <user,query,document>, used as tensors for the matrix dimension reduction technique. Tan et al. (2006) use queries and clicked documents in order to represent the usage history by means of a language model. Regarding systems that do not use the interaction information, Liu et al. (2004) and Speretta and Gauch (2005) use the past queries and opened documents as input of their classification approach for the profile construction, whilst Shen et al. (2005) and Teevan et al. (2005) extract the terms from these past queries and accessed documents. The latter, being a desktop search application, also considers actions such as creating a new document.
Another source of implicit information is given by Noll and Meinel (2007). Noll and Meinel used the tagging set of a user in order to create a final profile, by calculating tag frequencies as the importance of each tag to the user. However, the act of tagging a piece of content can be considered implicit or explicit depending on the final goal of the user. For instance, if the only intend of the tagging action is to facilitate the learning of the personalization system, it can be considered an explicit action. In the case of Noll and Meinel’s approach the tags are extracted from other tag corpora, such as bookmarking and content services, thus the users had already realized the tagging action with a different goal, and can be considered as an implicit action.

Chirita el al. (2006) make use of a corpus that exemplifies the interests of the user. They use the desktop documents as the source of the user profile, extracting terms related to this corpus and applying clustering techniques. They test several term extraction techniques such as document summarization, sentence selection, centroid calculation or NLP (Natural Language Processing).

<table>
<thead>
<tr>
<th>REFERENCE</th>
<th>IMPLICIT ITEM</th>
<th>PREPROCESSING</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Chirita et al. 2006)</td>
<td>Personal corpus</td>
<td>Data mining</td>
</tr>
<tr>
<td>(Chen et al. 2002)</td>
<td>Opened documents</td>
<td>Classification</td>
</tr>
<tr>
<td>(Chen and Kuo 2000)</td>
<td>Clickthrough*</td>
<td>Term correlation</td>
</tr>
<tr>
<td>(Dou et al. 2007)</td>
<td>Clickthrough*</td>
<td>Probability/Classification</td>
</tr>
<tr>
<td>(Haveliwala 2002)</td>
<td>Past queries</td>
<td>Classification</td>
</tr>
<tr>
<td>(Jeh and Widom 2003)</td>
<td>Bookmarks</td>
<td>none</td>
</tr>
<tr>
<td>(Liu et al. 2004)</td>
<td>Clickthrough</td>
<td>Classification</td>
</tr>
<tr>
<td>(Lieberman 1995)</td>
<td>Bookmarks</td>
<td>Term extraction</td>
</tr>
<tr>
<td>(Noll and Meinel 2007a)</td>
<td>Tagging</td>
<td>Tag frequency</td>
</tr>
<tr>
<td>(Pitkow et al. 2002)</td>
<td>Opened documents</td>
<td>Classification</td>
</tr>
<tr>
<td>(Sakagami and Kamba 1997)</td>
<td>Browsing history</td>
<td>Term extraction</td>
</tr>
<tr>
<td>(Shen et al. 2005b)</td>
<td>Clickthrough</td>
<td>Term extraction</td>
</tr>
<tr>
<td>(Speretta and Gauch 2005)</td>
<td>Clickthrough</td>
<td>Classification</td>
</tr>
<tr>
<td>(Sun et al. 2005)</td>
<td>Clickthrough*</td>
<td>SVD</td>
</tr>
<tr>
<td>(Sugiyama et al. 2004)</td>
<td>Browsing history</td>
<td>Term extraction</td>
</tr>
<tr>
<td>(Tan et al. 2006)</td>
<td>Clickthrough*</td>
<td>None</td>
</tr>
<tr>
<td>(Teevan et al. 2005)</td>
<td>Clickthrough</td>
<td>Term extraction</td>
</tr>
<tr>
<td>(Yuen et al. 2004)</td>
<td>Browsing history</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 2.6. Overview of implicit feedback learning in personalized systems.
Hybrid Approaches

Hybrid learning approaches seek to combine the advantages of explicit and implicit feedback (see Table 2.7). An example of hybrid approach is applying implicit techniques in order to mine the user interests, but also let the users view and edit their profiles. Gauch et al. (2003) and Ahn et al (2007) follow this approach, by monitoring the visited documents and using terms extraction and classification techniques respectively to build the profiles from this implicit information. Another approach is exemplified in the WAIR system (Seo and Zhang 2001), which takes advantage of explicit feedback techniques in order to solve the cold start problem, i.e. when the user is new to the system. The system will learn more rapidly by asking for explicit relevance feedback to the user and, when the user has a sufficiently rich user profile, the next updates can be done by monitoring the interactions of the user with the content. The WAIR system uses four different sources of implicit feedback: bookmarking, scrolling and opening actions, and reading time. The AIS system (Billsus and Pazzani 2000) uses an explicit relevance feedback approach, allowing the user to rate the accessed documents as “relevant” or “irrelevant”. The system’s implicit interest indicators include opened documents, taking into consideration the time of viewing and the user choosing to access more information about the opened document. They also include negative implicit feedback in the form of not followed results after a query.

<table>
<thead>
<tr>
<th>REFERENCE</th>
<th>IMPLICIT</th>
<th>EXPLICIT</th>
<th>PRE-PROCESSING</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Ahn et al. 2007)</td>
<td>Browsing history</td>
<td>User Profile</td>
<td>Term extraction</td>
</tr>
<tr>
<td>(Billsus and Pazzani 2000)</td>
<td>Clickthrough</td>
<td>Relevance Feedback</td>
<td>Machine learning</td>
</tr>
<tr>
<td>(Gauch et al. 2003)</td>
<td>Browsing history</td>
<td>User Profile</td>
<td>Classification</td>
</tr>
<tr>
<td>(Seo and Zhang 2001)</td>
<td>Browsing History</td>
<td>Relevance Feedback</td>
<td>Term extraction</td>
</tr>
</tbody>
</table>

Table 2.7. Overview of hybrid feedback learning in personalized systems.

2.1.4 User Profile Exploitation

Query Operations

Query operations techniques focus on refining the need of information expressed by the user’s input query. Query operations “modify” the actual information need, expressed by means of the user query, in such a way that the query considers also the global interests of the user.

Queries in information retrieval are often seen as vectors in a finite space model (see Figure 2.1), where the axis represent each term of the search space, and in which the documents, the user interests and the query are represented as weighted term vectors. These weighted vectors
assign a weight to each term, indicating how important is the term for the representation of the document, the user interest, or the query. The idea of personalization through query operations is adding information from the user profile to bring the query closer, viewed as a vector model, to the global preferences of the user. Figure 2.1 shows the graphical representation of this conception. When “personalizing” the query, the modified query “comes closer” geometrically to the user representation. In summary, we can say that the final query would be the combination of the local (or short-term) interest of the user (i.e. the query $q$) and the global interests of either the user $u_1$ or $u_2$.

$$\hat{q} = (0.7, 0.3)$$
$$\hat{u}_1 = (0.2, 0.8)$$
$$\hat{u}_2 = (0.9, 0.1)$$

$$\hat{q}_{m_1} = \frac{\hat{q}}{2} + \frac{\hat{u}_1}{2} = (0.45, 0.55)$$
$$\hat{q}_{m_2} = \frac{\hat{q}}{2} + \frac{\hat{u}_2}{2} = (0.80, 0.20)$$

**Figure 2.1.** Query operation example for a two dimension projection.

Depending on how the query is modified, the query operations are classified into *term reweighting* or *query expansion* operations. The example above shows a modification of the query term weights; this is what is called *term reweighting*. When new terms are added to the query, complementing the query representation with extra terms, the technique is labeled as *query expansion*. Take this example: a user searches for the term Jaguar, normally the system would not be able to disambiguate the term between “Jaguar the car brand” and “Jaguar the animal”. But if the system has some sort of query expansion technique, and the user profile contains the term “animal”, the system would be likely to return documents with the correct disambiguation by adding this term and finally using as input query “Jaguar animal” instead of
“Jaguar” alone. Finally, a system can have a combination of the two techniques, changing both the term importance weights and also adding new terms to the query.

Query expansion is often used in personalized meta-search engines (see section 2.1.5), these search systems redirect an input query to one or more external search engines, performing a merge or aggregation of each returned search result list. Terms from the user profile can be added to original queries and sent to several popular search engines. Applying term reweighting techniques to meta-searchers is more difficult, as normally these techniques need to access the internal functions of the search engine (although some, though not the most popular, allow term reweighting through optional parameters)

Relevance feedback (Rocchio and Salton 1971; Salton and Buckley 1990) is a particular case where query operations take place. This technique takes explicit relevance judgments from users, who decide which documents returned by a query are or not relevant. In personalization, rather than extracting the relevant information from this explicit user interaction, the search system uses the already learnt user profile as input of the relevance feedback’s query reformulation techniques. It is important that this query reformulation does not make the user profile predominant, which could induce to results that, although relevant to the user preferences, are not relevant to the original query. A generalization of both term-reweighting and query expansion techniques can be found in (Baeza-Yates and Ribeiro-Neto 1999).

An example of pure query expansion system is presented by Chen and Sycara (1998) and Chen and Kuo (2000), their approach exploits the term-correlation matrix that represents the user profile for the query expansion, by selecting the top most correlated terms to the query terms. Pitkow et al. (2002) and Shen et al. (2005) construct a user model based on a term weighted vector, expanding each user query with the top most important (i.e. those with higher weights) terms in the user model. In the desktop search engine presented by Teevan et al. (2005), the system builds implicitly a “personal index”, built up from implicit interactions of the user with the computer desktop and from interactions of the user with the search system. Terms and related weights are then extracted from this index and used for the query term reweighting and expansion at query time, enabling personalization for the desktop search. Chirita el al (2006) also personalize the desktop’s content retrieval. The user profile, learned from the documents in the user’s desktop, is composed of term clusters. Whenever the user issues a query, the top terms in the top clusters are added to the user query. Sugiyama et al. (2004) differentiate long-term preferences, as past queries or session browsing data and short-term preferences, as the current session’s history. Once the user profile is collected, they apply query expansion and term reweighting using the classic Rocchio query reformulation (Rocchio and Salton 1971).
Koutrika and Ioannidis (2005) apply query expansion but with a different query reformulation technique. Their user profiles are represented as a set of terms linked by expansion operators: AND, OR, NOT, and replacement. For instance, following the example profile in Figure 2.2, if the user issued the query “apple”, the final executed query, after the query expansion, would be “(Apple Inc. OR Apple Computer, Inc.) AND computers NOT fruit”. Krulwich and Burkey (1997) use a query reformulation technique based on decision trees. They extract terms from the currently open document, and apply this information to the decision tree, which represents the user profile. The output of this decision tree is a personalized query which results on documents both related to the document the user is viewing at the moment and to the user’s interests.

![Figure 2.2](image-url)  
*Figure 2.2. Example of user profile based on logic term operators*  

Martin and Jose (2004) create proactive queries, i.e., with no initial user query, analyzing the documents presented in the user profile, and using the explicit relations that the user indicated by grouping documents in different bundles. The query is then presented to the user who can choose to edit it and/or to launch it in a Web search engine. Chen and Sycara (1998) also create proactive queries, obtained from the top query terms of the user profile.

There are other approaches that, although share the idea of modifying the query with the user model, make use of different retrieval models. Tan et al. (2006) use a language model IR approach (Baeza-Yates and Ribeiro-Neto 1999). They make use of two different language models, one for the actual query of the user, and the second for the user profile representation. The final query is a combination of these two language models, in which the query is modified by the information given by the user profile. Although the system proposed by Liu et al. (2004) doesn’t alter the terms of the query, it does change the information in the sense that the query is biased towards one topic or the other, by selecting topic specific search engines. The user model is a term-topic correlation matrix that is used to relate the user’s query to a list of topics. For instance, a user with preferences for computers and electronics will be more likely to have a higher similarity between the query “*apple*” and the topic *Computer* than to the topic *Food*. The
query is then submitted several times. In a first mode, no category is indicated, and in subsequent modes top inferred categories are indicated. Finally, the results are merged using a voting algorithm and taking into consideration the ranking of the categories produced by the user model.

**Link-Based personalization**

Together with the previous section, these techniques are more often seen in commercial personalized search engines. Query operations are often applied because they fit well in personalized meta-search engines (see section 2.1.5). Link-based personalization is used because it follows the trend of link-based ranking algorithms. These have been a huge success in the past years, beginning with Google’s PageRank (Brin and Page 1998). Link-based personalization affects these document ranking techniques, which are based on the idea that “a page has a high rank if the sum of the ranks of its backlinks is high”. This query-independent score pushes the document up in the result set, so that pages that are considered “important” by the page rank algorithm are considered more relevant to any given user or query. One main advantage of these approaches is that the system does not have to take into consideration the content of the document, only the hyperlinks inherent in any Web page.

Page rank values are often computed by web crawlers that start from an initial page and do a random walk through the links of the page and the subsequent links of the pages pointed by the initial page. In general, link-based personalized algorithms are modifications of Google’s PageRank (Haveliwala 2002; Jeh and Widom 2003) or the HITS authority and hub algorithm (Chirita et al. 2003; Tanudjaja and Mui 2002). However, there are different ways to introduce personalized search in page rank algorithms:

- **Topic sensitive page rank.** A different page rank value is computed for every topic, in order to capture more accurately the notion of importance within each category. Thus, the system is able to personalize the final results with the user’s desired topics by combining the topic-biased page ranks. The topic information can be extracted from a category hierarchy (Haveliwala 2002; Tanudjaja and Mui 2002), using hard relations with already existent Web categories like ODP, or starting from a set of documents considered representative of the user interests (Chakrabarti et al. 1999), using a classifier to relate representative documents, the crawled documents and the query to the set of predefined topics.

- **Relevant Documents.** A set of relevant documents is used to alter the normal page rank algorithm and give a higher rank value to documents related (through links) to this
initial set (Chirita et al. 2003; Jeh and Widom 2003). This set of relevant documents can be extracted from the bookmarks of the user, which are considered a good source of interests.

Personalized alterations of the page rank algorithms have a tradeoff of scalability, as computing these values requires high computational resources, and it is impossible nowadays to compute a full personal page rank value for every user (this would be with no doubt, the ideal use case), which was the original Brin and Page suggestion. Some solutions to this have been the calculation of only a small set of values for small set of topics (Chakrabarti et al. 1999; Haveliwala 2002), or more efficient algorithms where partial page rank vectors are computed, allowing the combination at runtime of these into a final personalized vector (Jeh and Widom 2003).

**Document scoring**

Document scoring algorithms modify the final ranking of the result set of documents. Most search engines compute a ranking value of relevance between the document and the need of information (e.g. the user’s query). Note that this ranking can be the combination of several scores, but all these are user independent. A personalized search engine can then compute a personalized ranking value for every document in the result set. The benefits of this approach is that this value has only to be computed for the returned top result set of documents. The main drawback is that this value has to be computed at query time. This algorithm is also suitable for meta search engines (Kerschberg et al. 2001), as the user-dependent algorithm can focus on a small quantity of the top returned documents, being able to compute a personalized score on the fly, by accessing the document’s content or even just using the provided snippet summaries.

The user-dependent score usually comes from a document-user similarity function, based on term-frequency similarity (Micarelli and Sciarrone 2004), classification and clustering (Middleton et al. 2003), Bayesian approaches (Zigoris and Zhang 2006), etc. Figure 2.3 shows the typical flow chart of this type of personalized IRSs.
The most common application of this user-document similarity score is its combination with the search user-independent score, resulting on a personalized result reorder. Systems that use classification techniques can also cluster the results and present first those clusters that have a higher similarity to the profile. Finally, the user-document similarity score can also be used to aid the navigation and browsing actions of the user. In this IR paradigm, the user can navigate the system’s corpus, while the system can suggest interesting links or adapt the browsing options in a personalized way. Table 2.8 classifies the document scoring techniques by which is the final use of the score value and how is the user-document similarity score obtained.

**Figure 2.3.** Typical schema of document scoring on personalized retrieval systems.
<table>
<thead>
<tr>
<th>REFERENCE</th>
<th>SCORE USE</th>
<th>SIMILARITY MEASURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Ahn et al. 2007)</td>
<td>Result reorder</td>
<td>Term-vector</td>
</tr>
<tr>
<td>(Aroyo et al. 2007)</td>
<td>Result Filtering</td>
<td>Property</td>
</tr>
<tr>
<td>(Billsus and Pazzani 2000)</td>
<td>Result reorder</td>
<td>Machine learning</td>
</tr>
<tr>
<td>(Chen et al. 2002)</td>
<td>Result reorder</td>
<td>Topic-vector similarity</td>
</tr>
<tr>
<td>(Chirita et al. 2005)</td>
<td>Result reorder</td>
<td>Topic similarity</td>
</tr>
<tr>
<td>(Chen and Sycara 1998)</td>
<td>Result reorder</td>
<td>Term-vector similarity</td>
</tr>
<tr>
<td>(Dou et al. 2007)</td>
<td>Result reorder</td>
<td>User/Topic similarity</td>
</tr>
<tr>
<td>(Gauch et al. 2003)</td>
<td>Result reorder</td>
<td>Term-vector similarity</td>
</tr>
<tr>
<td>(Kerschberg et al. 2001)</td>
<td>Result reorder</td>
<td>Search engine/popularity/topics</td>
</tr>
<tr>
<td>(Lang 1995)</td>
<td>Clustering</td>
<td>Classification</td>
</tr>
<tr>
<td>(Lieberman 1995)</td>
<td>Navigation</td>
<td>Term-vector similarity</td>
</tr>
<tr>
<td>(Ma et al. 2007)</td>
<td>Result reorder</td>
<td>Topic similarity</td>
</tr>
<tr>
<td>(Micarelli and Sciarrone 2004)</td>
<td>Result reorder</td>
<td>Term frequency</td>
</tr>
<tr>
<td>(Middleton et al. 2003)</td>
<td>Clustering</td>
<td>Classification</td>
</tr>
<tr>
<td>(Noll and Meintel 2007a)</td>
<td>Result reorder</td>
<td>Term-vector similarity</td>
</tr>
<tr>
<td>(Pitkow et al. 2002)</td>
<td>Result reorder</td>
<td>Term-vector similarity</td>
</tr>
<tr>
<td>(Sakagami and Kamba 1997)</td>
<td>Navigation</td>
<td>Term-vector similarity</td>
</tr>
<tr>
<td>(Seo and Zhang 2001)</td>
<td>Result reorder</td>
<td>Term-vector</td>
</tr>
<tr>
<td>(Sieg et al. 2007a)</td>
<td>Result reorder</td>
<td>Topic</td>
</tr>
<tr>
<td>(Speretta and Gauch 2005)</td>
<td>Result reorder</td>
<td>Topic</td>
</tr>
<tr>
<td>(Sun et al. 2005)</td>
<td>Result reorder</td>
<td>User-query-document</td>
</tr>
<tr>
<td>(Widyantoro et al. 1997)</td>
<td>Result reorder</td>
<td>Term-vector</td>
</tr>
<tr>
<td>(Yuen et al. 2004)</td>
<td>Clustering</td>
<td>Classification</td>
</tr>
<tr>
<td>(Zigoris and Zhang 2006)</td>
<td>Result reorder</td>
<td>Machine learning</td>
</tr>
</tbody>
</table>

Table 2.8. Classification of document scoring exploitation in personalized systems.

- **Result Reorder**

In result reorder techniques, the top n returned documents by the query are reordered according to the relevance of these documents to the user profile. The underlying idea is improving the ranking of documents that are not only relevant to the query, but also relevant to the user’s
wishes. Unlike query operations (see above 2.1.1), results reorder does not change the query information, thus guaranteeing the query relevance.

An example of result reordering approach is the HUMOS system (Micarelli and Sciarrone 2004), which modifies the results of the query returned by a popular search engine. Each user profile contains a set of weighted stereotypes, which represent an interest for a specific domain. Each stereotype has associated a topic and a set of terms related to the domain. A document is finally ranked by using a term frequency similarity, calculated by a scalar product between the occurrences of a term on the user profile and on the document, using the weight of the stereotype the term belongs to. They also introduce the concept of the Term Data Base (TDB), which is a set of terms related to the domains of interest of the user, that, in a lower degree, are also taken into consideration.

Zigoris and Zhang (2006) use a hierarchical Bayesian network representation of the user profile in order to reorder the search results. The main advantage is that the system can exploit profiles from other users when the system does not have any information about a new user to the system, palliating the cold start problem. The AIS system (Billsus and Pazzani 2000) uses a naïve Bayesian classifier over the user profile. Using as features the terms in a document, the document’s personalized score is computed by the predictor value given by the classifier.

