



Escuela Politécnica Superior
Departamento de Ingeniería Informática

Exploiting the conceptual space in hybrid recommender systems: a semantic-based approach

Dissertation written by
Iván Cantador Gutiérrez
under the supervision of
Pablo Castells Azpilicueta

Madrid, October 2008

We are leaving the era of search and entering one of discovery. What is the difference? Search is what you do when you are looking for something. Discovery is when something wonderful that you did not know existed, or did not know how to ask for, finds you.

Jeffrey M. O'Brien

“The race to create a ‘smart’ Google”

CNN Money, November 20th 2006

Contents

Abstract	xiii
Resumen	xv
Acknowledgements	xvii
1 Introduction	1
1.1 Motivation	2
1.2 Goals	7
1.3 Contributions	13
1.4 Structure of the thesis	15
1.5 Publications	17
I Context and related work	25
2 Recommender systems	27
2.1 Overview of recommender systems	28
2.1.1 Heuristic-based recommender systems.....	30
2.1.2 Model-based recommender systems	30
2.2 Content-based recommender systems	32
2.2.1 Limitations of content-based recommender systems	36
2.2.2 Examples of content-based recommender systems.....	38
2.3 Collaborative filtering systems.....	41
2.3.1 User-based collaborative filtering	42
2.3.2 Item-based collaborative filtering	44
2.3.3 Limitations of collaborative filtering systems	47
2.3.4 Examples of collaborative filtering systems	50
2.4 Hybrid recommender systems.....	53
2.4.1 Examples of hybrid recommender systems	55

2.5	General limitations of recommender systems	58
2.6	Evaluation of recommender systems	62
2.6.1	Accuracy metrics	62
2.6.2	Non-accuracy metrics	64
2.7	Summary	66
3	Semantic-based information representation and retrieval	67
3.1	Conceptual knowledge representation in Information Retrieval	68
3.2	Ontologies for domain knowledge representation	71
3.3	Ontologies and the Semantic Web vision	74
3.3.1	Indexing and retrieving information	79
3.3.2	Metadata	80
3.3.3	Annotations	82
3.3.4	Ontology description languages	87
3.4	Semantics in Information Retrieval	92
3.5	Semantics in Recommender Systems	94
3.6	Summary	98
II	Recommendation models: an ontology-based proposal	101
4	Content-based recommendation: a semantic-intensive approach	103
4.1	Semantic user profiles and preference extension	104
4.2	Semantic personalised content retrieval	112
4.3	Semantic contextualisation of user preferences	116
4.4	Semantic group profiles for content retrieval	117
4.5	Summary	131
5	Hybrid recommendation: a semantic multilayer approach	133
5.1	Communities of interest	134
5.2	Semantic multilayered communities of interest	136
5.3	Semantic hybrid recommendation models	138
5.4	An example	141
5.5	Summary	145
6	Evaluation of the recommendation models	147
6.1	Evaluation of group-oriented recommendations	148
6.2	Evaluation of hybrid recommendations with a small number of users	154
6.3	Evaluation of hybrid recommendations with a large number of users	161
6.4	Conclusions	168

III Further evaluations: an integrative experiment 171

7 Evaluation platform 173

7.1 News@hand: a semantic-based approach to recommending news	174
7.2 Related work.....	174
7.2.1 Content-based news recommender systems	175
7.2.2 Collaborative news recommender systems	176
7.2.3 Hybrid news recommender systems.....	177
7.3 System architecture	177
7.4 Graphical user interface.....	182
7.4.1 News recommendations.....	183
7.4.2 User feedback.....	185
7.4.3 User profile editor	186
7.5 Summary	188

8 User-centred evaluations in the prototype system 191

8.1 Knowledge base.....	192
8.1.1 Domain ontologies.....	193
8.1.2 Ontology population.....	194
8.2 Item annotation	200
8.2.1 Natural language processing of news contents.....	202
8.2.2 Automatic semantic annotation	204
8.2.3 Annotation database	207
8.3 User profiles	208
8.3.1 Manual definition of semantic preferences	208
8.3.2 Automatic transformation of social tags into semantic preferences	209
8.4 Experiments	218
8.4.1 Evaluation of the ontology population mechanism.....	218
8.4.2 Evaluation of the item annotation mechanism.....	219
8.4.3 Methodology for evaluating the recommendation models	220
8.4.4 Evaluating personalised and context-aware recommendations	222
8.4.5 Evaluation of hybrid recommendations	227
8.4.6 Evaluation of recommendations using semantic preferences obtained from social tags	232
8.5 Conclusions	234

9 Conclusions	237
9.1 Summary and contributions	238
9.1.1 Ontological knowledge representation	238
9.1.2 Semantic content-based recommendations.....	241
9.1.3 Semantic hybrid recommendations	243
9.1.4 Evaluation of the recommendation models.....	244
9.2 Discussion and future work.....	246
9.2.1 Semantic resources.....	247
9.2.2 Recommendation models	249
9.2.3 Evaluation framework.....	250
 A Acronyms	 253
 B News@hand API	 257
B.1 Database manager	258
B.2 Ontology plugin.....	259
B.3 User profile manager	262
B.3.1 User profile memory storage.....	263
B.3.2 User profile ontology handling	265
B.3.3 User profile management.....	265
B.4 Personalised recommenders	266
B.4.1 Semantic content-based recommendation	266
B.4.2 Semantic context-aware recommendation	267
B.4.3 Semantic preference expansion.....	267
B.5 Collaborative recommenders.....	268
B.5.1 Collaborative filtering recommendation.....	268
B.5.2 Semantic multilayer hybrid recommendation	268
B.6 Preference learner.....	269
B.7 Log manager.....	271
 C Introducción	 273
C.1 Motivación.....	274
C.2 Objetivos	280
C.3 Contribuciones	285
C.4 Estructura de la tesis.....	288
C.5 Publicaciones.....	290

D Conclusiones	299
D.1 Resumen y contribuciones	300
D.1.1 Representación del conocimiento ontológica	300
D.1.2 Recomendaciones semánticas basadas en contenido.....	303
D.1.3 Recomendaciones semánticas híbridas	305
D.1.4 Evaluación de los modelos de recomendación.....	307
D.2 Discusión y trabajo futuro	309
D.2.1 Recursos semánticos	309
D.2.2 Modelos de recomendación.....	312
D.2.3 Plataforma de evaluación	313
 References	 315