When the user profiles are represented as a set of taxonomic concepts, it is common to use a topic-document similarity to compute the personalization score (Chirita et al. 2005; Sieg et al. 2007a). The similarity score is calculated by means of a distance measure (e.g. a taxonomic distance) between the topics associated to the documents and the topics in the user profile. Vector similarity between the user representation and the document representation is one of the most common algorithms for computing the personalization score, this vector similarity is often calculated by the cosine value of the two vector representations. In the case of taxonomy-based systems (Chen et al. 2002; Gauch et al. 2003; Ma et al. 2007; Speretta and Gauch 2005), the similarity value is computed between the weighted topics representing the interests of the user and the topics associated to each search result. Pitkow et al. (2002) compute the vector similarity between the terms associated to the topics of the user profile and the title and metadata of the returned documents. Term-based recommender systems (Ahn et al. 2007; Chen and Sycara 1998; Seo and Zhang 2001; Widyantoro et al. 1997) compute the same vector similarity value, but using the term vector representation of the document content.

Collaborative filtering methods commonly perform a result reorder, combining the user profile with other user profiles (usually with a user-user similarity measure). Sun et al. (2005) and Dou et al. (2007) mine the query log clickthrough information in order to perform a collaborative
personalization of the result set; ranking higher those documents that similar users had clicked previously in similar queries. Sun et al (2005) applies a dimensional reduction preprocessing to the clickthrough data in order to find latent semantic links between users, queries and documents. This links enable ranking documents as interesting for the user. Dou et al. (2007) complement their collaborative similarity measure with a user-topic document-topic similarity value.

The Sensee TV framework by Aroyo et al. (2007) uses ontological properties to boost results that fulfill specific properties defined by the user. For instance, let us suppose that a user has a preference for historic documentaries centered on the regions of England. If the user issues a query “Friday” to search for programs that will be aired the next Friday, programs of this genre and related to England locations would be shown first to the user.

Meta search engines combination methods can be personalized by different criterions. In (Kerschberg et al. 2001) the users can express their preference for a given search engine, for a set of topics or for the desired popularity of the search results. The final relevance measure would be the combination of this personal ratings applied to each of the listings of the search engines.

- **Result Clustering**

Query results are clustered in a set of categories, presenting first the categories more relevant to the user (Lang 1995; Middleton et al. 2003; Yuen et al. 2004). The algorithm 1) takes the result set of a query, 2) obtains the set of categories related to the documents in the result set, 3) reorders the set of categories according to the user profile and 4) presents the top n documents for each category. In general, presenting the top three categories in each page with four-five documents for each category gives a good performance. The system has to allow the user to select a concrete category to see all the documents of the result set related to this category.

- **Navigation Support**

Navigation support affects how the user browses or navigates through the system’s content. This can be done by either suggesting links to follow next (Asnicar and Tasso 1997; Lieberman 1995) or by adapting the layout of information presented to the user (Sakagami and Kamba 1997). Lieberman (1995) assists the user’s Web browsing session by calculating the personalization score on the links of the current opened document. Those links with higher scores are suggested to the user. Asnicar and Tasso (1997) classify each link in the document as interesting or not to the user, creating a final reordered list of links by relevance. The links of the linked documents are also taken into consideration, having an iterative algorithm resembling
to a local personalized web crawler. The Anatagonomy system (Sakagami and Kamba 1997) introduces a way of personalizing a news portal. The personalization score is computed for recent news and, depending on this score, a personalize layout of a first page of news is presented to the user.

### 2.1.5 Personalization in Working Applications

The number of search engines with personalization capabilities has grown enormously in the past years, from social search engines, where users can suggest collaboratively which are the best results for a given query, to vertical search engines, where users can customize a domain specific search engine. There is an incoming interest by commercial search engine companies such as Yahoo, Microsoft or Google, but the latter has been the first to show truly personalization capabilities. The following is a list of those that have more properties in common with our proposed approach.

- **Google Personal**

  Google’s personalized search (currently discontinued) was based on a topic representation of the user preferences, manually selected by the user from ODP. The personalization only affected the search results related to a category selected by the user. The user could change the degree of personalization by interacting with a slider, which dynamically reorder the first ten results.

- **Google Co-op**

  Google Co-op allows the creation of shared and personalized search engines in the sense that users are able to tag web pages and filter results with this new metadata. Tags are not meant to be a full description of the content of the annotated Web pages. It is more oriented to what could be called “functionality tags” (e.g. tagging a page as a review report for the custom search engine for digital cameras). Domains and keywords can also be added to modify search ranking and expand the user’s query.

- **iGoogle**

  Recently, Google change the name of the personalized homepage to iGoogle⁹, stressing the personalization capabilities. Although we cannot be really sure what are the concrete applied techniques specifically on Google’s search engine, and these technologies are still incipient, two

---

⁹ [http://www.igoogle.com](http://www.igoogle.com)
US patents on personalized search have been filed by Google in recent years (Badros and Lawrence 2005; Zamir et al. 2005). These patents describe techniques for personalized search results and rankings, using search history, bookmarks, ratings, annotations, and interactions with returned documents as a source of evidence of user interests. The most recent patent specifically mentions "user search query history, documents returned in the search results, documents visited in the search results, anchor text of the documents, topics of the documents, outbound links of the documents, click through rate, format of documents, time spent looking at document, time spent scrolling a document, whether a document is printed/bookmarked/saved, repeated visits, browsing patterns, groups of individuals with similar profile, and user submitted information". Google patents consider explicit user profiles, including a list of weighted terms, a list of weighted categories, and a list of weighted URLs, obtained through the analysis of the aforementioned information. Techniques for sharing interests among users, and community building based on common interests, are also described. As an optional part of the user profiles, the patent mentions "demographic and geographic information associated with the user, such as the user's age or age range, educational level or range, income level or range, language preferences, marital status, geographic location (e.g., the city, state and country in which the user resides, and possibly also including additional information such as street address, zip code, and telephone area code), cultural background or preferences, or any subset of these".

- **Eurekster**

Although is mostly oriented to “search groups”. This search engine\(^\text{10}\) includes the ability to build explicitly a user profile by means of terms, documents and domains. It is a meta search engine based on Yahoo! search engine, so only query expansion and domain focused searches can be performed. Users can also mark which search results they think are the most relevant for a given query, so that similar queries can make use of this information.

- **Entopia Knowledge Bus**

Entopia is a Knowledge Management company which sold a search engine named k-bus, receiving many awards and being selected as the best search engine technology in 2003 by the Software & Information Industry Association. This search engine is promoted to provide highly personalized information retrieval. In order to rank the answers to a query, the engine takes into account the expertise level of the authors of the contents returned by the search, and the expertise level of the users who sent the query. Those expertise levels are computed by taking

\(^{10}\) [http://www.eurekster.com](http://www.eurekster.com)
into consideration previous interactions of different kinds between the author and the user on some contents.

- **Verity K2**
  
The latest version of the K2 Enterprise Solution of Verity, one of the leading companies in the search engine markets for businesses, includes many personalization features to sort and rank answers to a query. To build users profiles, K2 tracks all the viewing, searching, and browsing activities of users with the system. Profiles can be bootstrapped from different sources of information including authored documents, public e-mail forums in the organization, CRM systems, and Web server logs. A user can provide feedback not only to documents but also to a recommendation coming from a specific user, thus reinforcing the value of a document and also the relationship between both users.

- **MyYahoo**
  
The personalization features of yahoo personal search engine are still rather simple\(^1\). Users are able to “ban” a URL from the search results, or to save pages to a “personal Web” that will give a higher priority on these pages once they appear in a search result set.

### 2.2 Context Modeling for Information retrieval

One of the key drivers and developments towards creating personalized solutions that support proactive and context-sensitive systems has been the results from research work in personalization systems. The main indication derived from these results showed that it was very difficult to create generic personalization solutions, without in general having a large knowledge about the particular problem being solved. In order to address some of the limitations of classic personalization systems, researchers have looked to the new emerging area defined by the so-called context-aware applications and systems (Abowd et al. 1997; Brown et al. 1997).

The notion of context-awareness has been long acknowledged as being of key importance in a wide variety of fields, such as mobile and pervasive computing (Heer et al. 2003), computational linguistics (Finkelstein et al. 2002), automatic image analysis (Thanos et al. 2007), or information retrieval (Bharat 2000; Haveliwala 2002; Kim and Chan 2003), to name a few. The definitions of context are varied, from the surrounding objects within an image, or nearby shots of a video scene, to the physical location of the user. The definition and treatment of context varies significantly depending on the application of study (Edmonds 1999).

\(^1\) [http://my.yahoo.com](http://my.yahoo.com)
Context in information retrieval has also a wide meaning, going from surrounding elements in an XML retrieval application (Arvola et al. 2005), recent selected items or purchases on proactive information systems (Billsus et al. 2005), broadcast news text for query-less systems (Henzinger et al. 2003), recently accessed documents (Bauer and Leake 2001), visited Web pages (Sugiyama et al. 2004), past clickthrough data (Bharat 2000; Dou et al. 2007; Shen et al. 2005b), text surrounding a query (Finkelstein et al. 2002; Kraft et al. 2006), text highlighted by a user (Finkelstein et al. 2002), etc.

One of the most important parts of any context-aware system is the context acquisition. Note that this is conceptually different to profile learning techniques. On the one hand, context acquisition aims to discover the short-term interests (or local interests) of the user (Dou et al. 2007; Shen et al. 2005b; Sugiyama et al. 2004), where the short-term profile information is usually disposed once the user's session is ended. On the other hand, user profile learning techniques mine the long-term interests of the user, and such preferences are intended to be part of the user profile during multiple sessions.

One simple solution for context acquisition is the application of explicit feedback techniques, like relevance feedback (Rocchio and Salton 1971; Salton and Buckley 1990). Relevance feedback builds up a context representation through an explicit interaction with the user. In a relevance feedback session:

1) The user makes a query.
2) The IR system launches the query and shows the result set of documents.
3) The user selects the results that she considers relevant (or irrelevant) from the top n documents of the result set.
4) The IR system obtains information from the relevant (and irrelevant) documents, operates with the query and returns to 2).

Relevance feedback has been proven to improve the retrieval performance. However, the effectiveness of relevance feedback is considered to be limited in real systems, basically because users are often reluctant to provide such information (Shen et al. 2005b). This information is needed by the system in every search session, asking for a greater effort from the user than explicit feedback techniques in personalization. For this reason, implicit feedback is widely chosen among context-aware retrieval systems (Campbell and van Rijsbergen 1996; Kelly and Teevan 2003; Sugiyama et al. 2004; Vallet et al. 2008; White et al. 2006; White and Kelly 2006).

A complete classification of contextual approaches related to IR systems can be found in Table 2.7. Context-aware systems can be classified by 1) the concept the system has for context, 2)
how the context is acquired, 3) how the context information is represented and 4) how the context representation is exploited.

<table>
<thead>
<tr>
<th>REFERENCE</th>
<th>CONCEPT</th>
<th>ACQUISITION</th>
<th>REPRESENTATION</th>
<th>EXPLOITATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Akrivas et al. 2002)</td>
<td>Surrounding context</td>
<td>Term thesaurus</td>
<td>Concepts</td>
<td>Query operations</td>
</tr>
<tr>
<td>(Bauer and Leake 2001)</td>
<td>Desktop</td>
<td>Term extraction</td>
<td>Term Vector</td>
<td>Context revisit</td>
</tr>
<tr>
<td>(Bharat 2000)</td>
<td>Clickthrough</td>
<td>Clickthrough</td>
<td>Usage History</td>
<td>Context revisit</td>
</tr>
<tr>
<td>(Billsus and Pazzani 2000)</td>
<td>Clickthrough</td>
<td>Term extraction</td>
<td>Term Vector</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Budzik and Hammond 1999)</td>
<td>Desktop</td>
<td>Term extraction</td>
<td>Term Vector</td>
<td>Proactive queries</td>
</tr>
<tr>
<td>(Budzik and Hammond 2000)</td>
<td>Desktop</td>
<td>Text mining</td>
<td>Term Vector</td>
<td>Query operation</td>
</tr>
<tr>
<td>(Dou et al. 2007)</td>
<td>Clickthrough</td>
<td>Classification</td>
<td>Topic vector</td>
<td>Query operation</td>
</tr>
<tr>
<td>(Dumais et al. 2003)</td>
<td>Desktop</td>
<td>Term frequency</td>
<td>Inverted index</td>
<td>Context revisit</td>
</tr>
<tr>
<td>(Finkelstein et al. 2002)</td>
<td>Surrounding context</td>
<td>Term extraction</td>
<td>Term vector</td>
<td>Query operation</td>
</tr>
<tr>
<td>(Haveliwala 2002)</td>
<td>Surrounding context</td>
<td>Classification</td>
<td>Topic Vector</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Kraft et al. 2006)</td>
<td>Surrounding context</td>
<td>Entity extraction</td>
<td>Term Vector</td>
<td>Query operation</td>
</tr>
<tr>
<td>(Leroy et al. 2003)</td>
<td>Clickthrough</td>
<td>Term extraction</td>
<td>Terms Vector</td>
<td>Query operation</td>
</tr>
<tr>
<td>(Melucci 2005)</td>
<td>Location</td>
<td>None</td>
<td>Vector base</td>
<td>Query operation</td>
</tr>
<tr>
<td>(Rhodes and Maes 2000)</td>
<td>Desktop</td>
<td>Term extraction</td>
<td>Terms</td>
<td>Document scoring</td>
</tr>
<tr>
<td>(Shen et al. 2005a)</td>
<td>Clickthrough</td>
<td>Language Model</td>
<td>Terms, probability</td>
<td>Query operation</td>
</tr>
</tbody>
</table>
2.2.1 Concept of Context

Even if we narrow our study to information retrieval agents, there’s still a variety of interpretations of the concept of context. These interpretations include which are the sources of context, how can we extract this contextual information and, more importantly, how can we use this information to benefit the user’s search process. Context’s interpretation can go to something as simple as the query that the user just inputted into the retrieval system (Akrivas et al. 2002) to all the current opened desktop applications and current desktop interactions (Dumais et al. 2003), including ubiquitous properties such as location or time (Melucci 2005).

Clickthrough information

Clickthrough data is one of the most used sources for context acquisition. Differently from personalized systems, where historical clickthrough data is mined in order to construct a long-term user profile, in the case of context-aware retrieval systems the clickthrough data is normally bound to a single session of interaction with the system. One exception is the AIS system (Billsus and Pazzani 2000), which defines the contextual information as the last 100 interactions with the system, regardless of the session. Clickthrough data, in the case of contextual retrieval systems, is mined as a way of obtaining the short-term interests of the user, i.e. what is the task that the user is trying to achieve at the moment. Knowing this information can enable the system to adapt the retrieval mechanism in such a way that the current interests of the user are satisfied to a greater extent.

Similarly to personalization systems (see implicit feedback on section 2.1.3), when interpreting clickthrough data the system can choose to take into consideration both the query and the posterior interactions with the documents (Bharat 2000; Dou et al. 2007; Shen et al. 2005a;
Shen et al. 2005b), or limit the source of context to just the opened documents (Bauer and Leake 2001; Billsus and Pazzani 2000; Sugiyama et al. 2004; White and Kelly 2006).

**Desktop information**

Another concept of context is the information from the user’s desktop. For instance, opened windows and documents, sent emails, etc. One of the main restrictions on desktop based contexts is that, differently to clickthrough data, which can be harvested from any Web search application, desktop actions or desktop information have to be obtained from a local application. One advantage is that these retrieval systems can gather interaction information before the initial user interaction with the system, which can be a solution for the “cold start” problem. In clickthrough based systems, when starting a session the system has not yet obtained any contextual clue of the user intentions, whereas in desktop based systems, the user could have previously interacted with other applications.

Budzik and Hammond (1999, 2000) extract the contextual information from open applications, such as document writing or web browsers. Vishnu (2005) extends this context to messaging applications. The Stuff I’ve Seen system (Dumais et al. 2003) does not only take into consideration opened Web documents and word applications, but also uses information on emails and previously created documents. The Just-in-time system (Rhodes and Maes 2000) also considers Word and email applications, but they only consider the last 500 edited words by the user. The WordSieve system (Bauer and Leake 2001) monitors the short-term sequence of accesses to documents, building up a task representation of the user’s context.

**Surrounding query context**

Other interpretation of the user’s context is the information that could have surrounded the input of the query. This information can give clues on what produced the need of information of the user, interpreting what the user was viewing at the moment of generating the query. This could be similar to desktop-based systems, which seek to gather what information “surrounded” the user at the moment of executing the query. However, in this case the information is finer grained, as it is normally the proper user who indicates which is the surrounding information that motivated the query. For instance, the IntelliZap system (Finkelstein et al. 2002) allows the user to send a query over a highlighted term, taking the surrounding text as the context of this query. This of course compels the user to do an extra effort in order to provide this information, given that this contextual information can be provided by the user, and that it is available on the applications the user is interacting with. Kraft et al. (2006) combine the benefits of both desktop based and surrounding query approaches, by also taking into consideration the information of
open documents. Haveliwala (Haveliwala 2002) also gives importance to the surrounding text of the query, along with possible past queries. Akrivas et al. (2002) claim that the context of a query can be represented with the very same query terms. The authors’ hypothesis is that each term of a query can be disambiguated by analyzing the rest of the query terms. For instance, disambiguating the term 'element' with the chemistry meaning in the case it appears among other chemistry related terms or assigning an XML meaning in the case it appears among terms related to the XML field. It is unclear that this technique will be always successful, as a big percentage of user's queries contain no more than one or two terms (Jansen et al. 1998).

2.2.2 Context Acquisition and Representation

Context acquisition and representation approaches are similar to implicit learning approaches in personalization systems (see section 2.1.3). These approaches are in general simpler than personalization profiling approaches. The main cause of this is that context acquisition techniques are focused normally within a single session. User profiling approaches have to couple with mining large amounts of information, having to be able to discern which concepts appear because of being part of the long-term interests of the user, which appear because of a periodical interest (e.g. summer holidays) and which appear due to some random factor (e.g. searching for a present for my cousin’s birthday). For instance, Anand and Mobasher (2007) illustrated this problem with an example where a user searches into Amazon.com for a pregnancy book for a girlfriend, just to have the system learn this topic as an “interest” and to receive future recommendations about the pregnancy topic. On the other hand, a contextual system would have correctly acquired the pregnancy topic, and, even if the system fails to apprehend the user context, errors on contextual system do not have such an impact, as the contextual profiles are usually discarded at the end of a session.

The most common approach for context acquisition and representation follows these steps: 1) depending on the concept of context of the system, obtain the set of documents and/or queries which represent the current context, 2) extract the most important or representative terms for these items, optionally weighted by some statistical metric (e.g. tf.idf) and 3) add these term to the contextual profile. The weighting metric can vary from not weighting the terms at all (Rhodes and Maes 2000) to a simple term frequency count (Leroy et al. 2003; Sugiyama et al. 2004) or a combination of the term frequency with the inverse document frequency of the term (i.e. tf.idf) (Bauer and Leake 2001; Billsus and Pazzani 2000; Dumais et al. 2003; Shen et al. 2005b). Other systems apply the same technique with different heuristics. Budzik and Hammond (1999, 2000) give more importance to terms that are at the beginning of a document.
or that have an emphasized style. Finkelstein et al. (2002) apply a clustering algorithm in order to find the most important terms. White and Kelly (2006) use the wpq approach for query expansion term weighting (Robertson 1990). This weighting function is based on “the probabilistic distribution of terms in relevant and non-relevant documents”. When the user submits a query or visits a document, it is considered as a new interaction of the session, and the terms appearing on the query or the visited document are considered as the observable items for the interaction. The context is thus dynamic and changes with every user interaction, giving more weight to those terms that have been observed more frequently on past interactions, and on interactions closer to the current one. They exploited the viewing time as an implicit relevance indicator, using a time threshold value to determine if a document was relevant or not. Interestingly, setting these threshold values accordingly to the task that the user had to perform proved to perform better than setting them accordingly to each user’s characteristics, which proved that the task information (i.e. the context) can be even more valuable than the overall information of the user. Finally, Shen et al (2005a) use a language model approach to weight the extracted terms.

Other approaches for context acquisition include classification techniques (Dou et al. 2007) (Haveliwala 2002; Vishnu 2005), where the extracted content is classified into a set of topics, storing this weighted topics as the contextual profile. Some approaches use semantic relations, Akrivas et al. (2002) use a term thesaurus to expand the semantic of the extracted terms in the query, and represent the context as the concepts in this thesaurus. Kraft et al. (2006) apply entity recognition techniques and store these entities as the user context. Finally, some approaches do not apply any type of acquisition approach, such as Bharat (2002) who only stores the clickthrough data and Melucci (2005), which example of ubiquitous system uses sensor data such as time or location.

2.2.3 Context Exploitation

Context exploitation share similarities with exploiting a user profile (see section 2.1.4). Like personalization approaches, some contextual approaches exploit the contextual profile to perform several query operations, in which the user query is modified (Kraft et al. 2006), expanded (Akrivas et al. 2002), or created as a context-related proactive query (Budzik and Hammond 1999), i.e. without interaction from the user. There are also document scoring techniques in which the results are reordered based on a document-context similarity (Sugiyama et al. 2004). However, there are also techniques that are only seen in contextual systems, such as
saving historical context representations in order to let the user revisit in the future this information (Bharat 2000).