List of figures

2.1	Components of a recommender system.....	29
2.2	Content-based recommendations.....	32
2.3	Syskill and Webert rating interface and annotation pages (Pazzani & Billsus, 1997).....	39
2.4	News Dude user interface (Billsus & Pazzani, 1999).	40
2.5	User-based collaborative filtering recommendations.	42
2.6	Item-based collaborative filtering recommendations.	45
2.7	Amazon.com collaborative recommendations.	51
2.8	GroupLens rating and recommendation pages (Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994). Predicted scores are shown as bar graphs.	52
2.9	Fab rating page (Balabanovic & Shoham, 1997).	56
2.10	P-Tango user profile editor and on-line newspaper (Claypool, Gokhale, Miranda, Murnikov, Netes, & Sartin, 1999). The former allows the user to choose sections and keywords of interest. The latter provides a slider to enter ratings.....	56
2.11	TiVo sorted list of recommended TV shows (Ali & Van Stam, 2004).....	58
2.12	Example of Recommendation Query Language syntax.....	59
2.13	Three ROC curves with different levels of goodness according to their AUC.	64
3.1	The current Web is oriented to human beings (Catells, 2003).....	75
3.2	Content as it is structured in the current Web (left image) vs. the same content as it might be structured in the Semantic Web (right image).....	77
3.3	Inverted index structure (Baeza-Yates & Ribeiro Neto, 1999).	79
3.4	Vision of parallel semantic networks in the Semantic Web.	82
3.5	Example of a web page about the life of the painter Vicent Van Gogh.	83
3.6	Example of a first annotation level in a web page, where content keywords are identified and assigned to a set of raw categories.	84
3.7	Simple annotation of content keywords using HTML meta-tags and document-oriented XML tags.....	84
3.8	Example of basic metadata structure where each semantic annotation contains string-valued properties.....	85
3.9	XML-based structured annotations.....	85
3.10	Example of structured and interrelated metadata.	86

3.11	Example of metadata represented in the form of an ontology.....	87
3.12	The layered technologies of the Semantic Web (Passin, 2004).	87
3.13	Example of RDF(S) graph.	89
3.14	Example of RDF(S) syntax.	90
3.15	Example of RDQL query.	90
3.16	Example of OWL expressivity capabilities.	91
4.1	Ontology-based user profiles and item descriptions.	104
4.2	Representation of user preferences as concepts of domain ontologies.	106
4.3	Semantic preference extension.	108
4.4	Example of semantic preference extension computation.	109
4.5	Pseudocode of the semantic spreading algorithm.	111
4.6	Personalised ontology-based content retrieval.	113
4.7	Example of user and item weighted-concept vectors.	115
4.8	Contextualised semantic user preferences.	117
4.9	Screenshot of the personalisation framework used to evaluate ontology-based group modelling strategies.....	122
4.10	Group formation following the additive utilitarian strategy. The ranked list of items for the group would be (d ₅ -d ₆ , d ₈ , d ₄ -d ₁₀ , d ₁ , d ₉ , d ₂ , d ₇ , d ₃).	123
4.11	Group formation following the multiplicative utilitarian strategy. The ranked list of items for the group would be (d ₆ , d ₅ , d ₈ , d ₁₀ , d ₄ , d ₉ , d ₂ -d ₈ , d ₁ , d ₃).	123
4.12	Group formation following the Borda count strategy. The ranked list of items for the group would be (d ₆ , d ₅ , d ₁ , d ₄ -d ₈ , d ₉ , d ₁₀ , d ₂ , d ₇ , d ₃).	124
4.13	Group formation following the Copeland rule strategy. The ranked list of items for the group would be (d ₅ , d ₁ , d ₆ , d ₉ , d ₄ , d ₈ , d ₁₀ , d ₂ , d ₇ , d ₃).	125
4.14	Group formation following the approval voting strategy. The ranked list of items for the group would be (d ₄ -d ₅ -d ₆ -d ₈ -d ₁₀ , d ₁ -d ₇ -d ₉ , d ₂ -d ₃).	126
4.15	Group formation following the least misery strategy. The ranked list of items for the group would be (d ₆ , d ₅ , d ₄ -d ₈ -d ₁₀ , d ₇ , d ₂ , d ₉ , d ₃ , d ₁).	126
4.16	Group formation following the least misery strategy. The ranked list of items for the group would be (d ₁ -d ₅ -d ₉ , d ₂ -d ₄ -d ₆ -d ₈ , d ₃ -d ₁₀ , d ₇).	127
4.17	Group formation following the average without misery strategy. The ranked list of items for the group would be (d ₆ -d ₅ , d ₈ , d ₄ -d ₁₀ , d ₂ , d ₇).	127
4.18	Group formation following the fairness strategy. The ranked list of items for the group could be (d ₅ , d ₆ , d ₄ , d ₈ , d ₁₀ , d ₇ , d ₁ , d ₂ , d ₉ , d ₃), following the user selecting order u ₁ , u ₂ and u ₃ , and setting L=3.	128
4.19	Group formation following the plurality voting strategy. The ranked list of items for the group could be (d ₅ , d ₆ , d ₄ , d ₈ , d ₁₀ , d ₁ , d ₉ , d ₂ , d ₇ , d ₃), following the user selecting order u ₁ , u ₂ and u ₃ , and setting L=3.	128
4.20	Group recommendations by the combination of ontology-based user profiles.	130

4.21	Group recommendations by the combination of personalised ranked item lists.....	131
5.1	Semantic concept clustering based on shared interests of the users.....	136
5.2	Groups of users obtained from shared semantic concept clusters.	137
5.3	Multilayered CoI built from shared semantic concept clusters.....	138
5.4	Average precision vs. recall curves for users assigned to cluster 1 (left), cluster 2 (centre), and cluster 3 (right). The graphics on top show the performance of the UP and UP-q models. The ones below correspond to the NUP and NUP-q models.	145
6.1	Screenshot of the personalisation framework used to evaluate ontology-based group modelling strategies.....	148
6.2	Set of pictures used in the evaluation of group-oriented recommendations..	149
6.3	Average distances d_1 and d_2 for the subject profile and ranking combination methods.....	151
6.4	Average distances d_1 and d_2 for user profile and ranking combination methods.	152
6.5	Average subject satisfaction.....	153
6.6	Average normalised linear addition user satisfaction.....	154
6.7	Minimum inter-cluster distance at different concept clustering levels.	155
6.8	Symmetric user similarity matrices at layers 1, 2, 3 and 4 between user profiles u_i and u_j , ($i, j \in \{1, 20\}$) obtained at clustering level $Q=4$. Dark and light grey cells represent respectively similarity values greater and lower than 0.5. White cells mean no relation between users.....	157
6.9	Avg. precision vs. recall curves for users assigned to the clusters obtained with the UP (black lines) and UP-q (grey lines) models at levels $Q=6$ (graphics on the left), $Q=5$ (graphics in the middle), and $Q=4$ (graphics on the right) clusters. Dotted lines represent the results achieved without preference spreading.....	160
6.10	Screenshot of a MovieLens page, where most recent and rated movies are shown.....	161
6.11	Screenshot of an IMDb page, where information about a movie is shown: title, plot, date, genres, director, writer, cast, etc.	162
6.12	MovieLens-IMDb ontology. White boxes correspond to IMDb entities, while coloured boxes are associated to classes that store the information obtained from MovieLens rating repository.	163
6.13	Cumulative distributions of IMDb features (genres, actors, directors, languages, countries, keywords) per movie.....	165
6.14	Cumulative distribution mappings of our recommender values into MovieLens ratings.....	166

6.15	MAE for our content-based (CB), and UP, UP-q, NUP and NUP-q hybrid recommenders.	167
6.16	MAE for UP-q and CF recommenders built with 100 (left) and 1,000 (right) users.	168
7.1	Architecture of News@hand.	178
7.2	Recommendation and user profiling modules of News@hand.	179
7.3	A typical news recommendation page in News@hand.	183
7.4	Example of meta-information provided by News@hand to news items.	184
7.5	News@hand panel to establish constraints for group recommendations.	185
7.6	Pop-up window to tag a news item in News@hand.	185
7.7	Pop-up window to evaluate a news item in News@hand.	186
7.8	Semantic preference editor and ontology browser of News@hand.	187
7.9	Personal data editor of News@hand.	188
7.10	Personal rating manager of News@hand.	188
8.1	Disambiguation information of the term “apple” in Wikipedia.	195
8.2	Wikipedia categories for the term “Teide”.	195
8.3	Automatic RSS feed extraction and semantic annotation in News@hand....	200
8.4	Semantic annotation mechanism.	201
8.5	XML output provided by Wraetlic after the NLP of a text.	203
8.6	News@hand ontology browser with auto-complete search functionalities. ..	208
8.7	The tag filtering architecture.	211
8.8	The tag filtering process.	212
8.9	Pseudocode of the compound noun and misspelling detection mechanism.	214
8.10	Pseudocode of the morphologically similar term group technique.	216
8.11	Pseudocode of the WordNet synonym merging technique.	217
8.12	Average precision values for the top 5, 10 and 15 news items, taking into account those items evaluated as relevant to the task goal and the user profile.	227
8.13	Average Mean Squared Error of item-based collaborative filtering (CF) and semantic multilayer hybrid (UP-q) recommendation strategies using 10%, 20%, ..., 90% of the available ratings for building (training) the models, and the rest for testing.	231
8.14	Matching Flickr and del.icio.us tags to Wikipedia ontology. Graphs show how many tags each user had in the raw tag cloud, how many tags were filtered, how many corresponded to a Wikipedia entry, and finally how many categories were selected to represent the given tag cloud.	234
B.1	The three-layer JDBC connection manager architecture.	258
B.2	The ontology manager architecture.	260
B.3	The user profile management architecture.	263

List of tables

2.1	Common limitations of content-based recommendation techniques.....	38
2.2	Common limitations of collaborative filtering techniques.	50
2.3	General limitations of recommendation techniques.....	61
3.1	Different ontology classification schemas.....	74
3.2	Categorisation scheme of metadata about the concept “cat” proposed by Aristotle (Breitman, Casanova, & Truszkowski, 2007).	81
4.1	Parameters of the semantic spreading algorithm.	110
4.2	Categorisation of group recommendation approaches and examples.	121
5.1	Users’ interest degrees for each topic, and expected user clusters to be obtained.	142
5.2	Initial concepts for each of the six considered topics.	143
5.3	User clusters and associated similarity values between users and clusters. The maximum and minimum similarity values are shown in bold and italics respectively.	143
5.4	Concepts assigned to the obtained user clusters classified by semantic topic.....	144
6.1	User clusters and associated similarity values between users and clusters obtained at concept clustering levels $Q=4, 5, 6$	156
6.2	Concept clusters obtained at clustering level $Q=4$	158
6.3	Information about the size of the IMDb and MovieLens data and knowledge bases used in our experiments.	163
8.1	Number of classes and instances available in News@hand knowledge base.....	192
8.2	Some classes belonging to the domain ontologies of News@hand.....	193
8.3	Database entries created after searching for the term “java”.....	197
8.4	Database entries automatically created for the term “ny”.	198
8.5	Average number of annotations per news item.	207
8.6	Conversion of special characters to a base form.....	213
8.7	Average class and ontology population accuracies.	218
8.8	Average number of annotations per news item, and annotation accuracies... ..	219
8.9	Functionalities to be evaluated in each testing case.....	221

8.10	Summary of the search tasks performed in the experiment.....	222
8.11	Experiment tasks configurations.....	223
8.12	Topics and concepts allowed for the predefined user profiles in the evaluation of personalised and context-aware recommenders.	224
8.13	Topics and concepts of the manually-defined user profiles in the evaluation of personalised and context-aware recommenders.	225
8.14	Topics and concepts allowed for the user profiles in the evaluation of the hybrid recommenders.	229
8.15	Average relevance values for the top 5 ranked news items recommended by News@hand when using semantic profiles obtained from social tags.....	233
B.1	Main classes of the database manager component.....	259
B.2	Main ontology entity classes.....	260
B.3	Main ontology plugin classes.	261
B.4	Main ontology plugin repository classes.	262
B.5	Main ontology annotation classes.	262
B.6	Main classes of the user profile memory storage component.	264
B.7	Main classes of the user profile ontology handling component.....	265
B.8	Main classes of the user profile management component.	266
B.9	Main classes of the semantic content-based recommendation component. ..	266
B.10	Main classes of the semantic contextualisation component.....	267
B.11	Main classes of the semantic preference expansion component.....	267
B.12	Main classes of the collaborative filtering component.....	268
B.13	Main classes of the semantic multilayer hybrid recommendation component.....	269
B.14	Main classes of the clustering component.	269
B.15	Main classes of the semantic preference learning component.....	270
B.16	Summary of the log database tables and attributes. The most relevant attributes are in bold fonts.	271

Abstract

The ever-increasing volume and complexity of information flowing into our daily lives challenge the limits of human processing capabilities in a wide array of information seeking and e-commerce activities. In this context, users need help to cope with this wealth of information, in order to reach the most interesting products, while still getting novelty, surprise and relevance.

Recommender systems suggest users products or services they may be interested in, by taking into account or predicting their tastes, priorities or goals. For that purpose, user profiles or usage data are compared with some reference characteristics, which may belong to the information objects (content-based approach), or to other users in the same environment (collaborative filtering approach). Inspired by Information Retrieval and Machine Learning techniques, both approaches are based on statistical or heuristic models that attempt to capture the correlations between users and objects.

Commercial applications like *Amazon online store* (www.amazon.com), *Google News* (news.google.com) or *YouTube* (www.youtube.com), are examples of significant success stories of recommendation techniques. However, several limitations of the current recommender systems remain, such as the sparsity of user preference and item content feature spaces, the difficulty of recommending items to users with few preferences declared, or the lack of flexibility to incorporate contextual factors into the recommendation processes.

Some of these limitations can be related to a limited understanding and exploitation of the semantics underlying both user profiles and item descriptions. In this respect, an enhancement of the semantic knowledge, and its representation, describing interests and contents can be envisioned as a potential direction to deal with those limitations.

This thesis explores the development of an ontology-based knowledge model to link the (explicit and implicit) meanings involved in user interests and resource contents. Upon this knowledge representation, several content-based and collaborative recommendation models are proposed and evaluated. The models have been integrated in a prototype, in which they are empirically tested with real users. The prototype is designed as an open, flexible evaluation platform of further use in addressing open research problems in the area of recommender systems.