**Query operations**

Query operations are the most common techniques in contextual systems. Normally the contextual information is exploited as a relevance feedback model, and applied to the current user’s query (Akrivas et al. 2002; Budzik and Hammond 2000; Finkelstein et al. 2002; Sugiyama et al. 2004; White and Kelly 2006). Finkelstein et al. (2002) combine this classic query expansion technique with a classification approach: if the expanded query can be classified into a topic, then a specific search engine can be used for retrieval. Leroy et al. (2003) compare this classic approach with a genetic-based exploitation approach, and also compare the impact of using negative feedback as a source of a “negative” context representation. Their conclusions were that "the genetic algorithm with negative expansion should be activated for low achievers but not for high achievers; relevance feedback should be activated for high achievers". Here the term ‘high achievers’ means users that perform better on typical retrieval tasks. Kraft et al. (2006) also complement the classic query expansion approach with an Iterative Filtering Meta-search (IFM), which it is basically the generation of multiple queries from the context vector and a final fusion of each of the search results returned by topic specific search engines.

Other systems do not follow the classic relevance feedback query reformulation model. For instance, the system presented by Melucci (2005) adjusts the vector space base according to the context information. In this case the launched query will be modified by this change of the vector base, thus being modified ultimately by the context information. Sen et al. (2005a) use the context representation in order to change the current language model representation of the user’s query, by modifying the document prior with the context and therefore combining the short-term contextual information with the current query.

**Context revisit**

Some systems use the context as a way of letting the user “recover” past contextual information, which could be useful to the current task at hand. Barhat (2000) stores the clickthrough information from past interactions of the user with the retrieval system. The goal is to let the user inspect all the visited documents from a past query, with information as the time spent viewing the document. This allows the user to revisit what could have found relevant in past contexts similar to the current. This concept was later applied by commercial search engines, such as Google and Yahoo!. Bauer and Leake (2001) apply the current context on an
information agent, capable to return resources that were used in past similar contexts, or to proactively retrieve Web results. The Stuff I’ve Seen system (Dumais et al. 2003) allows the user to easily revisit web pages, sent emails, or created documents, by exploiting an inverted index.

**Document scoring**

In document scoring approaches, a contextualized score is computed for a given document. Normally this contextualized score is given by a context-document similarity value. Again, classic vector space models are used to compute this similarity value, by calculating the cosine value between the term vector representation of the current context and the document (Rhodes and Maes 2000; Shen et al. 2005b). Shen et al (2005b) only apply this similarity value if the current query is similar enough to the current context. The novelty of this work is that the contextualization effects are not only applied to query search results, they rather have a more interactive nature: whenever the user returns to the result list after clicking through a result item, the initial result set is reordered, or more results are added through query expansion.

In classification-based approaches this similarity value is given by the vector similarity between the topic vector representation of the context and the topic vector representation of the document (Dou et al. 2007; Vishnu 2005). Haveliwala (2002) also exploits a topic-based representation, although in this case the document weights come from selecting a specific topic biased score.
Chapter 3

A Personalized Information Retrieval Model Based on Semantic Knowledge

Personalized retrieval widens the notion of information need to comprise implicit user needs, not directly conveyed by the user in terms of explicit information requests (Micarelli and Sciarrone 2004). Again, this involves modeling and capturing such user interests, and relating them to content semantics in order to predict the relevance of content objects, considering not only a specific user request but the overall needs of the user. When it comes to the representation of semantics (to describe content, user interest, or user requests), domain ontologies, as envisioned in the Knowledge Representation field (Gruber 1993), provide a highly expressive ground for describing units of meaning and a rich variety of interrelations among them. Ontologies achieve a reduction of ambiguity, and bring powerful inference schemes for reasoning and querying. Not surprisingly, there is a growing body of literature in the last few years that studies the use of ontologies to improve the effectiveness of information retrieval (Guha et al. 2003; Kiryakov et al. 2004; Stojanovic et al. 2003; Vallet et al. 2005) and personalized search (Gauch et al. 2003; Speretta and Gauch 2005). However, past work that claims the use of ontologies (Gauch et al. 2003; Kerschberg et al. 2001; Speretta and Gauch 2005) for user profile representation, does not fully exploit the variety of interrelations between concepts, but only the taxonomic relations, thus taking limited advantage of the potential for a finer semantic analysis (for knowledge propagation, inference, etc.), which is key in the approach here proposed. Our proposed personalization framework is set up in such a way that the models wholly benefit from the ontology-based grounding.

One of the aimed effects of our research is to achieve an improvement in the accuracy and reliability of personalization capabilities applied to IR, by way of a higher semantics awareness. Personalization can indeed enhance the subjective performance of retrieval, as perceived by users, and it is therefore a desirable feature in many situations, but it can easily be perceived as erratic and obtrusive if not handled adequately. Two key aspects to avoid such pitfalls are a) to appropriately manage the inevitable risk of error derived from the uncertainty of a formal
representation of users’ interests, and b) to correctly identify the situations where it is, or it is not appropriate to personalize, and to what extent (Castells et al. 2005).

As discussed in section 2.1, three main components can be distinguished in personalized IR systems: 1) user profile representation, capturing long-term preferences and interests of the user, 2) user profile acquisition, by which the user profile is obtained and updated, and 3) user profile exploitation, by virtue of which the retrieval system adapts itself to the user profile. Broadly speaking, information retrieval deals with modeling information needs, content semantics, and the relation between them (Salton and McGill 1986). The personalization system presented herein builds and exploits an explicit awareness of (meta)information about the user. We focus on providing an approach for user profile representation and exploitation; the user profile could be either directly provided by the user or implicitly evidenced along the history of her actions and feedbacks (see section 2.1.3).

3.1 Ontology-based User Profile Representation

Our personalization approach is based on an underlying user preference model in the form of conceptual user profiles (as opposed to e.g. sets of preferred documents or keywords), where user preferences are represented as a vector of weights (numbers from -1 to 1), corresponding to the intensity of the user interest for each concept in the ontology, where negative values indicate a dislike for a concept. We will therefore refer to the semantic preferences as a mapping \( P: U \rightarrow \mathcal{P} \) between the user \( u \in U \) and the set of preferences \( \mathcal{P} \), and we will denote it as \( P(u) \).

Since our system is focused on personalized retrieval, our user model is focused on the user interests. In our approach we use a quantitative model, in which user preferences are expressed over a set of concepts (e.g. “I like X”), in contrast with stating qualitative relations (e.g. “I prefer X over Y”) (Chomicki 2003). This follows the usual trend of personalized systems in IR (see section 2.1.1). In our model, the system can represent the user’s preferences over each concept defined in the domain ontology \( \mathcal{O} \), assigning a degree of preference or dislike to each concept.

Therefore, an instantiation of a set of preferences \( P(u) \) can be represented as the vector space \([-1,1]^{|\mathcal{O}|}\), which relates a preference weight \( P_x(u) \in [-1, 1] \) over each concept \( x \in \mathcal{O} \), i.e. defined within the domain ontology \( \mathcal{O} \).

Although, as stated, negative values (i.e. allowing the representation of dislikes) are supported by the presented system, these have to be treated cautiously, especially when the negative
values have been automatically generated by a profile acquisition module. Getting implicit or explicit feedback of the user to readjust the user profile is a common strategy in many personalized retrieval systems (Kelly and Teevan 2003; Rocchio and Salton 1971). While a wrong positive preference prediction can be amended that way, a negative weight for a concept may cause the system to low-rank every content that contains that concept, thus missing the chance to obtain feedback from the user for that concept and correct the incorrect prediction, since the concept will have difficulty to ever find its way into a result set. The system will thus have to handle negative preferences with caution. Possible approaches to follow when handling negative preferences include notifying the user when an inferred negative preference has had a significant impact on the current result set, or applying negative preferences to a lesser extent than positive ones. One example of the use of negative preferences is their exploitation as a possible mechanism of parental control, indicating a negative preference for violent content, and making the system filter out documents that match negative preferences.

Several advantages stem from a concept-based representation, in contrast to common keyword-based approaches:

- **Semantic richness.** Concept preferences are more precise and carry further (domain) knowledge than simple keyword terms. For instance, if a user states an interest for the keyword ‘Jaguar’, the system does not have further information to distinguish Jaguar, the wild animal from Jaguar, the car brand. A preference stated as ‘WildAnimal:Jaguar’ (this is read as “the instance Jaguar from the wild animal class) lets the system understand unambiguously the preference of the user, and also allows the use of more appropriate related semantics (e.g. synonyms, hyperonims, subsumption, etc.). This, together with disambiguation techniques, leads to the effective personalization of text-based and multimedia content.

- **Hierarchical representation.** Concepts in an ontology are represented in a hierarchical way, through different hierarchy properties (e.g. subClassOf, instanceOf, partOf, locatedIn, etc.). Parents, ancestors, children and descendants of a concept give valuable information about the concept’s semantics. For instance, the concept animal is highly enriched by each animal class semantics and a hypothetical taxonomy that the concept could subsume.

- **Inference.** Ontology standards, such as RDF\textsuperscript{12} and OWL\textsuperscript{13}, support inference mechanisms that can be exploited by the system to further enhance personalization, so

\textsuperscript{12} http://www.w3.org/RDF/
that, for instance, a user interested in animals (superclass of cat) is also recommended items about cats. Inversely, a user interested in lizards, snakes, and chameleons can be inferred to be globally interested in reptiles with a certain confidence. Also, a user keen of Sicily can be assumed to like Palermo, through the transitive ‘locatedIn’ relation.

The ontology-based representation of user interests is richer, more precise, and less ambiguous than a keyword-based or item-based model. It provides an adequate grounding for the representation of coarse to fine-grained user interests (e.g. interest for broad topics, such as football, sci-fi movies, or the NASDAQ stock market, vs. preference for individual items such as a sports team, an actor, a stock value), and can be a key enabler to deal with the subtleties of user preferences, such as their dynamic, context-dependent relevance. An ontology provides further formal, computer-processable meaning on the concepts (who is coaching a soccer team, an actor’s filmography, financial data on a stock), and makes it available for the personalization system to take advantage of.

Figure 3.1. User preferences as concepts in an ontology.

Figure 3.1 presents an example of conceptualized user preferences. Let’s suppose a user indicates an interest about the topic ‘Leisure’. The system is then able to infer preferences for ‘Leisure’ sub-topics, obtaining finer grain details about the user preference. Note that original and more specific preferences will prevail over the system’s inferences. In this case the user is not interested in modern music, which prevails over the higher-level topic inference.

13 [http://www.w3.org/TR/owl-features/]
Not only hierarchical properties can be exploited for preference inference. Assuming that the user has a preference for the ‘USA’ region, the properties ‘visit’ and ‘locatedIn’ could be exploited in order for the system to infer new preferences. Firstly, the system could use the ‘visit’ property in order to infer that the user could be interested in islands in general (assuming that the user has a sufficient degree of interest for ‘Island Travel’). Secondly, given that the user is interested in the ‘USA’ region, the system could specify the inferred interest for islands in this region, therefore considering, for instance, a potential interest for the ‘Hawaii’ islands. In this case a Hawaii tourist guide would be likely to have a positive value (even though low and tentative) of preference for the user. More details on preference expansion are given in section 4.5.3.

### 3.2 A Semantic Approach for User Profile Exploitation

Exploiting user profiles involves using this information in order to adapt the IR system. The goals addressed so far have been focused on delivering preference-based improvements for content filtering and retrieval, in a way that they can be very easily introduced to support the retrieval functionalities, such as searching, browsing, and recommending.

The personalization system assumes that the items in a retrieval space \( \mathcal{D} \) are annotated, based on which semantic index is built (Castells et al. 2007; Kiryakov et al. 2004), i.e. content is associated to weighted semantic metadata which describes the meaning carried by the item, in terms of a domain ontology \( \mathcal{O} \). That is, each item \( d \in \mathcal{D} \) is associated to a set of semantic annotations through a mapping function \( M: \mathcal{D} \to [0,1]^{\mathcal{O}} \), denoted as \( M(d) \in [0,1]^{\mathcal{O}} \).

Therefore, for each \( x \in \mathcal{O} \), the weight \( M_x(d) \) indicates the degree to which the concept \( x \) is important in the meaning of \( d \). Thus, as shown in Figure 3.2, there is a fuzzy relationship between users and the indexed content of the system, through the ontology layer. Although the use of this ontology layer is transparent to the user, the system can take advantage of an ontological representation of user preferences: unambiguous, richer relations and inference capabilities (see 3.1). Based on preference weights, measures of user interest for content units can be computed, with which it is possible to discriminate, prioritize, filter and rank contents (a collection, a catalog section, a search result) in a personalized way.
Figure 3.2. Links between user preferences and search space.

A fundamental notion for this purpose is the definition of a measure of content relevance for the interests of a particular user. Building on this measure, specific document scoring algorithms can then be developed for filtering and sorting a list of content items, and re-ranking search results, according to user preferences.

The basis for the personalization of content retrieval is the definition of a matching algorithm that provides a personal relevance measure (PRM), according to her semantic preferences. The measure is computed as a function of the semantic preferences of the user $u$, and the semantic metadata of the document $d$, and is thus denoted as $\text{prm}(d,u)$. In this calculation, user preferences $P(u)$ and content metadata $M(d)$ can be represented as two vectors in an $N$-dimensional vector space, where $N$ is the number of elements in the universe $O$ of ontology concepts, and the coordinates of the vectors are the weights assigned to ontology concepts in user preferences and document annotations, representing respectively the intensity of preferences and the degree of importance for the document. Semantic preferences also include inferred preferences, using for example deductive inference, so if a user expresses preference for the animal concept, preferences for each subclass of animal (i.e. `Bird` concept) would be inferred (for more information see section 4.5.3).
The procedure for matching these vectors has been primarily based on a cosine function for vector similarity computation, as follows:

\[
prm(d,u) = \cos(P(u), M(d)) = \frac{P(u) \cdot M(d)}{|P(u)| \times |M(d)|} = \frac{\sum_{t=1}^{T} (P_x(u) \times M_x(d))}{\sqrt{\sum_{t=1}^{T} (P_x^2(u))} \times \sqrt{\sum_{t=1}^{T} (M_x^2(d))}}
\]

**Equation 1.** Personal Relevance Measure calculation.

Figure 3.3 shows a visual representation of the similarity between the semantic preference and metadata vectors, reduced to a three-dimension space for simplicity. As in the classic IR vector-space model (Ricardo and Berthier 1999), information expressed in terms of vectors are more alike the closer the vector are represented in the finite-dimensional space. In classic IR, one vector represents the query and the other matching vectors are the representation of the documents. In our representation, the first vector is the user preference, whereas the second vector is also essentially the representation of the content in the system’s search space.

Figure 3.3. Visual representation of metadata and preference's vector similarity

The PRM algorithm matches two concept-weighted vectors and produces a value between [-1, 1]. Values near -1 indicate that the preferences match negatively the content metadata (i.e. two vectors are dissimilar), values near 1 indicate that the user interests do match the content. Since the annotated content is considered an external resource by our model, and we wish to make as few assumptions as possible about it, our approach assumes that the annotations may lack weights, or even a clear weighting criterion, as is sometimes the case. In such situation, the PRM function assigns a weight of 1 by default to all metadata. Even so, it is considered worth to keep weights in the annotations, for homogeneity and reusability (see e.g. (Vallet et al. 2005)).
Figure 3.4 shows an example of the computation of the preference value, in a simplified setting where $O = \{\text{Flower, Dog, Sea, Surf, Beach, Industry}\}$ is the set of all domain ontology terms (classes and instances). According to her profile, the user is interested in the concepts of ‘Flower’, ‘Surf’, and ‘Dog’, with different intensity, and has a negative preference for ‘Industry’. Hence, the preference vector for this user is $P(u') = (0.7, 1.0, 0.0, 0.8, 0.2, -0.7)$. A still image is annotated with the concepts of ‘Dog’, ‘Sea’, ‘Surf’ and ‘Beach’, therefore the corresponding metadata vector is $M(d') = (0.0, 0.8, 0.6, 0.8, 0.2, 0.0)$.

The PRM of the still image for this user shall therefore be:

$$\text{prm}(d', u') = \frac{(0.7 \times 0.0) + (1.0 \times 0.8) + (0.0 \times 0.6) + (0.8 \times 0.8) + (0.0 \times 0.2) + (-0.7 \times 0.0)}{\sqrt{0.7^2 + 1^2 + 0.0^2 + 0.8^2 + 0.0^2 + (-0.7)^2 \times 0.0^2 + 0.8^2 + 0.6^2 + 0.8^2 + 0.2^2 + 0.0^2}} \approx 0.69$$

This measure can be combined with the relevance measures computed by the user-neutral algorithms, producing a personalized bias on the ranking of search results. Search personalization is primarily applied in our system based on the weighting scheme (see document scoring on section 2.1.4), and the computations described in the previous section. This model enables different personalization actions, such as cutting down (i.e. filtering) search results, reordering the results or providing personalized forms of content navigation support. In such
personalization features, the PRM measure described in the preceding section acts as the
personalization score, i.e. the user-document similarity value.

Personalization of search must be handled carefully. An excessive personal bias may drive
results too far from the actual query. For this reason, we have avoided query reformulation
techniques, favoring document scoring techniques instead, such as user personalized filtering
and result reordering as a post process to the execution of queries, in such a way that the
intensity of the personalization effect can be more easily controlled. In this spirit, the
personalized score defined by the PRM values has to be combined with the query-dependent
rank (QDR) values returned by the intelligent retrieval modules. That is, the final combined
rank (CR) of a document \(d\), given a user \(u\) and her query \(q\) is defined as a function of both
values:

\[
CR(d, q, u) = f(prm(d, u), QDR(d, q))
\]

**Equation 2.** Final personalized Combined Rank.

The question remains as to how both values should be combined and balanced. As an initial
solution, we use a linear combination of both:

\[
CR(d, q, u) = \lambda \cdot prm(d, u) + (1 - \lambda) \cdot QDR(d, q)
\]

**Equation 3.** Linear combination of PRM and QDR.

where the value of \(\lambda\), between 0 and 1, determines the **degree of personalization** of the
subsequent search ranking.

What is an appropriate value for \(\lambda\), how it should be set, and whether other functions different
from a linear combination would perform better, are work in progress in this task, but some
initial solutions have been outlined (Castells et al. 2005). Explicit user requests, queries and
indications should always take precedence over system-learned user preferences.

Personalization should only be used to “fill the gaps” left by the user in the information she
provides, and always when the user is willing to be helped this way. Therefore, the larger the
gap, the more room for personalization. In other words, the degree of personalization \(\lambda\) can be
proportional to the size of this gap. One possible criterion to estimate this gap is by measuring
the specificity of the query. This can be estimated by measuring the generality of the query
terms (e.g. by analyzing the generality of a term within a term thesaurus), the number of results,
or the closeness of rank values. For instance, a search for the topic ‘Sports’ is rather generic, has
a large number of related subtopics, a large number of concepts are related to this topic, and a
query for ‘Sports’ would probably return thousands of content items (of course this depends on
the repository). It therefore leaves quite some room for personalization, which would be a reason for raising $\lambda$ in this case.

Ultimately, personalized ranking, as supported by the adapted IR system, should leave degree of personalization as an optional parameter, so it could be set by the user herself, as in the former Google personalized web search application (see section 2.1.5). See also (Dwork et al. 2001; Fernández et al. 2006; Lee 1997; Manmatha et al. 2001; Renda and Straccia 2003; Vogt and Cottrell 1999) for state of the art on combining rank sources.

Building on the combined relevance measure described above, a personalized ranking is defined, which will be used as the similarity measure for the result reordering.

The personal relevance measure can also be used to filter and order lists of documents while browsing. In this case the room for personalization is higher, in general, when compared to search, since browsing requests are usually more unspecific than search queries. Moreover, browsing requests, viewed as light queries, typically consist of Boolean filtering conditions (e.g. filter by date or category), and strict orderings (by title, author, date, etc.). If any fuzzy filters are defined (e.g. when browsing by category, contents might have fuzzy degrees of membership to category), the personalization control issues described above would also apply here. Otherwise, personalization can take over ranking all by itself (again, if requested by the user).

On the other hand, the PRM measure, combined with the advanced browsing techniques provides the basis for powerful personalized visual clues. Any content highlighting technique can be played to the benefit of personalization, such as the size of visual representations (bigger means more relevant), color scale (e.g. closer to red means more interesting), position in 3D space (foreground vs. background), automatic hyperlinks (to interesting contents), etc.
Chapter 4

Personalization in Context

Specific, advanced mechanisms need to be developed in order to ensure that personalization is used at the right time, in the appropriate direction, and in the right amount. Users seem inclined to rely on personalized features when they need to save time, wish to spare efforts, have vague needs, have limited knowledge of what can be queried for (e.g. for lack of familiarity with a repository, or with the querying system itself), or are not aware of recent content updates. Personalization is clearly not appropriate, for instance, when the user is looking for a specific, known content item, or when the user is willing to provide detailed relevance feedback, engaging in a more conscientious interactive search session. Even when personalization is appropriate, user preferences are heterogeneous, variable, and context-dependent. Furthermore, there is inherent uncertainty in the system when automatic preference learning is used. To be accurate, personalization needs to combine long-term predictive capabilities, based on past usage history, with shorter-term prediction, based on current user activity, as well as reaction to (implicit or explicit) user feedback to personalized output, in order to correct the system’s assumptions when needed.