Resumen

El incesante crecimiento en el volumen y complejidad de la información que nos abruma diariamente reta a los límites de la capacidad de procesamiento humana en una amplia gama de actividades de búsqueda y comercio electrónico. En este contexto, se hace necesario el ayudar a afrontar esa sobrecarga presentando a los usuarios los productos más interesantes, a la vez que ofreciendo novedad, sorpresa y relevancia.

Los sistemas de recomendación sugieren a los usuarios aquellos productos o servicios que les pueden interesar teniendo en cuenta o prediciendo sus gustos, preferencias u objetivos. Para alcanzar este fin, perfiles de usuario o históricos de uso son comparados con algunas características de referencia que pueden estar asociadas a los objetos de información (aproximación basada en contenido), o al entorno social de los usuarios (aproximación basada en filtrado colaborativo). Inspiradas en técnicas de áreas del conocimiento como la Recuperación de Información y el Aprendizaje Automático, las aproximaciones anteriores hacen uso de modelos estadísticos o de heurísticas que intentan capturar las correlaciones entre usuarios y objetos.

Aplicaciones comerciales como *Amazon* (www.amazon.com), *Google News* (news.google.com) o *YouTube* (www.youtube.com) han demostrado el gran éxito de las estrategias de recomendación existentes. Sin embargo, diversas limitaciones de los sistemas de recomendación actuales siguen vigentes, como la poca densidad de los espacios de preferencias de usuario y atributos de contenido, la dificultad de recomendar ítems a usuarios con pocas preferencias declaradas, o la falta de flexibilidad para incorporar variables contextuales en los procesos de recomendación.

Algunas de estas limitaciones se pueden asociar a un limitado entendimiento y explotación de la semántica subyacente tanto en los perfiles de usuario como en las descripciones de objeto. De este modo, una mejora en la representación semántica del conocimiento que permita describir intereses y contenidos podría ayudar a solventar esas limitaciones.

Esta tesis explora el desarrollo de un modelo de representación de conocimiento basado en ontologías que permite enlazar los significados explícitos e implícitos en los intereses de usuario y en los contenidos de recursos. A partir de la representación de conocimiento propuesta se presentan y evalúan una serie de modelos de recomendación basados en contenido y colaborativos. Por otra parte, la posterior integración de estos modelos en un prototipo ha ofrecido primeros resultados empíricos con usuarios reales, y da la oportunidad de abordar problemas pendientes de resolver en el campo de los sistemas de recomendación.

Acknowledgements

First of all, I would like to express my gratitude to Pablo Castells, my advisor, for giving me the opportunity to work with him and his group, for his wise guidance, advice and suggestions, and for allowing me to pursue my research and training through the attendance of already numerous conferences, and the participation in projects with people from different nationalities and institutions.

I would like to give very special thanks to José R. Dorronsoro. He was my supervisor during the first two years of my PhD. By working with him I learnt skills that are essential for research, from addressing a scientific problem to writing a paper. He also taught me how to teach, always insisting that I should have in mind the students' academic background and learning difficulties. These lessons represent an invaluable legacy, but above all, what I will never forget is his comprehension and support in difficult moments.

My most sincere thank you to my colleagues at NETS group, David Vallet, Mariano Rico, Miguel A. Corella, José M. Fuentes, Fernando Díez, and Sergio López, for the unforgettable time we spent together. Special thanks to Alejandro Bellogín. Without his help in implementing and evaluating News@hand during the last year, I would not have finished the thesis on time.

I also want to thank the people I met during my research visits in the UK. It is not always easy to live in a foreign country, and they made me feel like I was at home. I thank all the guys from KMi at the Open University, and my great housemates, Farah Huzair, Alex Borda, Nourdin Bejjit, Patrick Palicica, and Frank Schiller. I had a wonderful time in Milton Keynes with all of them. Special thanks to Enrico Motta for giving me the opportunity of working with his team, to Vanessa López, who kindly and patiently listened to my doubts and questions about AquaLog, and to Carlos Pedrinaci for the hopeful encouragement to finish the PhD. From ECS at the University of Southampton, I thank Nigel R. Shadbolt for accepting my request to do an internship there. Many thanks to Harith Alani and Martin Szomszor. The work we did together was very satisfying and useful for me. Thanks to Benedicto Rodríguez for introducing and taking care of me in the group, and to Manuel Salvadores, with whom I had long and interesting conversations over a cup of coffee and a couple of croissants. My stay at So'ton was a lovely experience.

I warmly acknowledge the people of GTI group for the shared moments in the lunches in our department, and in the meetings of MESH project, Javier Molina, Victor Valdés, José M. Martínez, Jesús Bescós, Álvaro García, Fabrizio Tuburzi, and

Fernando López. Regarding the MESH project, I would like to thank Jérôme Picault and Myriam Ribière, from Motorola, for their extraordinary work and help during the implementation and testing of News@hand. I would also like to thank Pedro Concejero, Jorge Munuera, and Paulo Villegas (the TID guys), for making the project journeys so enjoyable.

From the EPS at UAM, my kind thanks to all my fellow teachers, Manuel Cebrián, Jaime Moreno, Antonio C. Fernández, Rosa M. Carro, Álvaro Ortigosa, Estefanía Martín, Pablo A. Haya, Miguel A. Mora, Alejandro Echeverría, Francisco Saiz, Estrella Pulido, Ana M. Gonzalez, Pablo Varona, Eduardo Serrano, and Luis F. Lago. Special thanks to Juana Calle, who always efficiently helped me with the administrative tasks.

Beyond UAM realms, I extend my gratitude to Daniel Olmedilla, Rubén Lara, and Enrique Alfonseca, for their interest and comments on the thesis. Huge thanks to my friends Ignacio J. García, Israel Díez, and Iker Aguirre for always being there, listening to my stories and grumbles about the doctorate studies.

I would like to dedicate this paragraph to Miriam. There are no words to express what I feel for all the moments we spent together. The last seven years have been the most wonderful of my life thanks to her. There were very sad moments as well, but they have totally disappeared from my mind. Who knows what will happen in the future. I wish our paths would meet again someday.

Last but not least I thank my parents, José and Antonia, and my brothers, Borja and Rubén, for their continuous support and trust in me. They were and always will be my pillars. This thesis is dedicated to the four of them.

To my family

Chapter 1

Introduction

A general overview of the thesis is provided in this chapter, focusing on the definition of the problems that motivated the work, an outline of the proposals developed to address them, and the resulting outcomes of the research.

Section 1.1 presents the motivation which gave rise to this work, stating the problems to be confronted, and enumerating the limitations of the existing approaches reported in the literature. Section 1.2 defines the scope of this study by setting the partial objectives to be achieved. Next, Section 1.3 summarises the specific contributions of the research presented herein. Section 1.4 describes the structure of this document, and finally, Section 1.5 lists the publications that resulted from the research undertaken in this thesis.

1.1 Motivation

During the last two decades, a point has been reached in the era of telecommunications in which the huge amount of available information overwhelms our daily activities. The amount of new content produced every day (news, scientific articles, movies, songs, web pages, etc.), largely overcoming human processing capabilities, and the unstructured nature of most of such information, raise important issues for its effective use and utility.

This information overload brought on the need to design systems capable of performing an efficient **information retrieval** upon billions of documents. The information these systems manage may consist not only of web pages, but also of other types of text documents, as well as any kind of image, video or audio files, properly annotated with textual metadata. The documents to be retrieved are commonly annotated with keywords that describe aspects of their content in a summarised way. For text documents, annotations may consist of e.g. those terms which are more “informative” (e.g., those that appear more frequently on single documents, but are uncommon in the collection of documents as a whole). For multimedia contents, annotations may involve concepts which are manually declared by users, or are automatically extracted by means of some advanced content analysis technique at the signal level. These annotations can be used to generate index tables that establish weighted associations between each keyword and the documents in which they appear, using data structures that allow a very fast retrieval of the documents associated with a given keyword (Baeza-Yates & Ribeiro Neto, 1999). Search engines essentially differ from each other in their mechanisms to generate annotations and indices, as well as in the algorithms to retrieve and rank documents from keywords.

In this scenario, the user may know his objectives, related to the information he wants to retrieve and the possible descriptions of it. If so, when looking for specific documents, he is able to input queries stated as lists of related terms. For instance, if the user is planning his holidays, and is interested in gathering documents containing information about the Republic of Indonesia (which is a country comprising more than 17,000 islands in the Pacific Ocean), he could enter queries like “Indonesia”, “Republic of Indonesia”, “Indonesia islands”, etc.

There is no doubt about the success that information retrieval systems have obtained over the last years offering their content search services on the Internet. Given a query, commercial search engines like *Google* or *Yahoo!* select and display lists of hundreds to billions of potentially relevant documents, in a ranked way (based on similarity measures between queries and annotations). Often, the results expected by the user are placed at the beginning of the lists. However, on occasions those documents are positioned in such a way that the user will actually never reach them.

Therefore, there are several aspects that have not been satisfactorily addressed by current systems. Among them, one of the most important is **semantic ambiguity**. Let us suppose that the user from the previous example focuses his search about Indonesia in one of its islands: Java. In order to do this, he introduces the query “Java” into a web search engine. Expecting to find documents about that island, he actually finds out that the first results obtained with that query correspond to documents that do not contain that concept at all. Instead, he is displayed all sort of web pages related to the well-known programming language with the same name. The first results concerning the island are far away from the top of the list.

In this example, the results should have been prioritised according to the meaning of the term “Java” in each case. Disambiguation could have been possible if the system had considered the set of queries previously entered by the user about Indonesia. “Semantic distances” could have been measured in some way between previous query terms (i.e., Indonesia, republic, island, etc.) and terms appearing in indexed documents that were related to the two previously described meanings of the word “Java”, i.e., the Indonesian island and the programming language. Thus, it could have been deduced that, with high probability, the user in that “context” was interested in obtaining documents associated with the first meaning. In an information retrieval environment, the consideration of context (obtained from recent actions of the user in the system) has often been called **contextualised information search**.

The semantic context, as defined in the previous example, can be considered as a set of user preferences with a short life span within a specific user’s session in the system. Initially, these preferences are temporary, and could be described as current interests or goals of the user. However, if they were repeated in time with a certain frequency (e.g., daily), they could be incorporated into a permanent description of the user’s interests, which is known in the literature as *user profile*. Analogously to the context, this profile could then be used to modify the order in which the query results are displayed.

Let us consider two users. The first one has a profile built up (either manually or automatically) with concepts related to tourist lodgings, travel agencies, etc. The second one, on his side, is a computer science engineer who defined his profile with concepts related to operating systems, computer applications, etc. Let us suppose that both users enter the query “Java” into the same web search engine, whose internal information retrieval algorithm is able to take into account a user’s preferences when retrieving contents for him. Now, the result lists provided to each user should be different. The first one should receive a list in which the first documents involve the Indonesian island, while the second one should get a different list containing results on the programming language. This type of capability is known in the literature as **personalised information search**.

Of course, the current context does not necessarily need to always agree with the preferences of the user profile. Getting on with the former example, the computer science engineer could be interested in getting information about the island of Java, even for professional reasons because, say, he may need to attend a meeting in that island. A balance between contextualisation and personalisation might be the key to obtain more precise and user-relevant search results.

Anyway, up to this point, and independently from considering context or personal preferences, the user is aware of his necessities and information search targets, and seems to know how to express them with keyword-based queries. However, this is not always the case. Every day, out on the street, reading newspapers, watching television, listening to the radio, or chatting with a friend, we discover facts of whose existence we were not aware, but which are important or interesting for us, or even affect our lives in a transcendental manner.

“**Word of mouth**” is a technique that consists in passing information by means of verbal communication, especially with recommendations, in an informal, personal way, rather than by communication media, advertisement, organised publications, or traditional marketing. It is typically considered a spoken communication, although dialogs in the Internet in, for example, blogs, forums or e-mails, are usually included in this definition. Advertisement based in word of mouth is highly estimated by vendors. It is believed that this form of communication has a valuable credibility due to the source where it comes from. People are more inclined to trust the word of mouth than more formal advertisement techniques, because the communicator is less likely to have an ulterior interest (e.g., it does not try to sell something). Also, people tend to trust other people they know.

In words of Jeffrey M. O'Brien, extracted from his article “The race to create a ‘smart’ Google” published in CNN Money on November 2006:

We are leaving the era of search and entering one of discovery. What is the difference? Search is what you do when you are looking for something. Discovery is when something wonderful that you did not know existed, or did not know how to ask for, finds you.

In order to face this new challenge, in the mid nineties, **recommender systems** rise up as an independent research field of Information Retrieval and Artificial Intelligence. The objective of the researchers focuses then on estimating the relevance of those items which the user has not seen yet, independently from the fact he had not searched for them. The way in which this estimation is performed allows the distinction of two main recommendation strategies (Adomavicius & Tuzhilin, 2005): content-based and collaborative filtering.

Content-based recommender systems calculate the relevance of an item for a user according to the relevance that other “similar” items seemed to have for him in the past. Similarity measures between items are based on features of their contents.

Thus, for example, a touristic recommender system could suggest lodgings in several countries of Oceania to a user with a flying history to Indonesia, a member country of that continent.

In these systems, the belief that recommendations faithfully reflect the user's preferences, obtained from past actions and personal evaluations or ratings on several items, is usually considered an advantage. However, this can become a great disadvantage. Since we only consider the user's profile, the space of new, potentially interesting items is limited to those that share characteristics with previously seen items. *Content over-specialisation* and lack of diversity (a.k.a. *portfolio effect*) in recommendations are currently two of the most notable problems of this type of strategies.