The idea of contextual personalization, proposed and developed here, responds to the fact that human preferences are multiple, heterogeneous, changing, and even contradictory, and should be understood in context with the user goals and tasks at hand. Indeed, not all user preferences are relevant in all situations. For instance, if a user is consistently looking for some contents in the Formula 1 domain, it would not make much sense that the system prioritizes some Formula 1 picture with a helicopter in the background, as more relevant than others, just because the user happens to have a general interest for aircrafts. In the semantic realm of Formula 1, aircrafts are out of (or at least far from) context. Taking into account further contextual information, available from prior sets of user actions, the system can provide an undisturbed, clear view of the actual user’s history and preferences, cleaned from extraordinary anomalies, distractions or “noise” preferences. We refer to this surrounding information as contextual knowledge or just context, offering significant aid in the personalization process. The effect and utility of the proposed techniques consists of endowing a personalized retrieval system with the capability to
filter and focus its knowledge about user preferences on the semantic context of ongoing user activities, so as to achieve coherence with the thematic scope of user actions at runtime.

As already discussed in the background section of this work, context is a difficult notion to grasp and capture in a software system. In our approach, we focus our efforts on this major topic of retrieval systems, by restricting it to the notion of semantic runtime context. The latter forms a part of general context, suitable for analysis in personalization and can be defined as the background themes under which user activities occur within a given unit of time. We therefore refer to this semantic runtime context as the information related to personalization tasks, using the simplified term context for it. The problems to be addressed include how to represent the context, how to determine it at runtime (acquisition), and how to use it to influence the activation of user preferences, "contextualize" them and predict or take into account the drift of preferences over time (short and long-term).

As will be described in section 4.3, in our current solution to these problems, a runtime context is represented as (is approximated by) a set of weighted concepts from the domain ontology. How this set is determined, updated, and interpreted, will be explained in section 4.4. Our approach to the contextual activation of preferences is then based on a computation of the semantic similarity between each user preference and the set of concepts in the context, as will be shown in section 4.5.1. In spirit, the approach tries to find semantic paths linking preferences to context. The considered paths are made of existing semantic relations between concepts in the domain ontology. The shorter, stronger, and more numerous such connecting paths, the more in context a preference shall be considered.

The proposed techniques to find these paths take advantage of a form of Constraint Spreading Activation (CSA) strategy (Crestani 1997), as will be explained in section 4.5. In the proposed approach, a semantic expansion of both user preferences and the context takes place, during which the involved concepts are assigned preference weights and contextual weights, which decay as the expansion grows farther from the initial sets. This process can also be understood as finding a sort of fuzzy semantic intersection between user preferences and the semantic runtime context, where the final computed weight of each concept represents the degree to which it belongs to each set.

Finally, the perceived effect of contextualization should be that user interests that are out of focus, under a given context, shall be disregarded, and only those that are in the semantic scope of the ongoing user activity (the "intersection" of user preferences and runtime context) will be considered for personalization. As suggested above, the inclusion or exclusion of preferences needs may not be binary, but may range on a continuum scale instead, where the contextual
weight of a preference shall decrease monotonically with the semantic distance between the preference and the context.

### 4.1 Notation

Before continuing, we provide a few details on the mathematical notation that will be used in the sequel. It will be explained again in most cases when it is introduced, but we gather it all here, in a single place, for the reader’s convenience.

- $\mathcal{O}$: The domain ontology (i.e. the concept space).
- $\mathcal{R}$: The set of all relations in $\mathcal{O}$.
- $\mathcal{D}$: The set of all documents or content in the search space.
- $M : \mathcal{D} \to [0,1]^{\mathcal{C}}$: A mapping between document and their semantic annotations, i.e. $M(d) \in [0,1]^{\mathcal{O}}$ is the particular metadata instance of a document $d \in \mathcal{D}$.
- $\mathcal{U}$: The set of all users.
- $\mathcal{P}$: The set of all possible user preferences.
- $\mathcal{C}$: The set of all possible contexts.
- $\mathcal{P}_\mathcal{O}, \mathcal{C}_\mathcal{O}$: An instantiation of $\mathcal{P}$ and $\mathcal{C}$ for the domain $\mathcal{O}$, where $\mathcal{P}$ is represented by the vector-space $[-1,1]^{\mathcal{O}}$ and $\mathcal{C}$ by $[0,1]^{\mathcal{O}}$.
- $P : \mathcal{U} \to \mathcal{P}$: A mapping between users and preferences, i.e. $P(u) \in \mathcal{P}$ is the preference of user $u \in \mathcal{U}$.
- $C : \mathcal{U} \times \mathbb{N} \to \mathcal{C}$: A mapping between users and contexts over time, i.e. $C(u,t) \in \mathcal{C}$ is the context of a user $u \in \mathcal{U}$ at an instant $t \in \mathbb{N}$.\(^{14}\)
- $EP : \mathcal{U} \to \mathcal{P}$: Extended user preferences.
- $EC : \mathcal{U} \times \mathbb{N} \to \mathcal{C}$: Extended context.
- $CP : \mathcal{U} \times \mathbb{N} \to \mathcal{P}$: Contextualized user preferences, also denoted as $\Phi(P(u),C(u,t))$.

\(^{14}\) In order to ease notation, we will refer to time as a discreet incremental set of natural values $\{1,2,\ldots,n\}$, alternatively represented as $\{t_1, t_2,\ldots, t_n\}$, where each value of $t$ will determine an instant in which a relevant event occurred. A value of $t_{n+i} > t_n$, $i > 0$ means that the event identified $t_{n+i}$ occurred closer in time than $t_n$. 


where \( v \in [-1,1]^{\Omega} \)

We shall use this vector notation for concept-vector spaces, where the concepts of an ontology \( \Omega \) are the axis of the vector space. For a vector \( \vec{v} \in [-1,1]^{\Omega} \), \( v_x \in [-1,1] \) is the coordinate of \( v \) corresponding to the concept \( x \in \Omega \). This notation will be used for all the elements ranging in the \([-1,1]^{\Omega} \) space, such as document metadata \( M_x(d) \), user preferences \( P_x(u) \), runtime context \( C_x(u,t) \), and others.

\[ Q \]

The set of all possible user requests, such as queries, viewing documents, or browsing actions.

\( \text{prm} : \mathcal{D} \times U \times \mathbb{N} \rightarrow [-1,1] \)

\( \text{prm}(d,u,t) \) is the estimated contextual interest of user \( u \) for the document \( d \) at instant \( t \).

\( \text{sim} : \mathcal{D} \times Q \rightarrow [0,1] \)

\( \text{sim}(d,q) \) is the relevance score computed for the document \( d \) for a request \( q \) by a retrieval system external to the personalization system.

\( \text{score} : \mathcal{D} \times Q \times U \times \mathbb{N} \rightarrow [-1,1] \)

\( \text{score}(d,q,u,t) \) is the final personalized relevance score computed by a combination of \( \text{sim} \) and \( \text{prm} \).

### 4.2 Preliminaries

Our strategies for the dynamic contextualization of user preferences are based on three basic principles: a) the representation of context as a set of domain ontology concepts that the user has “touched” or followed in some manner during a session, b) the extension of this representation of context by using explicit semantic relations among concepts represented in the ontology, and c) the extension of user preferences by a similar principle. Roughly speaking, the “intersection” of these two sets of concepts, with combined weights, will be taken as the user preferences of interest under the current focus of user action. The ontology-based extension mechanisms will be formalized on the basis of an approximation to conditional probabilities, derived from the existence of relations between concepts. Before the models and mechanisms are explained in detail, some preliminary ground for the calculation of combined probabilities will be provided and shall be used in the sequel for our computations.

Given a finite set \( \Omega \), and \( a \in \Omega \), let \( P(a) \) be the probability that \( a \) holds some condition. It can be shown that the probability that \( a \) holds some condition, and it is not the only element in \( \Omega \) that holds the condition, can be written as:

\[
P \left( a \cap \bigcup_{x \in \Omega - \{a\}} x \right) = \sum_{\delta \subset \Omega - \{a\}} \left\{ (-1)^{|\delta|+1} \prod_{x \in \delta} P(x) \cdot P(a|x) \right\}
\]

**Equation 4.** Probability of holding condition \( a \), inside a finite set \( \Omega \).
Provided that $a \cap x$ are mutually independent for all $x \in \Omega$ (the right hand-side of the above formula is based on the inclusion-exclusion principle applied to probability (Whitworth 1965)). Furthermore, if we can assume that the probability that $a$ is the only element in $\Omega$ that holds the condition, then the previous expression is equal to $P(a)$.

We shall use this form of estimating “the probability that $a$ holds some condition” with two purposes: a) to extend user preferences for ontology concepts, and b) to determine what parts of user preferences are relevant for a given runtime context, and should therefore be activated to personalize the results (the ranking) of semantic retrieval, as part of the process described in detail by Crestani (1997). In the former case, the condition will be “the user is interested in concept $a$”, that is, $P(a)$ will be interpreted as the probability that the user is interested in concept $a$ of the ontology. In the latter case, the condition will be “concept $a$ is relevant in the current context”. In both cases, the universe $\Omega$ will correspond to a domain ontology $O$ (the universe of all concepts).

Equation 4 provides a basis for estimating $P(a)$ for all $a \in O$ from an initial set of concepts $x$ for which we know (or we have an estimation of) $P(x)$. In the case of preferences, this set is the initial set of weighted user preferences for ontology concepts, where concept weights are interpreted as the probability that the user is interested in a concept. In the case of contextual relevant concepts, the initial set is a weighted set of concepts found in elements (links, documents, queries) involved in user actions in the span of a retrieval session. Here this set is taken as a representation of the semantic runtime context, where weights represent the estimated probability that such concepts are important in user goals. In both cases, Equation 4 will be used to implement an expansion algorithm that will compute probabilities (weights) for all concepts starting from the initially known (or assumed) probabilities for the initial set. In the second case, the algorithm will compute a context relevance probability for preferred concepts that will be used as the degree of activation that each preference shall have. Put in rough terms, the (fuzzy) intersection of context and preferences will be found with this approach.

Equation 4 has some interesting properties with regards to the design of algorithms based on it. In general, for $X=\{x_i\}_{i=0}^n$, where $x_i \in [0,1]$, let us define $R(X)$:

$$R(X) = \sum_{\delta \in \Pi_n} \left( (-1)^{|\delta|+1} \times \prod_{i \in \delta} x_i \right)$$

**Equation 5.** Probability of holding a condition for over a set of independent variables.
It is easy to see that this function has the following properties:

- \( R(X) \in [0,1] \)
- \( R(\emptyset) = 0 \)
- \( R(X) \geq x_i \) for all \( i \) (in particular this means that \( R(X) = 1 \) if \( x_i = 1 \) for some \( i \)).
- \( R(X) \) increases monotonically with respect to the value of \( x_i \), for all \( i \).
- \( R(X) \) can be defined recursively as: \( R(X) = x_0 + (1 - x_0) \cdot R([x_i]_{i=1}^n) \). \( R(X) \) can thus be computed efficiently. Note also that \( R(X) \) does not vary if we reorder \( X \).

These properties will be useful for the definition of algorithms with computational purposes.

Note that the properties of \( R(X) \) can only be in general satisfied if \( x_i \in [0,1] \). Let us suppose now that we are using \( R(X) \) for the estimation of the set of preferences \( P(a) \), given an initial set \( P(x) \). We have defined \( P(x) \in [-1,1] \), \( P(a) \in [-1,1] \). While the subset of positive preferences \( P^+(x) \in [0,1] \) does satisfy the restriction of \( R(X) \), the subset of negative preferences, i.e. the subset \( P^-(x) \in [-1,0) \) does not. Furthermore, negative preference values pervert Equation 4, as it is based on pure probability computation. For solving this we can redefine \( P^-(x) \) as \( \tilde{P}^-(x) = 1 + \left( P^-(x) \right) \in (0,1] \) as the probability of disliking the concept \( x \). We can estimate the final value of preferences \( P(a) \) as the following combination:

\[
P(a) = R(P^+(x)) - R(\tilde{P}^-(x))
\]

Equation 6. Independent calculation of negative and positive preferences.

That is, we will calculate separately the estimation of preferences for the probability of liking a concept \( x \) and the probability of disliking it. The final \( P(a) \in [-1,1] \) would be the result of subtracting the probability of disliking a concept \( x \) to the probability of liking it. We therefore adopt the simplistic approach of assuming that the probabilities of liking and disliking a concept are antagonistic.

### 4.3 Semantic Context for Personalized Content Retrieval

Our model for context-based personalization can be formalized in an abstract way as follows, without any assumption on how preferences and context are represented. Let \( \mathcal{U} \) be the set of all users, let \( \mathcal{C} \) be the set of all contexts, and \( \mathcal{P} \) the set of all possible user preferences. Since each user will have different preferences, let \( P : \mathcal{U} \rightarrow \mathcal{P} \) map each user to her preferences. Similarly, each user is related to a different context at each step in a session with the system, which we shall represent by a mapping \( C : \mathcal{U} \times \mathbb{N} \rightarrow \mathcal{C} \), since we assume that the context evolves over
time. Thus we shall often refer to the elements from \( \mathcal{P} \) and \( \mathcal{C} \) as in the form \( P(u) \) and \( C(u, t) \) respectively, where \( u \in \mathcal{U} \) and \( t \in \mathbb{N} \).

**Definition 1.** Let \( \mathcal{C} \) be the set of all contexts, and let \( \mathcal{P} \) be the set of all possible user preferences. We define the **contextualization of preferences** as a mapping \( \Phi : \mathcal{P} \times \mathcal{C} \rightarrow \mathcal{P} \) so that for all \( p \in \mathcal{P} \) and \( c \in \mathcal{C} \), \( p \models \Phi(p,c) \).

In this context the entailment \( p \models q \) means that any consequence that could be inferred from \( q \) could also be inferred from \( p \). For instance, given a user \( u \in \mathcal{U} \), if \( P(u) = q \) implies that \( u \) “likes/dislikes \( x \)” (whatever this means), then \( u \) would also “like \( x \)” if her preference was \( p \).

Now we can particularize the above definition for a specific representation of preference and context. In our model, we consider user preferences as the weighted set of domain ontology concepts for which the user has an interest, where the intensity of interest can range from -1 to 1.

**Definition 2.** Given a domain ontology \( \mathcal{O} \), we define the set of all preferences over \( \mathcal{O} \) as \( \mathcal{P}_\mathcal{O} = [-1,1]^{\mathcal{O}} \), where given \( p \in \mathcal{P}_\mathcal{O} \), the value \( p_x \) represents the preference intensity for a concept \( x \in \mathcal{O} \) in the ontology.

**Definition 3.** Under the above definitions, we particularize \( \models_\mathcal{O} \) as follows: given \( p, q \in \mathcal{P}_\mathcal{O} \), \( p \models_\mathcal{O} q \iff \forall x \in \mathcal{O} \), either \( q_x \leq p_x \), or \( q_x \) can be deduced from \( p \) using consistent preference extension rules over \( \mathcal{O} \).

Now, our particular notion of context is that of the **semantic runtime context**, which we define as the background themes under which user activities occur within a given unit of time.

**Definition 4.** Given a domain ontology \( \mathcal{O} \), we define the set of all semantic runtime contexts as \( \mathcal{C}_\mathcal{O} = [0,1]^{\mathcal{O}} \).

With this definition, a context is represented as a vector of weights denoting the degree to which a concept is related to the current activities (tasks, goals, short-term needs) of the user.

In the next sections, we define a method to build the values of \( C(u,t) \) during a user session, a model to define \( \Phi \), and the techniques to compute it. Once we define this, the activated user preferences in a given context will be given by \( \Phi(P(u), C(u, t)) \).
4.4 Capturing the Context

Previously analyzed implementation-level representation of semantic runtime context is defined as a set of concepts that have been involved, directly or indirectly, in the interaction of a user $u$ with the system during a retrieval session. Therefore, at each point $t$ in time, context can be represented as a vector $C(u,t)\in[0,1]^O$ of concept weights, where each $x\in O$ is assigned a weight $C_x(u,t)\in[0,1]$. This context value may be interpreted as the probability that $x$ is relevant for the current semantic context. Additionally, time is measured by the number of user requests within a session. Since the fact that the context is relative to a user is clear, in the following we shall often omit this variable and use $C(t)$, or even $C$ for short, as long as the meaning is clear.

In our approach, $C(t)$ is built as a cumulative combination of the concepts involved in successive user requests, in such a way that the importance of concepts fades away with time. This simulates a drift of concepts over time, and a general approach towards achieving this follows. This notion of context extraction is extracted from the implicit feedback area (White et al. 2005b), concretely our model is part of the ostensive models, as one that uses a time variable and gives more importance to items occurring close in time (Campbell and van Rijsbergen 1996).

Right after each user’s request, a request vector $Req(t)\in C_o$ is defined. This vector may be:

- The query concept-vector, if the request is a query.
- A concept-vector containing the topmost relevant concepts in a document, if the request is a “view document” request.
- The average concept-vector corresponding to a set of documents marked as relevant by the user, if the request is a relevance feedback step.
- If the request is a “browse the documents under a category $c$” request, $Req(t)$ is the sum of the vector representation of the topic $c$ (in the $[0,1]^O$ concept vector-space), plus the normalized sum of the metadata vectors of all the content items belonging to this category.

As the next step, an initial context vector $C(t)$ is defined by combining the newly constructed request vector $Req(t)$ from the previous step with the context $C(t-1)$, where the context weights computed in the previous step are automatically reduced by a decay factor $\xi$, a real value in $[0,1]$, where $\xi$ may be the same for all $x$, or a function of the concept or its state. Consequently, at a given time $t$, we update $C_x(t)$ as
\[ C_x(t) = \xi \cdot C_x(t-1) + (1 - \xi) \cdot \text{Req}_x(t) \]

Equation 7. Runtime semantic context.

The decay factor will define for how many action units will be a concept considered, and how fast a concept will be “forgotten” by the system. This may seem similar to pseudo-relevance feedback, but it is not used to reformulate the query, but to focus the preference vector as shown in the next section.

4.5 Semantic Expansion of Context and Preferences

The selective activation of user preferences is based on an approximation to conditional probabilities: given \( x \in O \) with \( P_x(u) \in [-1,1] \) for some \( u \in U \), i.e. a concept on which a user \( u \) has some interest/dislike, the probability that \( x \) is relevant for the context can be expressed in terms of the probability that \( x \) and each concept \( y \) directly related to \( x \) in the ontology belong to the same topic, and the probability that \( y \) is relevant for the context. With this formulation, the relevance of \( x \) for the context can be computed by a constrained spreading activation algorithm, starting with the initial set of context concepts defined by \( C \).

Our strategy is based on weighting each semantic relation \( r \) in the ontology with a measure \( w(r,x,y) \) that represents the probability that given the fact that \( r(x,y) \), \( x \) and \( y \) belong to the same topic. As explained above, we will use this as a criteria for estimating the certainty that \( y \) is relevant for the context if \( x \) is relevant for the context, i.e. \( w(r,x,y) \) will be interpreted as the probability that a concept \( y \) is relevant for the current context if we know that a concept \( x \) is in the context, and \( r(x,y) \) holds. Based on this measure, we define an algorithm to expand the set of context concepts through semantic relations in the ontology, using a constrained spreading activation strategy over the semantic network defined by these relations. As a result of this strategy, the initial context \( C(t) \) is expanded to a larger context vector \( EC(t) \), where of course \( EC_x(t) \geq C_x(t) \) for all \( x \in O \).
Let $\mathcal{R}$ be the set of all relations in $\mathcal{O}$, let $\tilde{\mathcal{R}} = \mathcal{R} \cup \mathcal{R}^{-1} = \mathcal{R} \cup \{ r^{-1} \mid r \in \mathcal{R} \}$, and $w : \tilde{\mathcal{R}} \rightarrow [0,1]$. The extended context vector $EC(t)$ is computed by:

$$EC_y(t) = \begin{cases} 
  C_y(t) & \text{if } C_y(t) > 0 \\
  R \left( \prod_{x \in \mathcal{O}, r \in \mathcal{R}, r(x,y)} EC_x(t) \cdot w(r, x, y) \cdot \text{power}(x) \right) & \text{otherwise} 
\end{cases}$$

Equation 8. Expanded context vector

where $R(X)$ is defined by equation 5, $\mathcal{R}$ is the set of all concept relations in the ontology $\mathcal{O}$ and $\mathcal{R}^{-1}$ is the set of all inverse relations of $\mathcal{R}$, i.e. a concept $x$ has an inverse relation $r^{-1}(x,y) \iff \{\exists r(y,x) \mid r \in \mathcal{R}\}$. Finally, $\text{power}(x) \in [0,1]$ is a propagation power assigned to each concept $x$ (by default, $\text{power}(x) = 1$). Note that we are explicitly excluding the propagation between concepts in the input context (i.e. these remain unchanged after propagation). The personalization system assumes that the relation weights $\mathcal{R}$ and $\mathcal{R}^{-1}$ and propagation power values are already assigned. In our experimental setup (see section 5.2), $\mathcal{R}$ and $\mathcal{R}^{-1}$ was tuned up manually. There are techniques that can be applied in order to calculate $\mathcal{R}$ and $\mathcal{R}^{-1}$ automatically, and even making these relations dependable on the linked concepts. Different measures can be used, such as the popularity of a concept (similar to PageRank), distance between the concepts, subsumption (level in the hierarchy), rarity of the property, or trust for the property/concept (Aleman-Meza et al. 2005). In this setup, the number of usable properties was rather low, so the manual approach was straightforward, and thus we decided to leave the automatic computation out of scope of our evaluation, although in a larger or richer KB a manual setting could be preferred. The values of propagation power were left to 1.