To solve these problems, collaborative filtering systems calculate the relevance of an item for a user by considering the relevance that other items had in the past for "similar" people. In this case, similarity measures are calculated from correlations between item evaluation patterns. For instance, let us suppose that the majority of people who have travelled to Jakarta, the Indonesian capital, have also travelled to its neighbour country Singapore, giving positive feedback about their stays. A collaborative filtering system could recommend lodgings in Singapore to a user with a travelling history to Indonesia, even though he had never shown an explicit interest for the former country in his profile.

The collaborative filtering approach does not limit the recommendation space, and avoids over-specialisation and lack of content diversity. However, it incorporates its own limitations, among which, one of the most important is the "*grey sheep*" problem, which is defined as the difficulty of recommending items to particular users with uncommon preferences (evaluation patterns), very different to those of the rest of users.

This problem could be addressed incorporating a content-based strategy. In fact, in order to jointly face the characteristic limitations of each of the two exposed types of recommendations – content-based and collaborative filtering – a combination of both is proposed in the literature under the title of **hybrid recommender systems**.

Currently, there is a growing interest for hybrid recommendation systems, which are becoming an integral part of a great number of important e-commerce web portals like *Amazon.com*, where book recommendations are offered, *FilmAffinity.com*, where films are recommended, *Last.fm*, which recommends songs and music groups, or *Google News* (news.google.com), which makes personalised news recommendations. In all of them, the use of classical recommendation models has been very successful. However, the current generation of recommender systems still requires additional improvements to obtain more effective algorithms that might be used in a greater variety of applications. These improvements include, among others:

- The application of strategies that take into account initial situations where there are only a few user preferences or evaluations (*cold-start problem*), and situations where there is a low density of correlations between evaluations due to the high relative number of users or items (*sparsity problem*).
- The addition of *contextual information* to the recommendation processes.
- The use of more *flexible algorithms*, which can be adapted by the user, or that are able to make recommendations not only to a single user, but to a group of users with similar tastes and interests.

The way in which these aspects can be partially or totally resolved in a satisfactory way represent open research lines in the area. The difficulties associated to the previously exposed aspects have been addressed independently, but no recommender model has been established that solves them in an integrative and effective way.

This thesis contends that an important reason for these difficulties is the limited comprehension and exploitation of the underlying semantics, both in user preferences and in the content characteristics of the items. Classic models describe user and item profiles by keyword lists (in content-based approaches) or by numerical evaluations (in collaborative filtering approaches). The components of these lists are apparently unrelated to each other, and their (semantic) meanings are not taken into consideration when making recommendations.

In recommender systems, the necessity for a **semantic representation of knowledge** which allows a simple, scalable, and portable description of the involved domains is being manifested in recent works (Middleton, Roure, & Shadbolt, 2004; Mobasher, Jin, & Zhou, 2004; Anand & Mobasher, 2007; Sieg, Mobasher, & Burke, 2007; Shoval, Maidel, & Shapira, 2008). Since the users' tastes and interests are defined over the content of items to be recommended, user and item profiles have to be built up from a common knowledge representation. This representation should be understandable by humans, and processable by machines (computer programs). Additionally, it should be easy to extend and adapt it to other domains. The ideal would be that information gathered by a recommender system could also be exploited by other systems, even if they managed items with a very different nature. In order to achieve this, it would be convenient to use standard knowledge representation models and languages.

In this thesis, the use of **ontologies** as the conductive channel to satisfy the previous need is proposed. Both in computer sciences and information sciences, an ontology is a formal representation of a set of concepts belonging to a domain, as well as the existent relations between those concepts (Gruber, 1993). It can be used to describe that domain and/or to reason about its properties. Ontologies are used as a way of representing knowledge about the world or a part of it, in fields as

diverse as Artificial Intelligence, Semantic Web, Software Engineering, Biomedical Computer Science or Library Science. Some of the fundamental elements of an ontology are: *individuals* (instances or basic information objects), *classes* (categories, sets, types of objects), *attributes* (aspects, properties, characteristics that an individual or class might have), and *relations* (special attributes that relate pairs of classes and/or individuals).

More specifically, this work proposes a three-folded knowledge representation model, in which a space for interrelated semantic concepts (by means of ontologies, and describing one or several application domains) is incorporated between the user and the item spaces. In this model, user and item profiles are defined with vectors which components are weighted concepts of the ontology space. On top of that form of knowledge representation, a set of recommendation mechanisms is proposed and evaluated. These mechanisms are oriented to one or more users, combine content-based and collaborative filtering strategies, and incorporate semantic contextual information obtained from annotations of items that were involved in recent user actions and evaluations. An implementation and integrated start-up of the previous mechanisms into a prototype system is also presented.

The opportunity to incorporate metadata into the user profiles and the descriptions of the recommended items, as well as the ability to infer knowledge from the existing semantic relations between concepts of the domain ontologies, will be key aspects of the exposed proposals.

1.2 Goals

The final goal of this thesis is the implementation and evaluation of enhanced recommendation models incorporating a conceptual space between the preferences of the users and the content features of the items to recommend. By identifying and exploiting the underlying relations between users and items in the above conceptual space, the proposed models should address limitations existing in current recommender systems.

Rooted in classic information retrieval techniques, content-based recommender systems generally represent the user preferences and item features as term vectors. With these representations, vector similarities are calculated (e.g., by computing the cosine of the angle formed by the vectors) as relevance measures of the items for the users. Thus, for example, suppose a user profile defined by the vector $\mathbf{u} = (\text{indonesia} = 0.7; \text{java} = 0.9; \text{island} = 0.2)$, where each term is assigned a weight in $[0,1]$ that measures the intensity of the interest of the user for that concept. Suppose an item whose content is described by the vector $\mathbf{d} = (\text{java} = 0.6; \text{island} = 0.5)$. A

simple recommendation model which computes the cosine between the vectors \mathbf{d} and \mathbf{u} would return a preference value of 0.38:

$$\text{pref}(\mathbf{d}, \mathbf{u}) = \cos(\mathbf{d}, \mathbf{u}) = (0.6 \cdot 0.9 + 0.5 \cdot 0.2) / \left(\sqrt{0.6^2 + 0.5^2} \cdot \sqrt{0.7^2 + 0.9^2 + 0.2^2} \right) = 0.38$$

This model leads to two main problems. The first problem is associated to the *semantic ambiguity* of the terms. In the previous example, “java” references a preference of the user for the Indonesian island. Now let us consider two new items $\mathbf{d}_1 = (\text{java} = 0.4; \text{hotel} = 0.8)$ and $\mathbf{d}_2 = (\text{java} = 0.4; \text{software} = 0.8)$. In \mathbf{d}_1 , the component “java” corresponds to the above island, but in \mathbf{d}_2 , it is associated to the computer programming language that shares the same name. The meanings underlying the term “java” are totally different for the two items. However, the computation of the similarities between the user profile \mathbf{u} and the vectors \mathbf{d}_1 and \mathbf{d}_2 results in $\text{pref}(\mathbf{d}_1, \mathbf{u}) = \text{pref}(\mathbf{d}_2, \mathbf{u}) = 0.19$, giving the same preference to both items, when the second one potentially lacks interest for the user. In this case, the distinction between the two semantic concepts, for example by declaring $\mathbf{d}_1 = (\text{island}; \text{java} = 0.4; \text{hotel} = 0.8)$, $\mathbf{d}_2 = (\text{programming}; \text{java} = 0.4; \text{hotel} = 0.8)$ and $\mathbf{u} = (\text{indonesia} = 0.7; \text{island}; \text{java} = 0.9; \text{island} = 0.2)$, is essential for not producing undesirable recommendations.

The second problem is the assumption of term independence. Now let us suppose the following two items: $\mathbf{d}_1 = (\text{java} = 0.4; \text{hotel} = 0.8)$ and $\mathbf{d}_2 = (\text{java} = 0.4; \text{archipelago} = 0.8)$. In this case, the term “java” is related to the Indonesian island in the two items, and the user preference value assigned to both of them is again $\text{pref}(\mathbf{d}_1, \mathbf{u}) = \text{pref}(\mathbf{d}_2, \mathbf{u}) = 0.19$. Nonetheless, taking into account the user profile $\mathbf{u} = (\text{indonesia} = 0.7; \text{java} = 0.9; \text{island} = 0.2)$, we could assume that item \mathbf{d}_2 should have a higher relevance because the concept “archipelago” (i.e., a set of islands) is more related to the preference “island” than the concept “hotel”, included within item \mathbf{d}_1 . The need of considering (semantic) relations between concepts when recommendations have to be made is evident in this example.

The conclusion we can reach about the previous two limitations has been already mentioned in the literature (Balabanovic & Shoham, 1997; Ungar & Foster, 1998): in many current recommender systems, there is a ***lack of understanding and exploitation of the underlying semantics*** about the tastes and interests of the users, and the contents of the recommended items. To confront such problem, the first goal established in this thesis is:

- G1. The definition of a formal non-ambiguous knowledge representation which takes into consideration relations between concepts.** We shall study proposals based on ontologies. Both user profiles and item descriptions will be formed by concepts (classes and instances) belonging to multiple domain ontologies. The semantic relations, which will be defined in the ontologies, should be exploited by the different recommendation models to be explored.

In an ontological representation, the semantic relations enrich the meaning of each concept. For example, if a user shows a high generic interest for aspects related to islands, having a profile $\mathbf{u} = (\text{island} = 0.9)$, we could assume that he might be also keen on specific islands. Thus, the extension of his profile to $\mathbf{u} = (\text{island} = 0.9; \text{island:java} = 0.1)$ not only might be correct, but also beneficial to find more relevant items. In this case, the preference expansion has been done through the “instance of” property that relates a class (island) to a specific individual (Java). There are other types of relations. Some of them are common to any ontological representation, such as the relation “subclass of”: “continental island” and “oceanic island” are subclasses of “island”. Other relations, however, are arbitrary defined within the ontology domains. For example, in an ontology about Geography, it could exist a relation “capital of”: a “city” is the capital of a “country”, “Jakarta” is the capital of “Indonesia”.

The preference expansion makes the user profiles less sparse in the conceptual space, since they cover larger areas of the latter. The *sparsity* of preferences and evaluations is thus a problem which has been addressed in several works (Billsus & Pazzani, 1998; Sarwar, Karypis, Konstan, & Riedl, 2000). It is closely related with the *cold-start problem*, which is based on the difficulty of recommending items when a user is new in a system, having none or few preferences defined (Schein, Popescul, & Ungar, 2001). These two effects appear in both content-based and collaborative approaches. To address them, the *need of enriching the semantic descriptions* offered by an ontology-based knowledge representation causes the second goal of the thesis:

- G2. The enrichment of concept-based user profiles and item descriptions by exploiting the relations existing between their concepts.** We shall investigate strategies which spread the user preferences and item content features towards concepts linked by relations existing in the domain ontologies. The spreading algorithms should be designed according to issues such as the attenuation of the expanded preference weights, or the possibility of finding loops in the spreading paths. Furthermore, we shall evaluate the effect of the semantic propagation on the results obtained with the recommendation models to be proposed.

Apart from enriching the semantic descriptions of users and items, an ontology-based knowledge representation enhances the understanding of their meanings. This fact might facilitate the comprehension of the concepts involved in the current context of a content retrieval or recommendation environment. In classic systems, the *preference contextualisation* is a very complex task. It is in fact an open research line, and has been studied in recent works (Räck, Arbanowski, & Steglich, 2006; Anand & Mobasher, 2007; Vallet, Castells, Fernández, Mylonas, & Avrithis, 2007). In Section 1.1, the contextualisation was motivated with a particular example of term disambiguation. The concepts annotating results of latest queries (e.g., Indonesia, republic, island, etc.) were used to infer that “Java”, in that current context, was referencing to the Indonesian island, instead to the programming language. Another possible application of contextualisation is the focusing or reinforcement of user preferences. Those concepts that have been recently referenced (e.g., by item evaluations) could be taken into account more strongly by the recommendation models.

The proposed knowledge representation also incorporates mayor flexibility in the recommendation processes, allowing the application of *user profile merging* strategies. Several vectors describing the preferences of a set of users could be easily combined to generate an individual group profile, which would further used for recommending items in a collective way. As an illustrative example, let u_1 and u_2 be two users whose profiles are respectively defined by the vectors $\mathbf{u}_1 = (\text{indonesia} = 0.6; \text{java} = 0.9)$ and $\mathbf{u}_2 = (\text{java} = 0.1; \text{island} = 0.4)$. Assuming that the vectors are combined using the average sum of their components, the resultant group profile would be $\mathbf{u}_g = (\text{indonesia} = 0.3; \text{java} = 0.5; \text{island} = 0.2)$. In the literature, group-oriented recommendations have been proposed in very different applications, such as the collective suggestion of music compositions (McCarthy & Anagnost, 1998), movies (O'Connor, Cosley, Konstan, & Riedl, 2001), touristic attractions (Ardissono, Goy, Petrone, Segnan, & Torasso, 2003) or television shows (Ali & Van Stam, 2004).

The two previous issues are examples that evidence the *need of flexibility* in recommender systems, and motivate the third goal of this research:

- G3. Building a personalised recommendation model which allows the incorporation of semantic context, and the adaptation to the preferences of one or more users.** We shall propose a content-based model that makes use of our ontology-based knowledge representation. This model should be flexible to context-aware or group-oriented recommendations. We shall evaluate the effect of adding semantic context into the basic model, and shall study several strategies for the merging of user profiles.

As already mentioned in Section 1.1, content-based recommender systems focus in the preferences of an individual user, and do not exploit the benefits offered by techniques based on the “word of mouth” phenomena to find out items relevant for the user that are not explicitly related to his preferences, but are recommended to people with similar tastes and interests. The fact of taking into account only one user profile may lead to *content over-specialisation* and *lack of diversity* (a.k.a. portfolio effect) in the recommendations.