### 4.5.1 Spreading Activation Algorithm

The algorithms for expanding preferences and context will be based on the so called Constrained Spreading Activation (CSA) strategy (see e.g. (Crestani 1997; Crestani and Lee 1999; Crestani and Lee 2000)). The first work on CSA was developed by Salton and Buckley (Salton and Buckley 1988). Another relevant reference is (Rocha et al. 2004), where CSA is used to improve the recall of a retrieval system using domain ontologies.
Based on definition 2, $EC(t)$ can be computed as follows, where $C_{0}(t) = \{ x \in \mathcal{O} \mid C_{x}(t) > 0 \}$ is the initial updated input with new context values resulting after the current request. Given $x \in \mathcal{O}$, we define the semantic neighborhood of $x$ as $N[x] = \{ (r, y) \in \mathcal{R} \times \mathcal{O} \mid r(x, y) \}$.

This algorithm can also be used as a standalone method for expanding preferences (i.e. compute the $EP$ vector from the initial $P$), except that time is not a variable, and a different measure $w$ is used. Figure 4.1 shows a simple pseudocode of the algorithm.

```
expand (C, EC, w)
    for $x \in \mathcal{O}$ do
        $EC_{x} = C_{x}$ // Initialization of Expanded Context
        in_path[x] ← false
    expand (x, w, 0)

expand (x, w, prev_cx)
    in_path[x] ← true
    for $(r, y) \in N[x]$ do // Optimization: choose $(r, y)$ in decreasing order of $EP_{y}$
        if not in_path[y] and $C_{y} = 0$ and $EC_{y} < 1$ then // The latter condition to save some work
            prev_cy ← $EC_{y}$
            // Undo last update from $x$
            $EC_{y} \leftarrow (EC_{y} - w(r, x, y) \ast power(x) \ast prev_cx) / (1 - w(r, x, y) \ast power(x) \ast prev_cx)$
            $EC_{y} \leftarrow EC_{y} + (1 - EC_{y}) \ast w(r, x, y) \ast power(x) \ast EC_{x}$
            if $EC_{y} > \varepsilon$ then expand (y, w, prev_cy)
            in_path[x] ← false
```

**Figure 4.1.** Simple version of the spreading activation algorithm.

To exemplify the expansion process, Figure 4.2 shows a simple preference expansion process, where three concepts are involved. The user has preferences for two of these concepts, which are related to a third through two different ontology relations. The expansion process show how a third preference can be inferred, “accumulating” the evidences of preference from the original two preferences.
The simple expansion algorithm can be optimized as follows, by using a priority queue (a heap H) w.r.t. $EC_x$, popping and propagating concepts to their immediate neighborhood (i.e. without recursion). This way the expansion may get close to $O(M \log N)$ time (provided that elements are not often pushed back into H once they are popped out of H), where $N = \mid \mathcal{O} \mid$ and $M = \mid \mathcal{R} \mid$.

With the suggested optimizations, here $M \log N$ should be closer to $M \log \mid \mathcal{C}_0 \mid$. The algorithm can thus be modified into the following:

**Figure 4.2.** Example of preference expansion with the CSA algorithm
There are a lot of optimizations in the CSA state of the art which goal is to prune the whole possible expansion tree, the most common were also adapted into the algorithm:

- Do not expand a node more than \( n_j \) jumps. This is the basic “stop condition” in CSA algorithms. The motivation is not expanding to concepts that are “meaningfully” far away from the original concept. For instance expanding the interest for cats to ‘LiveEntity’ does not add any useful semantics.

- Do not expand a node (or expand with a reduction degree of \( w_e = \frac{1}{n_e} \)) that has a fan-out greater than \( n_e \). The goal is to reduce the effect of “Hub” nodes that have many relations with other concepts. For instance, if a user is interested in a group of companies that trade on the Nasdaq stock exchange and belong to the Computer and Hardware sector, a correct inference is that the user could like other companies with

---

**Figure 4.3.** Priority queue variation of the spreading activation algorithm

![Algorithm](image-url)
these two features, but an inference could be considered incorrect if propagates the preference to the class ‘Company’ and expand to a thousand other companies barely related to do the original set.

- Once a node has been expanded up to $n_h$ hierarchical properties, do not expand the node down through hierarchical properties. The intention of this constraint is to not generalize a preference (semantically) more than once, as this is a risky assumption to make with the original user’s preferences. For instance, in the example of section 3.1, were the user likes snakes, lizards, and chameleons, the system can infer quite safely that the user has a probability to like reptiles in general, but it could seem not so straightforward to infer a preference for any kind of animal in general.

Figure 4.4 shows a final version of the algorithm with priority queue and optimization parameters:
The spreading activation algorithm is rich in parameters, and normally they have to be set according to the ontology or ontologies used for the preference expansion. Ontologies are varied on structure and definition, specialized ontologies usually have a high level of profundity, and
general ontologies usually contain a high amount of topic-concepts, with high level of fan-out for every topic. A summary of these parameters can be found in Table 4.1.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w(r,x,y), w(r)$</td>
<td>Probability that a concept $y$ is relevant for the current context if we know that a concept $x$ is in the context or in the user profile, and $r(x,y)$ holds. Also seen as the power of preference/context propagation that the relation $r \in \mathcal{R}$ has for concepts $x$ and $y$. Perhaps the most important parameter of the CSA algorithm, and also the most difficult parameter to decide. In our experiments (see section 5) these values were empirically fixed for every property in the ontology, not taking into account the involved concepts of the relation, this can be expressed as $w(r)$. Future work will be to study the power of propagation with the involved concepts, studying techniques of semantic relation between two concepts of the same ontology.</td>
</tr>
<tr>
<td>power($x$)</td>
<td>The power of preference/context propagation that a single concept $x$ has. By default equal to 1.</td>
</tr>
<tr>
<td>$\varepsilon$</td>
<td>The minimum threshold weight value that a concept has to have in order to expand its weight to related concepts. A high threshold value would improve the performance of the propagation algorithm, as less expansion actions are to be made. However, higher values of this threshold do exploit less the underlying semantics of the KB, thus resulting in poorer propagation inferences.</td>
</tr>
<tr>
<td>$n_j$</td>
<td>The maximum number of expansion steps performed by the expansion algorithm. Similar to the threshold value $\varepsilon$, this parameter has to be set as a tradeoff of performance versus quality of inference.</td>
</tr>
<tr>
<td>$w_e(x,n_e)$</td>
<td>Reduction factor $w_e$ of the extended context/preference $x$ applied to a node with a fan-out of $n_e$. In our implementation $w_e$ is defined as $w_e(x,n_e) = \frac{1}{n_e}$</td>
</tr>
<tr>
<td>$n_h$</td>
<td>Maximum number of times that a concept can be generalized.</td>
</tr>
</tbody>
</table>

**Table 4.1. Spreading activation algorithm optimization parameters**

### 4.5.2 Comparison to Classic CSA

The proposed algorithm is a variation of the Constrained Spreading activation (CSA) technique originally proposed by Salton (Salton and Buckley 1988). Here $EC$ corresponds to $I$, and $C$ corresponds to the initial values of $I$ when the propagation starts. The output function $f$ is here a
threshold function, i.e. $O_j = I_j$ if $I_j > \epsilon$, and 0 otherwise. The activation threshold $k_j$, here $\epsilon$, is the same for all nodes. In practice, we use a single property, $EC$, for both $O$ and $I$.

Instead of making $I_j = \sum_i O_i \cdot w_{i,j}$, we compute $I_j = R\left(\left\{O_i \cdot w_{i,j}\right\}_{i=1}^n\right)$, whereby $I_j$ is normalized to $[0,1]$, and corresponds to the probability that node $j$ is activated, in terms of the probability of activation of each node $i$ connected to $j$. $w_{i,j}$ is here $w(r)^{\text{power}(i)}$, where $r(i,j)$ for some $r \in \mathcal{R}$, and the power property of $i$ is a node-dependent propagation factor. Since the graph is a multigraph, because there may be more than one relation between any two nodes, $w_{i,j}$ has to be counted as many times as there are such relations between $i$ and $j$.

The termination condition, excluding the optimization parameters, is that no node with $I_j > \epsilon$ remains that has never been expanded.

Finally, the propagation to (and therefore through) a set of initial nodes (the ones that have an non-zero initial input value) is inhibited.

Regarding the spreading constraints, in the current version of our algorithm we use:

- Distance constraint.
- Fan-out constraint.
- Path constraint: links with high weights are favored, i.e. traversed first.
- Activation constraint.

### 4.5.3 Semantic Preference Expansion

In real scenarios, user profiles tend to be very scattered, especially in those applications where user profiles have to be manually defined, since users are usually not willing to spend time describing their detailed preferences to the system, even less to assign weights to them, especially if they do not have a clear understanding of the effects and results of this input. Even when an automatic preference learning algorithm is applied, usually only the main characteristics of user preferences are recognized, thus yielding profiles that may entail a lack of expressivity. In order to meet this limitation, a novel approach may be followed: the extension of preferences through ontology relations, following the same approach, and even the same algorithm, that is used to expand the runtime context.

The proposed approach for preference extension is to follow the same mechanism that was used for the extension of the semantic context, as described in section 4.5. The main difference is that here relations are assigned different weights $w'(r,x,y)$ for propagation, since the inferences one can make on user preferences, based on the semantic relations between concepts, are not necessarily the same as one would make for the contextual relevance.
For instance, if a user is interested in the Sumo sport, and the concept Japan is in the context, we assume it is appropriate to activate the specific user interest for Sumo, to which the country has a strong link: we have nationalSport (Japan, Sumo), and \( w(\text{nationalSport, Japan, Sumo}) \) is high. However, if a user is interested in Japan, we believe it is not necessarily correct to assume that her interest for Sumo is high, and therefore \( w'(\text{nationalSport, Japan, Sumo}) \) should be low.

In general, it is expected that
\[
w'(r,x,y) \leq w(r,x,y),
\]
i.e. user preferences are expected to have a shorter expansion than has the context.

Given an initial user preference \( P \), the extended preference vector \( EP \) is defined by:
\[
EP_y^\oplus = \begin{cases} P_y & \text{if } P_y > 0 \\ R\left(\{EP_x \cdot w'(r,x,y) \cdot \text{power}(x)\}_{x \in O, r \in R, r(x,y)}\right) & \text{otherwise} \end{cases}
\]

Equation 9. Expanded preference vector

Which is equivalent to equation 6 where \( EC, C \) and \( w \) have been replaced by \( EP, P \) and \( w' \), and the variable \( t \) has been removed, since long-term preferences are taken to be stable along a single session. Also, following the insights introduced in section 4.2. The final expanded preferences are computed in two expansion phases: firstly an expansion of the positive preferences \( EP^+_y \) and, secondly, an expansion of the mapping of negative preferences \( EP^-_y \), calculating the final value as the subtraction of the negative expanded preferences (dislikes) to the expanded positive preferences (interests):
\[
\]


### 4.6 Contextual Activation of Preferences

After the context is expanded, only the preferred concepts with a context value different from zero will count for personalization. This is done by computing a contextual preference vector \( CP \), as defined by
\[
CP_x = EP_x \cdot C_x \text{ for each } x \in O,
\]
where \( EP \) is the vector of extended user preferences. Now \( CP \) can be interpreted as a combined measure of the likelihood that concept \( x \) is preferred and how relevant the concept is to the current context. Note that this vector is in fact dependent on user and time, i.e. \( CP(u, t) \). This can be seen as the intersection value of expanded preferences and context (see Figure 4.5).
Note also that at this point we have achieved a contextual preference mapping $\Phi$ as defined in Section 4.3, namely $\Phi(P(u), C(u,t)) = CP(u,t)$, since $CPx(u,t) > Px(u,t)$ only when $EPx(u)$ has been derived from $P(u)$ through the constrained spreading expansion mechanism, and $CPx(u,t) < EPx(u)$.

### 4.7 Contextualization of Preferences

Finally, given a document $d \in D$, the personal relevance measure PRM of $d$ for a user $u$ in a given instant $t$ in a session is computed as $\text{prm}(d, u, t) = \cos(CP(u, t - 1), M(d))$, where $M(d) \in [0,1]$ is the semantic metadata concept-vector of the document.$^{15}$

---

$^{15}$ Note that in order to contextualize user preferences in the instant $t$, the context is taken one step earlier, i.e. at $t - 1$. This means that the last request (the one being currently responded by the system) is not included in the context. Otherwise, the last request would contribute twice to the ranking: once through $\text{sim}(d, q)$, and again via $\text{prm}(d, u, t)$. 

---
This \( \text{prm} \) function is combined with the query-dependent, user-neutral rank value, to produce the final rank value for the document:

\[
\text{score}(d, q, u, t) = f(\text{prm}(d, u, t), \text{sim}(d, q))
\]

**Equation 11.** Computation of final personalized and contextualized document score.

The similarity measure \( \text{sim}(d, q) \) stands for any ranking technique to rank documents with respect to a query or request. In general, the combination above can be used to introduce a personalized bias into any ranking technique that computes \( \text{sim}(d, q) \), such as the common query and content based algorithms: keyword search, query by example, metadata search, etc.

### 4.8 Long and Short Term Interests in the State of the Art

The approach introduced in this work makes a clear distinction between short-term (i.e. user context) and long-term (i.e. user preferences) interests. Although we only tackle the context acquisition technique, we acknowledge that profile learning techniques do differ in nature with context acquisition approaches. Our personalization approach uses two different representations for the user profile and the user context and treats them differently. While user preferences are permanent, or have long-term update modules, user context is far more dynamic and changes between and within sessions. Context here has a clear distinction with respect to preferences in its application. While preferences are meant for exploitation in the personalization phase, the purpose of context is to alter these preferences in such a way that only preferences related to context are taken into consideration in this exploitation phase. The dependence of the Contextualized Preference value \( CP(u, t) \) with the time variable indicates the drift nature of the contextualization of preferences, as contextualized concepts keep changing in “slots of contexts”. Figure 4.6 depicts this concept of contextualized preference drift.

![Figure 4.6](image-url) Characterization of concept drift in the contextualization algorithm
In general, proposals that tackle the differences between short and long-term interest also maintain two (or more) different interest representations. However, they usually do not have a clear distinction between this two representations, either by not differentiating the acquisition techniques (Ahn et al. 2007; Billsus and Pazzani 2000), or by applying them indistinctly during the exploitation phase (Ahn et al. 2007; Sugiyama et al. 2004).

Early work, such as the Alipes personalized news agent (Widyantoro et al. 1997), already tackled the differences on exploiting and acquiring user short and long-term interests. The context was obtained only from the actual session’s feedback, while the user profile was learnt from all the past historical feedback. The system takes into consideration negative feedback, but only for the context representation. A final profile is created by basic combination of the short and long-term profile, although each profile can have more or less assigned importance weight depending on how much has the system has already learnt about the user. For instance, the system can give more weight to the short-term profile when there are few learned documents in the long-term profile. Sugiyama et al. (2004) also applied a combination of the short and long-term profiles. They acquired three different profiles: a long-term profile, a short-term profile and a session-based profile. The three where obtained from the same term extraction technique, but were maintained differently: the long-term profile had a forgetting function, where preferences were discarded as the days passed, the short-term profile was obtained from the current day sessions, and was combined into the session based profile, obtained from the current session, giving a higher weight to the latter.

Other systems do not combine the profile in any form, but apply the exact same acquisition technique. Billsus and Pazzani (2000) and Ahn et al. (2007) apply a similar extraction technique to create both their long-term profile and short-term profile. They only limit the number of documents (from the browsing history) to be used as input in order to create the short-term profile: 100 and 20 document respectively. Regarding the exploitation phase, Billsus and Pazzani (2000) apply either the short or the long-term profile depending on which one is more similar to the current opened document. Ahn et al. lets the user select which profile to apply.

Shen and Tan apply the same language model approach to two different systems, a personalization system (Tan et al. 2006), and a context-aware system (Shen et al. 2005a). Both systems create a language model based on the past clickthrough history, limited to the current session for the context-aware system. Once the language models are created, both can be incorporated to the query, in order to take into consideration the interests of the user, her current context, or both.
4.9 An Example Use Case

As an illustration of the application of the contextual personalization techniques, consider the following scenario: Sue’s subscribed to an economic news content provider. She works for a mayor food company, so she has preferences for news related to companies of this sector, but she also tries to be up-to-date in the technological companies, as her company is trying to apply the latest technologies in order to optimize the food production chain. Sue is planning a trip to Tokyo and Kyoto, in Japan. Her goal is to take ideas from different production chains of several Japan partner companies. She has to document about different companies in Japan, so she accesses the content provider and begins a search session.

Let us assume that the proposed framework has learned some of Sue’s preferences over time or Sue has manually added some preference to the system, i.e. Sue’s profile includes the weighted semantic interests for domain concepts of the ontology. These include several companies from the food, beverage and tobacco sector and also several technological companies. Only the relevant concepts have been included and all the weights have been taken as 1.0 to simplify the example. This would define the $P(u_{Sue})$ vector, shown in Table 4.2.
Table 4.2. Example of user Profile: Sue's preferences

In our approach, these concepts are defined in a domain ontology containing other concepts and the relations between them, a subset of which is exemplified in Figure 4.7.

The propagation weight manually assigned to each semantic relation is shown in Table 4.3. Weights were initially set by manually analyzing and checking the effect of propagation on a list of use cases for each relation, and was tuned empirically afterwards. Investigating methods for automatically learning the weights is an open research direction for our future work.
When Sue enters a query $q_1$ (the query-based search engine can be seen essentially as a black box for our technique), the personalization system adapts the result ranking to Sue’s preferences by combining the query-based $\text{sim}(d,q)$ and the preference-based $\text{prm}(d,u_{Sue})$ scores for each document $d$ that matches the query, as described in Section 0. At this point, the adaptation is not contextualized, since Sue has just started the search session, and the runtime context is still empty (i.e. at $t = 0$, $C(u, 0) = \emptyset$).

But now suppose that the need of information expressed in $q_1$ is somehow related to the concepts Tokyo and Kyoto, as Sue wants to find information about the cities she’s visiting. She opens and saves some general information documents about the living and economic style of these two cities.

As a result, the system builds a runtime context out of the metadata of the selected documents and the executed query, including the elements shown in Table 4.4. This corresponds to the $C$ vector (which for $t = 1$ is equal to $\text{Req}(t)$), as defined in section 4.4.

Now, Sue wants to see some general information about Japanese companies. The contextualization mechanism comes into place, as follows.

1. First, the context set is expanded through semantic relations from the initial context, adding more weighted concepts, shown in bold in Table 4.5. This makes up the $EC$ vector, following the notation used in Section 4.5.
Table 4.5. Example of expanded context vector

<table>
<thead>
<tr>
<th>CONCEPT</th>
<th>WEIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location: Tokyo</td>
<td>1.0</td>
</tr>
<tr>
<td>Location: Kyoto</td>
<td>1.0</td>
</tr>
<tr>
<td>Location: Japan</td>
<td>0.88</td>
</tr>
<tr>
<td>Company: Japan Tobacco Inc.</td>
<td>0.55</td>
</tr>
<tr>
<td>Company: Yamazaki Baking Co.</td>
<td>0.55</td>
</tr>
<tr>
<td>Industry Sector: Food, Beverage &amp; Tobacco (F, B &amp; T)</td>
<td>0.68</td>
</tr>
<tr>
<td>Person: ‘Makoto Tajima’</td>
<td>0.23</td>
</tr>
</tbody>
</table>

Similarly, Sue’s preferences are extended through semantic relations from her initial ones, as show in section 4.5.3. The expanded preferences stored in the EP vector are shown in Table 4.6, where the new concepts are in bold.

Table 4.6. Example of extended user preferences

<table>
<thead>
<tr>
<th>CONCEPT</th>
<th>WEIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location: Yamazaki Baking Co.</td>
<td>1.0</td>
</tr>
<tr>
<td>Location: Japan Tobacco Inc.</td>
<td>1.0</td>
</tr>
<tr>
<td>Location: McDonald’s Corp.</td>
<td>1.0</td>
</tr>
<tr>
<td>Location: Apple Computers Inc.</td>
<td>1.0</td>
</tr>
<tr>
<td>Location: Microsoft Corp.</td>
<td>1.0</td>
</tr>
<tr>
<td>Brand: ‘Microsoft’</td>
<td>0.5</td>
</tr>
<tr>
<td>Brand: ‘Mcdonald’s’</td>
<td>0.5</td>
</tr>
<tr>
<td>Person: ‘Makoto Tajima’</td>
<td>0.5</td>
</tr>
<tr>
<td>Brand: ‘Apple’</td>
<td>0.5</td>
</tr>
<tr>
<td>Industry Sector: F, B &amp; T</td>
<td>0.70</td>
</tr>
</tbody>
</table>

The contextualized preferences are computed as described in section 4.6, by multiplying the coordinates of the EC and the EP vectors, yielding the CP vector shown in Table 4.7 (concepts with weight 0 are omitted).
Table 4.7. Example of contextualized user preferences

Comparing this to the initial preferences in Sue’s profile, we can see that Microsoft, Apple and McDonald’s are disregarded as out-of-context preferences, whereas Japan Tobacco Inc. and Yamazaki Baking Co. have been added because they are strongly semantically related both to the initial Sue’s preferences (food sector), and to the current context (Japan). Figure 4.8 depicts the whole expansion and preference contextualization process regarding the presented use case.

Figure 4.8. Visual representation of the preference contextualization.

4. Using the contextualized preferences showed Table 4.7, a different personalized ranking is computed in response to the current user query $q_2$ based on the $EC(u_{Sue}, t_1)$ vector, instead of the basic $P(u_{Sue})$ preference vector, as defined in section 4.6.
This example illustrates how our method can be used to contextualize the personalization in a query-based content search system, where queries could be of any kind: visual, keyword-based, natural language, etc. The approach could be similarly applied to other types of content access services, such as personalized browsing capabilities for multimedia repositories, automatic generation of a personalized slideshow, generation of personalized video summaries (where video frames and sequences would be treated as retrieval units), etc.
Chapter 5

Experimental work

Evaluating interactive and personalized retrieval engines is known to be a difficult and expensive task (Wilkinson and Wu 2004; Yang and Padmanabhan 2005). The classic IR evaluation model, known as Cranfield-style evaluation framework (Cleverdon et al. 1966) specifies an evaluation framework with 1) a document collection, 2) a set of topic descriptions that express an information need and 3) explicit relevance assessments made for each topic description. The only source of information about the user in the classic model is the topic description; which describes the current information need of the user. In order to add more user-specific information, such as the user interests or the current user context, the evaluation framework needed to be extended. The most common extension is adding the interaction model between the user and the retrieval system (Borlund 2003; Thomas and Hawking 2006; White 2004a; White and Kelly 2006). The evaluation is then made following the interaction model given a topic (i.e. the current user’s task). This interaction model is generally obtained from real user interaction with the users (explicit) or from query log analysis (implicit).

5.1 Evaluation of Interactive IR Systems: An Overview

We can broadly group user evaluation approaches into those that involve real users, which we shall call user centered approaches, and those that do not interact directly with users, or data driven approaches. Within these two approaches there is a broad spectrum of evaluation techniques. User centered approaches commonly involve user questionnaires (Dumais et al. 2003; Martin and Jose 2004; White et al. 2005a), side-by-side comparisons (Thomas and Hawking 2006), or explicit relevance assessments (Finkelstein et al. 2002). Data driven approaches normally involve query log analysis (Dou et al. 2007; Thomas and Hawking 2006) or test collections (Shen et al. 2005a). The following sections summarize the main characteristics of interactive retrieval evaluation. Table 5.1 is a brief classification of the studied techniques.
### 5.1.1 User Centered Evaluation

Borlund suggested an evaluation model for IIR (Interactive Information Retrieval) involving real users, or subjects recruited for the test (Borlund 2003). The author proposed a set of changes to the classic Cranfield evaluation model:

- The topics or information needs should be more focused on the human evaluators, who should develop individual and subjective information need interpretations. Borlund suggested the *simulated situation*, which is composed by a *simulated work task situation*: a short “cover story” that describes the situation (i.e. context) that leads to an individual information need, and an *indicative request*, which is a short suggestion to the testers on what to search for. A simulated situation aims to trigger a subjective, but simulated, information need and to construct a platform on which relevance in context can be judged. An example of a simulated situation can be found in Figure 5.1.

- The evaluation metrics should meet the subjective relevance of the users. Borlund suggested the Retrieval Relevance (RR) measure, which adds an additional component of subjective relevance. RR is composed of two relevance values, one regarding the relevance of the document to the topic (which can be precomputed over a prefixed
corpus) and a subjective relevance, given by the user, taking into consideration the simulated situation. Borlund states that the classic binary relevance marking, where documents are evaluated as relevant or not relevant, is not sufficient for IIR evaluation, where the contextual relevance has to be taken into consideration. Thus, the RR considers multiple degree relevance assessments. This tendency can also be observed in other works (Finkelstein et al. 2002; Järvelin and Kekäläinen 2000; Kraft et al. 2006) as well as in the interactive text retrieval evaluation community at large (Allan 2003).

Simulated situation

Simulated work task situation

After your graduation you will be looking for a job in industry. You want information to help you focus your future job seeking. You know it pays to know the market. You would like to find some information about employment patterns in industry and what kind of qualifications employers will be looking for from future employees.

Indicative request

Find, for instance, something about future employment trends in industry, i.e., areas of growth and decline.

Figure 5.1. Example of a simulated situation, from (Borlund 2003).

Some user centered evaluations approaches (Ahn et al. 2007; Dumais et al. 2003; Martin and Jose 2004; Rhodes and Maes 2000) use test questionnaires and personal interviews before or after the human tester interacts with the retrieval engine. For instance, the evaluation of the Stuff I’ve Seen application (Dumais et al. 2003) includes prior and posterior search questionnaires, filled in by the users before they start interacting with the application, and over a month afterwards. The questionnaire had both quantitative (e.g. “How many times did you use the system yesterday?”) and qualitative (e.g. “Do you find quickly web pages that you have visited before?”) questions. The responses were then compared with the pre-usage ones. Questionnaires give more direct feedback to the evaluators, but require considerable effort to extract and analyze the test results; interviews are also highly time consuming. Rhodes and Maes (2000) use post-questionnaires to evaluate their proactive system once the user has finished a more or less fixed task: writing an essay about a topic. Martin and Jose (2004) and Ahn et al. (2007) use post-questionnaires to complement their user interaction logs analysis.

Another approach for user centered evaluation is using the testers to create the interaction model with the system and then evaluate the system with a quantitative metric (see section 5.1.3). The
interaction model can be based on presented topics (i.e. simulated situations) or on a free interaction of the users with the system. In the majority of user-centered evaluations (Budzik and Hammond 2000; Kraft et al. 2006; Shen et al. 2005b; Sugiyama et al. 2004), the users have to perform a set of fixed tasks. For instance, Shen et al. (2005b) used the topics defined in the TREC interactive evaluation track\textsuperscript{16} (see section 5.1.4). For each topic, the users perform a free number of queries, in what it is known as a search session. At the end of the process the users will explicitly judge the relevancy of each search result (results can also be mixed with ones from the baseline search system). The result set from the final iteration are then evaluated and compared against a baseline result set. The baseline is normally a Web search engine, since usually the system acts as a proxy of a popular search engine. Leroy et al. (2003) use the TREC ad-hoc corpus, topics, and assessments for their quantitative evaluation. Vishnu (2005) has users interacting with desktop applications, as this is how the system gets session related information from the user. The users perform tasks that stimulate an interaction with the desktop environment, such as writing an essay. At the end of the experiment, the users try the contextualization of a Web search engine and rank the relevance of each result. Finally, the mean relevance metric is compared against the unordered results.

Thomas and Hawking (2006) presented a side-by-side evaluation schema capable of collecting implicit feedback actions and of evaluating interactive search systems. This evaluation schema consisted of a two panel search page with one query input, showing the results for the input query for both the baseline and the evaluated system. Thus, the users interacted with both search engines to be evaluated. The system was capable of collecting implicit feedback measures, i.e. to create an interaction model for interactive and context-aware system evaluations.

In order to derive quantitative measures, the users have to provide relevance assessments for the results returned by the system under evaluation. The relevance assessments can be collected by implicit or explicit feedback techniques. Explicit relevance assessments require a considerable amount of effort from the testers. Aiming to palliate this, some evaluations base the assessments on implicit feedback techniques, where the relevance of a result is inferred automatically by monitoring the user interaction (Dou et al. 2007; Thomas and Hawking 2006). The same indicator of implicit relevance used for user profiling or context acquisition (see section 2.1.3) can be used for this type of implicit evaluation. The time the user spends viewing the content, the amount of scrolling, or whether the user saves the results, are some of the relevance indicators that can be exploited. The use of implicit feedback for the generation of relevance

\textsuperscript{16} Evaluation track for interactive IR systems: http://trec.nist.gov/data/interactive.html
judgments has been proven to have a considerable accuracy. Thomas and Hawking (2006) were able to observe that implicit relevance judgments were consistent with explicit feedback evaluations 68 to 87% of the times.

### 5.1.2 Data Driven Evaluation

Data driven evaluations construct the interaction without an explicit participation of human subjects. The advantage of a data driven evaluation is that, once the dataset collection is prepared, the corpus can be easily reused for other test evaluations. A particularly difficult case for evaluators is when the Web is taken as document corpus. The Web is highly dynamic; search results for a specific topic can vary greatly over a short period of time. Interaction models strongly depend on which search results were displayed to the user. Therefore, the evaluation has to be as close as possible to the time the interaction model was created. With the aim of easing the construction of interactive evaluation corpus, White presented a model for the automatic creation of interaction models (White 2004a). The model could take into consideration parameters such as how many relevant documents are visited, or the “wandering” behavior. White suggests how these models can be further optimized with real user information.

Another solution is to exploit the query logs from the interaction of users with a publicly available search engine, which can be obtained either from the search engine providers themselves, or by means of some proxy-based log collection. The information of the query logs may range from clickthrough data, to the time spent viewing a document, or the opened applications. Dou et al. (2007) analyzed a large set of query log history in order to evaluate different personalization and contextualization approaches. This log history, obtained from the MSN search engine\(^\text{17}\), stores information such as cross-session user IDs (through the browser’s cookie information), user queries and clickthrough information. The authors are then able to implicitly state what results are relevant for the users, by globally analyzing the query and clickthrough information of a large set of users. In short, the authors compensate the uncertainty inherent to implicit relevance assessments with a large base of users. By analyzing this so called training set of clickthrough information, the authors can evaluate the effectiveness of the different approaches by computing the precision of the result set as reordered by each personalization algorithm.

\(^{17}\) [http://www.msn.com](http://www.msn.com)
5.1.3 Evaluation Metrics

There are numerous evaluation metrics in the literature for classic IR evaluation, which are largely applied by IIR authors (Budzik and Hammond 1999; Budzik and Hammond 2000; Shen et al. 2005a; Shen et al. 2005b; Sugiyama et al. 2004). These metrics rely on relevance assessments for the returned results. Whereas in classic evaluation the relevance assessments are based on the relevancy of the result to the current topic, the authors of IIR systems incorporate human subjective assessments, either implicitly by analyzing interaction logs, or explicitly, by asking the users to rate the results (relevance based metrics) or provide a “best” order (ranking position based metrics).

One of the best known relevance based metrics in the IR community are precision and recall (PR) (Baeza-Yates and Ribeiro-Neto 1999). These metrics take into consideration the ratio of relevant retrieved documents (precision) and the ratio of retrieved relevant documents retrieved (recall). These values are commonly displayed as a curve showing a vector of values at 11 fixed percentage points of recall (from 0 to 100% of relevant documents retrieved, with increments of 10%).

 Precision at _n_ (P@N), or cut off points, is another quite common metric in IIR evaluation (Budzik and Hammond 2000; Kraft et al. 2006; Shen et al. 2005b). P@N is the ratio between the number of relevant documents in the first _N_ retrieved documents and _N_. The P@N value is more focused on the quality of the top results, with a lower consideration on the quality of the recall of the system. This is one of the main reasons why this value is often used in content recommender systems, focused on presenting short and precise lists of recommendations. Another de facto standard metric is the mean average precision (MAP). Mean average precision is the average for the 11 fixed precision values of the PR metric, and is normally used for a simple and convenient system’s performance comparison. Shen et al. (2005a) use this evaluation metric for their system. Other authors use non-standard (or uncommon) metrics, such as the number of relevant documents in the first 10 results (Finkelstein et al. 2002), or the average of the explicit assessment of the user, given as values between 1 and 5 (Vishnu 2005).

In ranking position metrics, users inspect the returned result set and indicate what could have been the best subjective order. Based on this, a measure of distance between ranked lists (e.g. K-distance) is used in order to compute the performance of the retrieval system in terms of how close or far the system results are from the users’ list.

Some authors have focused on new metrics for IIR evaluation. Borlund adapted classic metrics to context-aware and multi-degree relevance assessments (Borlund 2003). Two of the metrics
presented were the Relative Relevance (RR) and the Half Life Relevance (HLR), which add multi-valued relevance assessments and is based on vectors of two relevance values, one related to the algorithm itself (classic ranking) and one related to the subjective relevance given by the user. Järvelin & Kekäläinen (2000) proposed two new evaluation metrics, CG (Cumulative Gain) and DGN (Discount Cumulative Gain). These take into consideration the position of the relevant document, giving more importance to highly relevant documents (i.e. relevant to topic and user context) that appear in the top positions, taking into account that those are the most likely to be noticed by the user. This metric was recently adapted to an interactive retrieval setting (Järvelin et al. 2008).

In order to validate their implicit evaluation methods, Dou et al. (2007) used rank scoring and average rank metrics, which have their origin in recommender system evaluations. The metrics assess the quality of a result list presented by a recommendation system. Both metrics are computed by using the clickthrough data and the position of the documents in the result set.

5.1.4 Evaluation Corpus

An corpus for experimental evaluation is considerably hard to prepare. If this is true in IR at large the difficulties in IIR systems is considerably higher, since the interaction model has to be incorporated, either in a fixed way, along with the corpus (Shen et al. 2005a), or as topic description “guiding” the user to an interaction with the system (Borlund 2003). Most authors prefer to either have a free corpus (the Web, using search proxies or local repositories as Desktop files), or use or adapt already existing corpora from the IR community (Shen et al. 2005a).

The TExT Retrieval Conference (TREC)\(^{18}\) is an annual international conference with the purpose of supporting large scale evaluation within the IR community. Every year, the conference includes a set of tracks, each focusing on a specific area of IR. The normal procedure for participating in a TREC conference is selecting a track, obtaining the corpus assigned to the track, evaluating the list of topics (i.e. search queries) on a retrieval engine developed by the participant, and submitting the results to the task committee, where the results will be evaluated. The relevance judgments are obtained with a polling method: the first 100 result for every topic of every submitted search engine is evaluated, instead of evaluating the whole collection, which given the size of most of the datasets (some going over 20M documents) is simply not realistically feasible.

\(^{18}\) http://trec.nist.gov/
The TREC Interactive Track investigates the user interaction with retrieval engines. The topic execution is made by human users, within which different interactions with the system take place, such as query reformulation or explicit relevance feedback (depending on the protocol of each year). The interactions, however, are not stored or added to the evaluation framework. Thus, the evaluation data are not as useful for reuse, since they lack the interaction model, and the experiments cannot be reproduced by other researchers. There is also an interactive task for video retrieval, TRECVID, originally a track of TREC and now an independent workshop. In TRECVID’s interactive task, the experimenter can refine queries and modify the ranked result list after the initial query. The interactive search usually performs significantly better than the automatic or manual query tasks.

TREC’s HARD Track (Allan 2003) investigates the knowledge of the user’s context in information retrieval systems. The topics are expressed similarly to other TREC tracks, except that additional metadata on the “context” of the user is given. This metadata includes the desired genre of the returned results, the purpose of the search or a geographic focus specification for documents. The relevance judgments take into consideration the context described in the metadata, judging a document as non relevant (to the topic), soft relevant (i.e. on topic but not relevant to the context), and hard relevant (relevant to both topic and context).

Shen et al. (2005a) adapted a TREC dataset, taking advantage of the fact that there already were relevant assessments and topic definitions for the document database. As this corpus lacked of an interactive model, the authors constructed one by monitoring three recruited human subjects as they searched for the evaluation topics.

5.2 Experimental Setup

The contextualization techniques described in the previous chapters have been implemented in an experimental prototype, and tested on a medium-scale corpus. Similarly to other IIR approaches discussed in the previous sections, a formal evaluation of the contextualization techniques may require a significant amount of extra feedback from users in order to measure how much better a retrieval system can perform with the proposed techniques than without them. For this purpose, it would be necessary to compare the performance of retrieval a) without personalization, b) with simple personalization, and c) with contextual personalization. This requires building a testbed consisting of: a document corpus, a set of task descriptions, a set of user profiles, the relevance judgments for each task description, and the interaction model, either fixed or provided by the users.
At the time of writing, there is no dataset that allowed the evaluation of our system. Ours is an adaptive, highly interactive, dynamic, context-ware retrieval system, and based on a semantic index. Hence, depending on these multiple variables, we had to build our own evaluation corpus, topics and task descriptions. For this, we reused an external KB (Popov et al. 2004), we recruited a number of users and developed the methodology that is following described. Anyhow, this methodology and the dataset can be totally or partially exported for the evaluation of comparable systems.

The document corpus that we set up for our experiments consists of 145,316 documents (445MB) from the CNN web site\(^{19}\), plus the KIM domain ontology and KB publicly available as part of the KIM Platform, developed by Ontotext Lab\(^{20}\), with minor extensions. The KB contains a total of 35,689 concepts (belonging to 281 RDF classes), and 465,848 relations (from 138 properties). The CNN documents are annotated with KB concepts, amounting to over three million annotation links. The relation weights $R$ and $R^{-1}$ were first set manually for each property on an intuitive basis, and tuned empirically afterwards by running a few trials.

The retrieval system used for this experiment is a semantic search engine co-developed by the author (Castells et al. 2007), which did not implement itself any personalization capability, but was developed in order to exploit a semantic index, resulting in a better performance than a classic keyword-based search engine. The retrieval system was adapted in a way that included: 1) a user profile editor, 2) personalization capabilities, 3) the context monitor and acquisition engine, 4) the semantic expansion module and 5) our contextual personalization approach.

Task descriptions are similar to the simulated situation explained in section 5.1.1. The goal of the task description is to provide sufficient query and contextual information to the user to perform the relevance assessments and to stimulate the interaction with the evaluation system.

Similar to the TREC Hard track (see 5.1.4) we count on real users to provide explicit relevance assessments with respect to a) query relevance, b) query relevance and general user preference (i.e. regardless of the task at hand), and c) query relevance and specific user preference (constrained to the context of her task).

The selected metrics in our experiments are PR and P@N values for single query evaluation and MAP, and both average PR and P@N values for the whole system performance evaluation. Average PR and P@N are the average values of the PR and P@N points for a set of queries. PR

---

\(^{19}\) http://dmoz.org/News/Online_Archives/CNN.com

\(^{20}\) http://www.ontotext.com/kim
was chosen because it allows a finer analysis of the results, as the values can be represented in a graph and different levels of performance can be compared at a glance. For instance, a retrieval engine may have a good precision, showing relevant results in the top five documents, whereas the same search system may lack a good recall performance, being unable to find a good proportion of relevant documents in the search corpus. See Figure 5.2 for a visual illustration of the precision and recall analysis areas. The target area to reach is the upper right part, as it would be indicative of the search engine having a good precision over all the results and achieving a good recall level.

Since the contextualization techniques are applied in the course of a session, one way to evaluate them is to define a sequence of steps where the techniques are put to work. This is the approach followed in the first set of experiments, for which the task descriptions consist of a fixed set of hypothetic context situations, detailed step by step. The second set of experiments, which are user centered, are focused on the overall performance of the search engine and the evaluation of real interactions with users. In this case, the interaction model is provided by the users, following the task descriptions. On the one hand, the first set of experiments give detailed and fine grained results, enabling an exhaustive analysis of the proposed approach. These tests were also used for parameter tuning: Parameters such as the relation weights $\mathcal{R}$ and $\mathcal{R}^{-1}$ and the CSA optimization parameters (see Table 4.1) were tuned using a small training set with three queries.

The second set of experiments allows for a more general and objective evaluation, though less detailed.

![Figure 5.2. Different areas of performance for a precision and recall curve.](image-url)
5.3 Scenario Based Testing: a Data Driven Evaluation

The scenario driven experiments are based, as explained earlier, on simulating search sessions by users with assumed preferences. Although subjective, these experiments aim to enable meaningful observations, and testing the feasibility, soundness, and technical validity of the defined models and algorithms. Furthermore, this approach allows a better and fine grained comprehension of the proposed system behavior, which is key due to the amount of different parameters and factors that affect the personalization and contextualization models.

5.3.1 Evaluation Tool

In this evaluation system, the user is able to input a keyword search and what we call semantic or ontology-based search, using the KB relations (Castells et al. 2007). The context monitor engine is based on simple implicit feedback techniques, extracting concepts from input queries (either keyword or semantic) and opened documents. The extracted concepts are applied within a timeline of actions, as explained in section 4.4, where a user input query is the unit of time. Finally, the semantic expansion and user profile contextualization algorithms are applied, using the contextualized preferences as input to the personalization algorithm (see section 4.6).

![Figure 5.3. Main window of the scenario based evaluation tool.](image-url)

To facilitate our own work with the execution of scenarios and result inspection, the experiments were conducted in a special-purpose evaluation tool, which is described next.
Figure 5.3 shows the main window of the evaluation environment’s User Interface (UI). The main input components of the UI are the following:

- **Keyword query input**

The user can use simple keyword queries in order to search the document corpus. Keyword search is done using the java text search engine Lucene, set up with the default parameters, using a common English list of stop words, with no stemming functionality.

In order to use our context acquisition technique, when a keyword-based query is entered, the keywords are matched to ontological concepts by using a simple mapping between concept label and keyword which, for our experiments, is available as part of the KIM KB release. This matching was slightly “fuzzified” by allowing a Hamming distance of up to two.

- **Semantic query input**

The users are able to input an ontology-based query, using KB relations (Castells et al. 2007). The users can directly input a formal ontological query, such as SPARQL\(^{21}\) (with syntax similar to SQL, a typical database query language). To facilitate the semantic query construction, users can use a simple dialog that allows the interactive edition of SPARQL queries, although not the whole query spectrum can be created with this dialog. The dialog allows the user to select which type of concept she is looking for, to add restrictions to the properties of the searched concepts, and to add restrictions on relations of these concepts with others from the KB. All the semantic topics used in the evaluated scenarios could be expressed by semantic queries with this editor. Optionally, the user may launch both a keyword and a semantic query, combining the results in a single list.

Figure 5.5 displays a snapshot of the query dialog UI for the generation of semantic queries. The figure shows the dialog that would generate the semantic query “Companies located in Central Europe that trade on the New York Stock Exchange”. Figure 5.4 is a snapshot of the dialog for the creation of complex relations, or restrictions. For instance, the property “located in Central Europe” is a complex relation since it has to be expressed as “located in a country which is a region of Central Europe”. Figure 5.4 shows the state of the dialog when this restriction is input.

\(^{21}\) http://www.w3.org/TR/rdf-sparql-query/
• **Personalization slider**

Users can adjust the level of personalization (or contextualized personalization) they wish to have for the current result set. The personalization effect is dynamically adjusted, showing a live reordering of results as the users slides the degree of personalization.

• **Profile editor**

The profile editor allows the manual selection of single concepts as user preferences. By exploiting the semantic description of the KB, users are also able to create *complex preferences*, i.e. preferences based on restrictions, like “I like airline companies” or “I don’t like tech companies that trade on the New York Stock Exchange”. These restrictions are created by means of semantic queries, using the dialogs meant for the semantic query construction (see Figures 5.4 and 5.5). This creation of complex preferences avoids having to search and
manually select those concepts as preferences. Figure 5.6 shows the main component of the profile editor UI.

![Concept profile editor](image)

**Figure 5.6.** Concept profile editor.

The main output UI components are:

- **Results summaries**

Query results are shown as a list of document snippets, showing the text most relevant to the query for each document. A gradient color bar at the near right of each snippet indicates the personalization or personalization in context score of each document. Concepts that have matched user preferences are highlighted using the following highlight color code: a) bold green when concepts match positive user interests, b) bold red when concepts match a negative interest, and c) a lighter color when concepts result from the semantic expansion algorithm: light green or light red depending on whether concepts have a final inferred positive or negative value, respectively. Figure 5.7 shows several examples of highlighted document snippets.

- **Document viewer**

Whenever the user clicks on a document snippet, the results appear in the large text panel of the main UI (see Figure 5.3). The document text has been cleaned of HTML formatting tags and link URLs. The text is also highlighted with the same color scheme as the document snippets.
Additionally, users can inspect all the annotated concepts, and their relation weights, that the document contains.

**Figure 5.7.** Examples of document snippets and text highlighting.

- **User profile, context and semantic expansion information panel**

In this information panel users can inspect the whole semantic expansion and preference contextualization processes. Users may check their original preferences, the context concepts and the results of the semantic expansion approach applied to preferences and context, and the final resulting contextualized preferences. Figure 5.8 shows an example of this information panel.

**Figure 5.8.** Contextual and personalization information window.
5.3.2 Methodology

Our experimental methodology can be considered to be an extension of Borlund’s simulated situation evaluation (see section 5.1.1) with further contextual and user preference indications. The user interest profiles were fixed for the whole experiment. They consisted of a set of weighted concepts and complex preferences, selected and manually assigned by the evaluators (role that we played ourselves) in order to have a heterogeneous set of preferences in which a subset is related to the evaluated tasks. Some of these preferences were chosen because of the relation with the task descriptions. The task descriptions consisted of a) a short topic, indicating when a document is relevant to the last input query, b) a contextual situation description, indicating when a document adjusts to the actual context of the user, c) a description pointing when a document is relevant to the user preferences, and d) a step by step interaction model. The latter means considering sequences of user actions defined \textit{a priori}, which makes it more difficult to get a realistic user assessment, since in principle the user would need to consider a large set of artificial, complex and demanding assumptions. Figure 5.9 is a task description example specifically for the use case described in section 4.9.

The rationale of this task description is a user who has a set of general preferences c). The user interacts with the retrieval system, following the interaction model indicated by d) and, at some point, the user expresses her next information need by means of topic a). This interaction model, including queries and visited documents, creates a contextual situation in which documents returned by topic a) are relevant according to the indications of b). The criteria for judging the relevance of each item in the final result set produced by topic a) is that a document is relevant if it is relevant to both the user preferences c) and the current user’s context b), taking into consideration the previous interaction model d).
Chapter 5 — Experimental Work

Figure 5.9. Example of user centered task description.

Task T: Japanese Companies: Food sector companies

a) Topic
Japan based companies

b) Relevancy to context
Relevant documents are those that mention a company that has or has had an operation based in Japan. The document has to mention the company. It is not mandatory that the article mentions the fact that the company is based in Japan.

c) Relevancy to preferences
Consider that the article adjusts to your preferences when one of the mentioned companies has a positive value in your user profile.

d) Interaction model
1. Query input[keyword]: Tokio Kyoto Japan
2. Opened document: n=1, docId=345789
3. Opened document: n=3, docId=145623

The final search result of the last query interaction, the topic query, is presented to the evaluator (role that we played ourselves) to provide the explicit relevance assessments of the whole result set. The document corpus, the task descriptions, the fixed interaction model and the relevance assessments finally form a full collection test for the proposal. This collection was stored in a way that facilitates the reproduction of the evaluation and the reuse of the whole experimental dataset. Appendix A provides further details on the final set of task descriptions.

5.3.3 Results

A final set of ten task descriptions were created as part of the evaluation setup. Figure 5.10 a) shows the PR curve for the use case scenario described in section 4.9. This is a clear example where personalization alone would not give better results, or would even perform worse than non-adaptive retrieval (see the drop of precision for recall between 0.1 and 0.4 in Figure 5.10 a), because irrelevant long-term preferences (such as, in the example, technological companies which are not related to the current user focus on Japan-based companies) would get in the way of the user. The experiment shows how our contextualization approach can avoid this effect and
significantly enhance personalization by removing such out-of-context user interests and leaving the ones that are indeed relevant to the ongoing course of action.

![Graphs showing comparative performance of personalized search with and without contextualization.](image)

**Figure 5.10.** Comparative performance of personalized search with and without contextualization.

It can also be observed that the contextualization technique consistently results in better performance with respect to simple personalization, as can be seen in the average precision and recall depicted by Figure 5.10 b), which shows the average PR results over the ten use cases. Figure 5.11 depicts the MAP histogram comparing the contextualized vs. non-contextualized personalization for the ten use cases. The *Context* bars compare personalized retrieval in context vs. retrieval without personalization (i.e. the baseline system), and the *Personalization* bars compare personalized retrieval in context vs. the baseline. We can observe that the contextualization approach is consistently outperforming the personalization approach. Specifically, in use case 9 the personalization approach performs even worse than the baseline approach. This is because there are a set of preferences that boost results not relevant to the current task at hand. The contextualization approach is able to filter out some of these irrelevant preferences, resulting in an improvement of the performance over both the personalization and the baseline approach. Appendix A provides more details on the individual performance of each evaluated scenario.
5.4 User Centered Evaluation

In this second approach, real human subjects are given three different retrieval task descriptions, similar to the simulated situations described by Borlund (Borlund 2003), each expressing a specific information need, so that users are given the goal of finding as many relevant documents to the task at hand as possible.

5.4.1 Evaluation Tool

For the evaluation of this system we used the same corpus and semantic search engine used in the scenario based evaluation (see section 5.3.1), but a different tool, which is shown in Figure 5.12.
Compared to the previous evaluation tool (see Figure 5.3), we simplified the retrieval UI in order to facilitate the interaction with real users. We used the same output document snippets and document text highlighting though. The main simplifications involve the user’s input:

- **Query input**

  We limited the query input to only keyword-based queries. Here the textual search engine was again Lucene, with the same parameters as the scenario-based methodology. In order to feed the context acquisition technique, the keyword to ontology concept mapping was also used.

- **Profile editor**

  The profile editor is simplified in order to show a set of predefined concepts. Users are only allowed to inspect and rate this predefined set of concepts. Preference values range from -5 to 5; -5 indicating that the user completely dislikes the concept, 5 indicating that the user completely likes it, and 0 indicating indifference for, or ignorance of the concept. These ratings are internally normalized to [-1, 1] to fit in the model described in the previous chapters. Figure 5.13 shows a subset of available concepts for users to edit their profiles during the evaluation.
5.4.2 Methodology

Similarly to the scenario based evaluation (see section 5.3.2), we extend Borlund’s simulated situation task descriptions. Each task description expresses a specific information need, so that users are given the goal of finding as many documents as possible relevant to the task description. Differently from the scenario based task description, here the sequence of actions is not fixed but is defined with full freedom by users as they seek to achieve the proposed tasks. Therefore, there is no need to include an interaction model in the task description. The task description is composed by 1) a paragraph indicating when a document is relevant to the retrieval task, 2) an indication on how to consider a document relevant to the user preferences and 3) an example of relevant document to the task.

A total of 18 subjects were selected for the experiment, all of them being PhD students from the author’s institution. Three tasks were set up for the experiment, which can be briefly summarized as:

![User Preference Edition](image)

Figure 5.13. User preference edition.
1. News about agreements between companies.
3. Information about cities hosting a motor sports event.

---

**Task 1: Agreements between companies**

1) **Relevancy to task**

Relevant documents are those that state an agreement between two companies, the article must name the two companies explicitly. For instance, articles about a collaboration or an investment agreement between two companies are considered relevant. Agreements where one company buys totally or partially another company are NOT considered relevant.

2) **Relevancy to preferences**

Consider that the article adjusts to your preferences when one of the mentioned companies has a positive value in your user profile.

3) **Example of relevant document to the task (excerpt)**

```
CNN.com - Microsoft, AOL settle lawsuit - May. 30, 2003

Microsoft, AOL settle lawsuit

The two companies also said they have reached a wide-ranging cooperative agreement, under which they will explore ways to allow people using their competing instant message systems to communicate with each other.

Microsoft has also agreed to provide America Online software to some computer
```

**Figure 5.14.** Task description for task 1: News about agreements between companies.

Figure 5.14 shows a complete description of the first task as an example. The reader is referred to appendix B for a complete description of all three tasks. Each task was tested a) with contextual personalisation, b) with simple personalisation, and c) without personalisation. The task descriptions were assigned using a Latin square distribution, in order for users not to repeat the same task twice or more. Each of the three modes were used with six users (3 modes × 6 users = 18 tests for each task), in such a way that each user tried each of the three modes a, b, and c, exactly once. This way, each mode is tried exactly 18 times: once for each user, and 6 times for each task, in such a way that neither mode is harmed or favoured by different task difficulty or user skills.
Users never knew which of the three modes was activated when performing a task. Modes were labelled anonymously (‘A’, ‘B’ and ‘C’ modes assigned to personalized, contextualized and baseline systems, respectively), and activated in different order (see task controllers on Figure 5.12).

User preferences were obtained manually from the users by asking them to explicitly rate a predefined list of domain concepts at the beginning of the session, using the simplified version of the profile editor, as illustrated in Figure 5.13.

The relevant documents for each task, i.e. the relevancy for the topic, were marked beforehand by an expert (a role that we played ourselves), so that users are relieved from providing extensive relevance judgements. The relevance judgment corpus was created by a polling technique, where the first 100 documents of a number of queries where evaluated by ourselves and marked as relevant or not to the task description. However, users are encouraged to check the document snippets and to open the documents that seem more relevant according to their subjective interests, in order to provide the system with more contextual tips, and to provide the users with more task information. Context information is gathered based on concepts annotating such selected results, and the concepts that are related to the keywords in user queries (using the keyword-concept mapping).

A typical task execution was as follows:

1. The user reads the task description.
2. The user executes a keyword query.
3. The percentage of relevant documents found is shown to the user.
4. The user reviews the top result set summaries and inspects those which seem to apply to the task and her preferences.
5. If the user has entered at least three queries and thinks to have reached a fair percentage of relevant documents for the task, she can push the stop button to finish the task. If not, she returns to step 2.

At the end of every task, the system asks the user to mark all the found relevant documents as related or unrelated to her particular interests and the search context the user followed. The users could choose between 4 points of relevancy: no relevant, somehow relevant, relevant and highly relevant. As other studies point out (Allan 2003; Borlund 2003; Järvelin and Kekäläinen 2000), there is a need for multi-graded relevance judgements in interactive and adaptive system evaluations, where there are often multiple criteria to decide the relevancy of a document. In our experiments a document can be relevant to the task (which is decided beforehand by ourselves), relevant to the preferences of the user (chosen by the users) and relevant to the context
Users were thus encouraged to give the highest relevant relevance assessment to those documents which were at the same time relevant to their preferences, to the topic description, and to the different interactions they performed during the retrieval session. Figure 5.15 shows the UI for inputting relevance assessments by users.

![Figure 5.15. Relevance assessment UI.](image)

The relevance assessments, together with the interaction logs are stored in order to compute the performance metrics, or to automatically recreate the interaction model of the user, thus allowing some level of reproduction and detailed analysis of the experiments. For the computation of metrics, two simplifications were made for each interactive sequence (i.e. for each task and user):

- The search space is simplified to be the set of all documents that have been returned by the system at some point in the iterative retrieval process for the task conducted by this user.
- The set of relevant documents is taken to be the intersection of the documents in the search space marked as relevant for the task by the expert judgement, and the ones marked by the user according to her particular interests.
5.4.3 Results

The user centered experiment aims to provide a general view of the performance of each approach when involving real users. On this basis, we selected the average PR and average P@N metrics in order to compare the tested techniques. For computing the metrics, we consider relevant those documents that were marked by the users as relevant or highly relevant. The average values were obtained by computing the PR and P@N values for every query entered by the user. For instance, if during the task execution the user input five different queries into the system, each of the five results sets are collected, and PR and P@N values are calculated based on the final relevance assessments given by the user at the end of the task execution. This PR and P@N points are then averaged over all users and all tasks, grouped by each of the three approaches to be compared.

Figure 5.16 shows the results obtained with this setup and methodology. The average precision and recall curve on the left of this figure shows a clear improvement at high precision levels by the contextualisation technique both with respect to simple personalisation and no personalisation. The graphics show a) the precision vs. recall curve, and b) the P@N cut off points. The P@N curve clearly shows a significant performance improvement by the contextual personalisation, especially in the top 10 results. Personalisation alone achieves considerably lower precision on the top documents, showing that the contextualisation technique avoids further false positives which may still occur when user preferences are considered out of context. This validates our hypothesis that the contextualization approach is able to improve the precision of a personalization system. The improvement of the contextualization approach decreases at higher recall levels, corresponding to those preferences that were related to the task, but the contextual algorithm was unable to match with the implicit task description, either by lack of implicit information or by lack of the necessary KB relations to expand the context to the concept preferences.
Figure 5.16. Comparative performance of personalized search with and without contextualization.

Table 5.2 shows the mean average precision values for contextual, simple, and no personalisation in this experiment, which reflects that our technique globally performs clearly above the two baselines.

<table>
<thead>
<tr>
<th>RETRIEVAL MODEL</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contextual personalization</td>
<td>0.1353</td>
</tr>
<tr>
<td>Simple personalization</td>
<td>0.1061</td>
</tr>
<tr>
<td>Personalization off</td>
<td>0.0463</td>
</tr>
</tbody>
</table>

Table 5.2. Results on Mean Average Precision for each of the three evaluated retrieval models.

Most cases where our technique performed worse were due to a lack of information in the KB, as a result of which the system did not find that certain user preferences were indeed related to the context. This resulted on a decrease of the recall performance. Allegedly, solving this lack of information of the KB, or improving the semantic intersection of user interest and user context, would result on a comparable recall performance to the personalization system (note that improving recall is not possible, as the contextualization technique does not add further information, just filters preferences) and even a higher precision of the contextualized system, over the personalization approach.
Chapter 6

Conclusions

This thesis develops a novel technique for personalized retrieval, where short-term context is taken into consideration, not only as another source of preference, but as a complement for long-standing user profiles, in order to aid in the selection of the context-relevant preferences that can produce more reliable and “in context” results.

6.1 Summary and Achieved Contributions

6.1.1 Personalization Framework Based on Semantic Knowledge

The thesis proposes a personalization framework which exploits an ontology based representation of user interests. The undertaken research focuses its contributions on two of the three main areas of a personalization framework, namely user profile representation and exploitation. User profile learning alone constitutes a wide and complex area of investigation (Ardissono and Goy 2000; Gauch et al. 2007; Wallace and Stamou 2002), and is not addressed per se in the scope of this work. The available achievements in that area are thus complementary and can be combined with the techniques proposed herein. See e.g. (Cantador et al. 2008) for an extension of the research presented here where this has been in fact carried out.

The models and methods proposed in the thesis build upon a user profile representation based on ontological concepts, which are richer and more precise than classic keyword or taxonomy based approaches. Our personalization model has the main advantage of exploiting any type of relations between concepts, beyond just terms or topic based user profile representations, as are used in typical classification-based personalization systems.

User profile exploitation relies on a semantic index, based on which a semantic user-content similarity score is computed, representing the degree of similarity between a semantic user profile and the semantic (meta-)data of a content object. This allows the application of our techniques to any multimedia corpora containing annotations linking raw content to the ontology-based conceptual space where user preferences and semantic context are modeled. The main benefits of our approach in this area, with respect to the previous work in the state of the art, include:
• **An intensive exploitation of concept-based user profiles:** a formal ontology grounding enables performance and capability enhancements such as a reduction of ambiguity when matching user preferences and relevant content, or the automatic extension and/or refinement of knowledge about the user by reasoning on her preferences.

• **A semantic method for user profile exploitation:** based on content stored in a semantic index and on concept space vector representation of user interests and content.

### 6.1.2 Personalization in Context

Context is an increasingly common notion in IR. This is not surprising since it has been long acknowledged that the whole notion of relevance, at the core of IR, is strongly dependent on context — in fact, it can hardly make sense out of it. Several authors in the IR field have explored similar approaches to ours in the sense that they find indirect evidence of searcher interests by extracting implicit information from objects manipulated by users in their retrieval tasks (Shen *et al.* 2005a; Sugiyama *et al.* 2004; White *et al.* 2005a).

A first distinctive aspect in our approach is the use of semantic concepts, rather than plain terms, for the representation of these contextual meanings, and the exploitation of explicit ontology-based information attached to the concepts, available in a knowledge base. This extra, formal information allows the determination of concepts that can be properly attributed to the context, in a more accurate and reliable way (by analyzing explicit semantic relations) than the statistical techniques used in previous proposals, which e.g. estimate term similarities by their statistic co-occurrence in a content corpus. The latter can still be used complementarily whenever the coverage of the knowledge base falls short.

We have here proposed an approach for the automatic acquisition and exploitation of a live user context, by means of the implicit supervision of user actions in a search session. Our approach exploits implicit feedback information obtained from the user’s interaction with the retrieval system, and constructs a semantic representation of the user’s context without any additional interaction with the user. The proposed method is based on annotated content, so it can be applied to any type of multimedia content system that describes documents by a set of concepts.

Based on the context model, a novel method for the combination of long-term and short-term preferences is developed. The proposal is based on the semantic combination between user preferences and the semantic runtime context. The semantic combination is performed by applying a form of Constraint Spreading Activation (CSA) algorithm to the semantic relations of the KB. We have shown how concept relations can be exploited in order to find meaningful
(though indirect) semantic connections between user context and user preferences. Experimental results are positive and show the potential of the technique in promoting further progress in the area of personalized retrieval systems, by placing user interests in context. The novelty of our approach in this area includes the following points:

- **Formal semantic representation of the user context:** enabling 1) a rich representation of context, 2) semantic inference over the user context, and 3) an enhanced machine-understanding of the user context through the exploitation of a domain KB.

- **Dynamic semantic context acquisition:** To the best of our knowledge, this is the first proposal of dynamic semantic context acquisition and construction, based on implicit feedback techniques. Our proposal also introduces a novel adaptation of an implicit ostensive model that exploits content semantically annotated.

- **Novel semantic expansion strategies** based on an adaptation of the Constraint Spreading Activation (CSA) approach applied to a semantic KB.

- **A notion of personalization in context,** consisting of a user preference filtering step, in which preferences not related to the current live user context are discarded. This makes up a novel modeling approach for contextual personalization which clearly discriminates personalization and contextualization techniques, differentiating the acquisition and exploitation of user preferences and context. The benefit is twofold: the personalization techniques gain accuracy and reliability by reducing the risk of having locally irrelevant user preferences getting in the way of a specific and focused user retrieval activity. Inversely, the pieces of meaning extracted from the context are filtered, directed, enriched, and made more coherent and senseful by relating them to user preferences.

### 6.1.3 User and Context Awareness Evaluation

Our proposal has been implemented upon a semantic search engine co-developed by the author (Castells et al. 2007), and tested on a document corpus of over 150K documents, with a Knowledge Base including over 35K concepts and 450K relations.

The evaluation of context-aware and user adaptive systems is a difficult task (Yang and Padmanabhan 2005). In this thesis, we have adopted an evaluation approach that applies two complementary methodologies. Firstly, we conducted a scenario based evaluation, based on simulated situations. This allowed us to have a better comprehension of our approach behavior, together with finer grained performance analysis. Secondly, we performed an evaluation with real human subjects. The scope of this second evaluation was to test the feasibility of our system
with real users interacting with the retrieval system. The results of both evaluations were encouraging, we believe that both evaluation methodologies gave very relevant results on the specific goals which were designed and that can provide a ground methodology in which similar systems can be evaluated on the impact of context to personalization.

This evaluation methodology can be applied to analyze the performance of both personalized and context-aware retrieval systems, but with different purposes. On the one hand, personalized systems can be evaluated with our methodology in order to analyze the behavior of the personalization approach when different situations (contexts) are presented to the user, i.e. our evaluation approach can test how “precise” is the personalization approach. On the other hand, our evaluation approach can test if a context-aware system does not lose the overall perspective of the user’s trends, i.e. how the context-aware systems adjust to the long-term interests of the user. In the case of systems that aim to cover both personalization and contextualization (Ahn et al. 2007; Billsus and Pazzani 2000; Sugiyama et al. 2004; Widyantoro et al. 1997) our methodology can provide fairly comprehensive and systematic evaluation of the combined application of short and long-term user interests to a retrieval system. The proposed methodology introduces the following novelties to the personalization and context-aware research area:

- **An evaluation methodology that analyzes the impact of contextualization over personalization:** To the best of our knowledge, this is the first time an evaluation and analysis of the combination of a personalization and contextualization approach has been carried out.

- **Novel methodologies for adaptive and interactive IR evaluation:** We have introduced two complementary methodologies that extend the simulated situations defined by Borlund (2003) in order to include a set of user preferences (either simulated or assigned by a user) and a hypothetical contextual situation. In the scenario based evaluation we have also extended this contextual situation with a simulated user interaction model of the user, in order to provide this information to implicit feedback based techniques, which are widely adopted within context-aware retrieval systems.

### 6.2 Discussion and Future Work

The directions to continue the research presented in this thesis, beyond the results achieved so far, are manifold. In the following sections we suggest and discuss some areas where further improvements are possible, as well as problems of interest for future research.
6.2.1 Context modeling

Context, as presented in this thesis, is seen as a set of semantic concepts and topics related to the user’s session and current interest focus. In order to get these related concepts, the system first has to monitor the interactions of the user with the system. However, when the user is just starting a search session, the system does not yet have any contextual clue whatsoever. This problem is known as cold start. Our current solution is to not filter the user’s preferences in the first iteration with the system, but a) the opportunity to get a potential performance enhancement of the system also in the first interaction is missed (as seen in the experiments), and b) context sources need not be restricted to user interactions with the system. Interactions of the user with other applications (Dumais et al. 2003), or other sources of implicit contextual information, like the physical location of the user (Melucci 2005), could be exploited as well to address this issue.

Another limitation of the contextualization approach is that it makes the simplification that consecutive user queries tend to be related, which does not hold when sudden changes of user focus occurs. The contextualization system still works reasonably as is in these cases (in particular, the potential distortion of such changes are quickly left behind by the time decay factor), but it might be improved by automatically detecting a user changes of focus (Huang et al. 2004; Sriram et al. 2004), and determining which concepts (if not all) of the current context representation are not more relevant for preference filtering after that point.

This work presents an adaptation of the ostensive implicit feedback model (Campbell and van Rijsbergen 1996) for the semantic runtime context construction. Other implicit feedback models have been proposed that could be also explored as an alternative for our approach, e.g. White et al (2005b) made an evaluation study on different implicit feedback techniques, including techniques based on the wpq term ranking method (Robertson 1990), on which the ostensive model is based. Approaches such as the ones surveyed in that study can be potentially adapted to our framework. According to the evaluation in White’s survey, other approaches seem more effective than the ostensive model in extracting the user’s context, though we suspect that such technique’s performance depend to a large extent on the type of interactive system the user is using (e.g. video vs. image vs. text retrieval engine). One example of model mentioned in this survey, the Jeffrey’s conditioning model, gives a higher weight to concepts appearing in the first search interactions, based on the assumption that the user is more certain of their task at the first stages of the retrieval session, and that further interactions could be also motivated by curiosity, rather than by task relevancy. Investigating the adaptation of different implicit approaches, such
as the Jeffrey’s model, which has a completely different context acquisition strategy, is an interesting direction for future work.

Although user profile learning is not the focus of this thesis, an adaptation of our context modeling approach to a profile learning technique can lead to interesting results. The transition from short-term context acquisition to a persistent, longer-term user profile can be found in related work, such as (Katifori et al. 2008; Widmer and Kubat 1996), where the profile construction approach is based on learning and forgetting functions over short-term profiles. Katifori et al. (2008) define three learning levels: short interests, mezzanine interests and long-term interests, associating each level with different forgetting functions and a threshold value, based on which concepts can be moved to a higher level (short → mezzanine → long). The context model proposed in this thesis can be similarly exploited to create a semantic long-term user profile.

Even though the proposed algorithms perform reasonably well in terms of execution time, further work to improving the current performance is possible, and indeed relevant to enhance the scalability of the methods. The most computationally intensive activity in our approach currently takes place in the semantic contextual and preference expansion to a large extent. In the experiments reported here, expansion time ranges from 0.3 to 1 seconds, depending on the user profile, on a KB with 35K concepts and 450K relations. The parameters that most impact the expansion cost are the decay factor and the maximum number of allowed expansion steps (see section 4.5.1), according to our empiric observations in different performance tests. In the aim to investigate the potential transfer of our techniques to commercial applications, further tuning and speeding up of the current implementation is envisioned. Larger KBs will be used for this purpose, such as dbpedia\(^{22}\), which contains 2.18M concepts and 218 M relations extracted from Wikipedia\(^{23}\).

The evaluation work can be further elaborated as well. The size of the evaluated data could be extended with further simulated tasks on the scenario based evaluation, and more users on the user centered approach. The user centered evaluation can be also complemented with more specific user questionnaires than the ones used in our experiments, which were rather general. Works conducted for instance by White (2004b) could provide further insights on how to use post-questionnaires on interactive systems, which could be extended to take into consideration the personalization effect as needed in our research. It would also be possible to simplify the

\(^{22}\) http://www.dbpedia.org

\(^{23}\) http://www.wikipedia.org
system and make it available to the public, in order to obtain a log of sufficient size for a data
driven evaluation approach (Dou et al. 2007).

6.2.2 Semantic resources

As is usual with systems that exploit semantic metadata, the overall performance of the
approach naturally depends on the quality of content annotations, and the richness of the
representation of the metadata within the KB. The practical problems involved in achieving a
good coverage and quality in that area are the object of a large body of research on ontology
construction (Staab and Studer 2004), semantic annotation (Dill et al. 2003; Kiryakov et al.
2004; Popov et al. 2004), semantic integration (Kalfoglou and Schorlemmer 2003; Noy 2004),
and ontology alignment (Euzenat 2004), and are not the focus of this work. Yet this kind of
metadata is not currently available in widespread content corpora such as the Web, although
recent initiatives are taking relevant steps in that direction, such as dbpedia, a large scale
knowledge base in a formal ontological language, which is automatically extracted from
Wikipedia by means of natural language processing techniques.

Our model is nonetheless not highly restrictive with regards to the formal construction of
ontologies. Indeed our proposal makes relatively soft assumptions about the format of the KB,
in the sense that only a set of concepts and a set of relations among them are required. The
generality of our model will also accept more simple knowledge representations. For instance,
the growing corpora of user based annotated content, known as folksonomies (Specia and Motta
2007), may very well suit our model. Folksonomy KBs fits our scheme of concept-related
corpora, as it has content annotated by user tags, and users related to a set of concepts, i.e. the
user generated tags.

An even simpler schema would be possible, based on simple term correlation techniques
(Asnicar and Tasso 1997), where a concept space could be built based on correlation-based
links, were highly correlated concepts (e.g. co-occurrent in common documents) would get a
non-labeled weighted relation between them. Another possible statistical approach would be to
apply dimensionality reduction techniques, such as latent semantic indexing (Sun et al. 2005),
which also output a set of related concepts. The lack of named properties in statistical
approaches can be foreseen as an important limitation, as the heterogeneity of relations is a key
component in our semantic expansion approach.

Testing and evaluating the performance that can be achieved based on such simpler approaches
would provide valuable insights as to the degree to which manual work is needed in the
construction of KBs and involved tradeoffs. On the one hand, our experimental setup uses a
fairly complex KB which is generated independently from the document corpus, without providing complete knowledge coverage. On the other hand, concept spaces produced by social tagging, correlation analysis techniques, or latent semantic indexes, produce simpler KBs but which, by being created from the document corpus, offer a much complete knowledge coverage. It is nonetheless relevant to note that the current experiments were carried out with an ontology that was semi-automatically created and populated by Web scrapping techniques (Popov et al. 2004), and even so we were able to obtain acceptable results, without the effort of manual ontology construction (which would allow for an even higher semantic resource quality).
References


Bauer, T. and D. Leake (2001). Real time user context modeling for information retrieval agents. CIKM '01: Proceedings of the tenth international conference on Information and knowledge management. ACM, Atlanta, Georgia, USA, 568-570.


Martin, I. and J. Jose (2004). Fetch: A personalised information retrieval tool. RIAO'04: Proceedings of the 8th Recherche d'Information Assistée par Ordinateur (computer assisted information retrieval), Avignon, France, 405-419.


Ricardo and Berthier (1999). Modern Information Retrieval, ACM.


SIGIR 2004.

SpringerVerlag.


Sugiyama, K., K. Hatano and M. Yoshikawa (2004). Adaptive web search based on user profile
constructed without any effort from users. WWW'04: Proceedings of the 13th

personalized Web search. WWW '05: Proceedings of the 14th international conference

accuracy. KDD '06: Proceedings of the 12th ACM SIGKDD international conference
on Knowledge discovery and data mining. ACM, Philadelphia, PA, USA, 718-723.


Teevan, J., S. Dumais and E. Horvitz (2005). Personalizing search via automated analysis of
interests and activities. SIGIR '05: Proceedings of the 28th annual international ACM
SIGIR conference on Research and development in information retrieval. ACM,
Salvador, Brazil, 449-456.

Other, Addison.

object labeling. IEEE Transactions on Circuits and Systems for Video Technology
17(3): 298-312.

Thomas, P. and D. Hawking (2006). Evaluation by comparing result sets in context. CIKM '06:
Proceedings of the 15th ACM international conference on Information and knowledge

Model. ESWC'05: Proceedings of the 2nd European Semantic Web Conference.
Springer, Heraklion, Greece, 455-470.

Vallet, D., F. Hopfgartner and J. Jose (2008). Use of Implicit Graph for Recommending
Relevant Videos: A Simulated Evaluation. ECIR'08: Proceedings of the 30th European

Technologies 29(1): 5-14.


151-173.

Wallace, M. and G. Stamou (2002). Towards a context aware mining of user interests for
Proceedings. 2002 IEEE International Conference on, Lausanne, Switzerland, 733-736
vol.731.


Published Work

Journals


Conferences, Workshops and Book Chapters


**Posters and Demonstrations**


Appendices
Appendix A

Detailed Results for the Scenario Based Experiments

This appendix gives more insights on the simulated tasks used and the results obtained from the scenario based experiments. Each task description includes:

- **Topic**: The last query that the user issued.
- **Relevancy to context**: Indications on which documents should be considered as relevant to the actual context of the user, described by the current retrieval session interactions.
- **Relevancy to preferences**: Indications on when a document must be considered relevant to the user interests.
- **Interaction model**: Gives the detailed interaction steps that the user followed before issuing the last query.
- **Precision and Recall**: The resultant PR graph for this specific task

![Task 1. Stock shares: Banking sector companies](image)

**Topc**: Stock shares.
**Relevancy to context**: Relevant documents are those who mention stock shares about companies related to the banking sector.
**Relevancy to preferences**: Consider that the document adjusts to your preferences when the company has a positive interest in the user profile.
**Interaction model**
1. Query input[semantic]: Companies active in the banking sector
2. Opened document: n=3, docId=021452
Appendix A

Task 2. Companies trading in the NYSE: The Hilton Company

**Topic**: Companies that trade on the New York Stock Exchange and their market brands.

**Relevancy to context**: A document is relevant if it mentions the Hilton Company and their hotel chain “Hampton Inn”. The document must indicate the relation between this company and their hotel chain.

**Relevancy to preferences**: Consider that the document adjusts to your preferences when either the company or the company’s brand has a positive interest in the user profile.

**Interaction model**
1. Query input[semantic]: Hilton Company
2. Opened document: n=1, docId=121475

Task 3. Companies and their brands: Homewood Suites hotel chain

**Topic**: Companies and their market brands.

**Relevancy to context**: Relevant documents must contain the hotel chain “Homewood Suites” and the company who owns it: Hilton Co.

**Relevancy to preferences**: Consider that the document adjusts to your preferences when either the company or the company’s brand has a positive interest in the user profile.

**Interaction model**: Query input[semantic]: Homewood suites brand
1. Opened document: n=1, docId=147562
2. Opened document: n=2, docId=012457
3. Opened document: n=3, docId=032122

Task 4. Companies and their brands: Public Companies active in the Food, Beverage and Tobacco sector

**Topic**: Companies and their market brands.

**Relevancy to context**: Relevant documents are those who mention a Public company or a company that has a partial state support together with their market brand (e.g. Kellogs Co. and Kellogs).

**Relevancy to preferences**: Consider that the document adjusts to your preferences when either the company or the company’s brand has a positive interest in the user profile.

**Interaction model**
1. Query input[semantic]: Companies active in the Food, Beverage and Tobacco sector
2. Opened document: n=1, docId=018546
3. Opened document: n=2, docId=078455
Task 5. Companies with high Fiscal Net Income: Japan based companies

**Topic:** Companies with Fiscal Net Income > $100M.
**Relevancy to context:** Relevant documents are those who mention a company based on Japan that has a high average Fiscal Net Income.
**Relevancy to preferences:** Consider that the document adjusts to your preferences when the company has a positive interest in the user profile.
**Interaction model:**
1. Query input[semantic]: Tokyo city
2. Query input[semantic]: Kyoto city
3. Opened document: n=3, docId=12669

Task 6. Companies that have a child organization: Companies that own a Magazine related branch

**Topic:** Companies and their child organizations.
**Relevancy to context:** Relevant documents are those who mention a company that happens to have a child organization that is related to the Magazine sector (e.g. Time Co. and Times Magazine)
**Relevancy to preferences:** Consider that the document adjusts to your preferences when the company has a positive interest in the user profile.
**Interaction model**
1. Query input[semantic]: Companies that own a magazine
2. Opened document: n=3, docId=089415

Task 7. Companies trading in the NYSE: Companies based on the USA

**Topic:** Companies that trade on the New York Stock Exchange.
**Relevancy to context:** Relevant documents are those who trade in the NYSE and are based on the USA
**Relevancy to preferences:** Consider that the document adjusts to your preferences when the company has a positive interest in the user profile.
**Interaction model**
1. Query input[Keyword]: Miami Chicago
2. Opened document: n=1, docId=113425
3. Opened document: n=2, docId=051425
Appendix A

Task 8. Travel: Airline companies that trade on NASDAQ

**Topic:** Travel.
**Relevancy to context:** Relevant documents are those who mention an airline company that trades on the NASDAQ stock exchange
**Relevancy to preferences:** Consider that the document adjusts to your preferences when the company has a positive interest in the user profile.
**Interaction model**
1. Query input[semantic]: Companies that trade on NASDAQ
2. Query input[semantic]: Airline companies

Task 9. Companies trading in the NYSE: Car industry Companies

**Topic:** Companies that trade on the New York Stock Exchange and their market brands.
**Relevancy to context:** Consider a document relevant to the task if it mentions a company active in the car industry sector, together with the brand that has in the market. The document has to explicitly mention this relation of ownership between the company and the brand.
**Relevancy to preferences:** Consider that the document adjusts to your preferences when either the company or the company’s brand has a positive interest in the user profile.
**Interaction model**
1. Query input[keyword]: Mercedes Maybach
2. Opened document: n=1, docId=154235
3. Opened document: n=2, docId=075482

Task 10. Oil energy in Iraq: North American companies active in the energy sector

**Topic:** Oil energy in Iraq.
**Relevancy to context:** Relevant documents are those who mention North American based companies that are active in the energy sector
**Relevancy to preferences:** Consider that the document adjusts to your preferences when the company is (or is partially) publicly owned.
**Interaction model**
3. Query input[semantic]: American companies active in energy sector
4. Opened document: n=1, docId=004585
Appendix B

User Centered Evaluation Task Descriptions

This appendix gives the task descriptions for the three retrieval tasks used in the user centered evaluation approach. Each task description contains:

- **Relevancy to task**: Indication on which documents must be considered relevant to the task, can be considered a task description.
- **Relevancy to preferences**: Indications on when a document must be considered as relevant to the user’s interests.
- **Example of relevant document**: Snippet of a document that is considered relevant to the task

### Task 1: Agreements between companies

**Relevancy to task**

Relevant documents are those that state an agreement between two companies, the article must name the two companies explicitly. For instance, articles about a collaboration or an investment agreement between two companies are considered relevant. Agreements were one company buys totally or partially another company are **NOT** considered relevant.

**Relevancy to preferences**

Consider that the article adjusts to your preferences when one of the mentioned companies has a positive value in your user profile.

**Example of relevant document to the task (excerpt)**

CNN.com - Microsoft, AOL settle lawsuit - May. 30, 2003

Microsoft, AOL settle lawsuit

The two companies also said they have reached a wide-ranging cooperative agreement, under which they will explore ways to allow people using their competing instant message systems to communicate with each other.

Microsoft has also agreed to provide America Online software to some computer manufacturers.
Task 2: Release of a new electronic gadget

Relevancy to task
Relevant documents must mention the release of a new electronic product. Examples of electronic products are music players, gaming devices, PCs, flat screens, mobile devices, etc. It must be a substantial product. For instance, a software program is considered non-relevant.

Relevancy to preferences
Consider that the article adjusts to your preferences when the company or companies that launch the product have a positive value in your user profile.

Example of relevant document to the task (excerpt)
CNN.com - Microsoft, AOL settle lawsuit - May. 30, 2003
CNN.com - Will fans want their MTV PC? - January 13, 2002
Will fans want their MTV PC?
The pioneer of music-oriented TV is looking to tempt media-hungry technophiles with a line of PCs and complementary products set for release early this year. Targeting 18-to-24-year-olds, MTV is looking to let that gadget-happy demographic watch TV, play DVDs, listen to music and browse the Internet, all on one device. The company, a unit of Viacom International, also expects to launch a line of products centered around video-game play, according to a statement.

Task 3: Cities hosting a motor sport related event

Relevancy to task
Relevant documents must describe an upcoming motor sport (e.g. motorcycle, formula one, car rally) together with information on the city that is hosting this event.

Relevancy to preferences
Consider that the article adjusts to your preferences when the hosting city belongs to a country that you have marked as preferred in your user profile. You can also consider relevant those documents that mention a motor sport that has a positive value in your profile.

Example of relevant document to the task (excerpt)
CNN.com - Canadian Grand Prix given reprieve - Oct. 16, 2003
Canadian Grand Prix given reprieve
The International Automobile Federation (FIA) issued a revised calendar with Montreal included as an additional 18th race, to be held on June 13 before the U.S. Grand Prix at Indianapolis on June 20.