To solve these problems, collaborative filtering strategies were proposed. These approaches are based on the computation of similarities (correlations) among user and item profiles, and their effectiveness has been demonstrated by their success in current commercial applications. However, they incorporate new limitations. One of them is that called the *grey sheep problem*, which consists of the difficulty of recommending items to people with particular preferences which are very rare in the rest of the user profiles, and do not allow finding correlations among them. Hybrid recommendation models combining features based on content and collaborative filtering might be suitable to confront the above problem.

In general, the comparison of users and items is globally done, so that partial but strong similarities might be lost. For example, two people with a high coincidence in their favourite places to visit might have very divergent interests about the type of accommodation they usually look for. The opinions of these two people concerning touristic destinations might be highly valuable for both of them, but could be ignored by a travel recommender system which computes a low global similarity for their profiles. Again, let u_1 and u_2 be two users whose profiles are defined by the vectors $\mathbf{u}_1 = (\text{java} = 0.4; \text{singapore} = 0.6; \text{hotel} = 0.8)$ and $\mathbf{u}_2 = (\text{java} = 0.5; \text{camping} = 0.7)$. The cosine-based similarity between these two vectors is 0.25:

$$\text{sim}(u_1, u_2) = \cos(\mathbf{u}_1, \mathbf{u}_2) = (0.4 \cdot 0.5) / \left(\sqrt{0.4^2 + 0.6^2 + 0.8^2} \cdot \sqrt{0.5^2 + 0.7^2} \right) = 0.25.$$

Now let us suppose the system is able to identify and separately group preferences related to touristic locations and preferences associated to types of accommodation. Based on these two conceptual groups, the user profiles can be split in two different subprofiles. For user u_1 :

$$\mathbf{u}_1^{\text{locations}} = (\text{java} = 0.4; \text{singapore} = 0.6), \mathbf{u}_1^{\text{accommodation}} = (\text{hotel} = 0.8).$$

For user u_2 :

$$\mathbf{u}_2^{\text{locations}} = (\text{java} = 0.5), \mathbf{u}_2^{\text{accommodation}} = (\text{camping} = 0.7).$$

Computing the cosine of the angle formed by the vectors belonging to the two preference groups we obtain new similarities between the users. In the case of the group related to touristic locations, the similarity value duplicates the global one.

$$\text{sim}_{\text{locations}}(u_1, u_2) = \cos(\mathbf{u}_1^{\text{locations}}, \mathbf{u}_2^{\text{locations}}) = (0.4 \cdot 0.5) / (\sqrt{0.4^2 + 0.6^2} \cdot \sqrt{0.5^2}) = 0.53.$$

In the case of group related to accommodation types, the similarity is null:

$$\text{sim}_{\text{accommodation}}(u_1, u_2) = \cos(\mathbf{u}_1^{\text{accommodation}}, \mathbf{u}_2^{\text{accommodation}}) = 0 / (\sqrt{0.8^2} \cdot \sqrt{0.7^2}) = 0.$$

If the system were able to discern the current context, it could make very different but accurate recommendations in each case. Following the previous example, if we only take into consideration the preferences for touristic locations, user u_2 can be suggested vacation packages to Singapore, since this city was positively evaluated by user u_1 , with whom the user shares an interest for Java island. On the contrary, if we only consider the preferences for accommodation types, user u_2 is not suggested any item based on the profile of user u_1 .

Motivated by the *difficulty of recommending items to users with uncommon preferences*, or to users that share interests under specific semantic scopes, the fourth goal of this thesis is the following:

- G4. Building hybrid models which combine user profiles in a collaborative way at several semantic scopes, based on different groups of shared preferences.** We shall define hybrid recommendation strategies which group shared user preferences, and compute similarities among users and items based on the semantics underlying the identified preference groups. We shall compare the results obtained with the proposed models against those provided by classic collaborative filtering techniques.

The evaluation of recommender systems is also an open research line in the literature (Herlocker, Konstan, Terveen, & Riedl, 2004; Adomavicius & Tuzhilin, 2005). For the proposals to be explored in this thesis, the setting of an experimentation framework raises questions about the definition of the domain ontologies, the semantic annotation of items, and the building of user profiles.

With the purpose of carrying out an *evaluation of the ontology-based knowledge representation and recommendation models*, the fifth and last goal in the thesis is:

- G5. The integration and evaluation of all the recommendation approaches in a prototype system.** We shall build a recommender system to validate the proposals. During the system implementation we expect to design, develop and evaluate techniques that automatically create the knowledge bases (i.e., processes for ontology instantiation/population, and item semantic annotation), and ease the manual definition of user profiles.

1.3 Contributions

The works presented in this thesis contribute to the development of models and algorithms that make use of semantic-based technologies to address limitations existing in the current recommender systems. Our main contributions are summarised in the following points:

- **Exploitation of ontology capabilities to enrich state-of-the-art recommender systems functionalities.** We propose an ontology-based knowledge representation model that is richer and less ambiguous than keyword-based or item-based models. The definition of user preferences and item features through semantic concepts belonging to domain ontologies facilitates the end-user's understanding of his profile and the obtained content-based recommendations. The model provides an adequate grounding for the representation of coarse to fine-grained user interests (e.g., interest for items such as a football team, an actor, a stock value), and can be a key enabler to deal with the subtleties of user preferences. An ontology provides further formal, computer-processable meaning on the concepts (who is coaching a team, an actor's filmography, financial data on a stock), and makes it available for a recommender system to take advantage of. Furthermore, ontology standards support inference mechanisms that can be used to enhance recommendations, so that, for instance, a user interested in movies concerning *history facts* (superclass of *war*) is also recommended movies about *wars*. Also, a user keen on videos about *Spain* can be assumed to like videos in which *Madrid* appears, through the *locatedIn* transitive relation. The recommendation models presented in this research make use of the above semantic inference mechanisms. First sections of Chapter 4 describe the proposed ontology-based knowledge representation model, explaining in more detail its advantages.
- **Development of novel semantic content-based and collaborative recommendation approaches.** We propose several hybrid recommendation models that merge semantic content-based and collaborative information. In these models, domain ontology relations are exploited to extend the user preferences and item annotations. In real scenarios, user profiles tend to be very scattered (having a relative number of preferences/evaluations with respect to the total of available concepts), particularly in those cases where the users have to explicitly declare their interests. 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 their inputs. On the other hand, applications in

which an automatic preference learning algorithm is applied tend to recognise very general characteristics of user preferences, thus producing profiles that may entail a lack of expressivity. Apart from the user profiles, item descriptions can be also enriched. Collaborative filtering systems suffer from the well-known “cold start” problem (Burke, 2002), in which a new item cannot be recommended until it is rated by a user. In this situation, no collaborative information exists, the use of content-based approaches is essential, and techniques to enhance the content descriptions might be very beneficial to find correlations between item characteristics and user interests. For all the above reasons, the implemented recommendation methods make use of a technique that extends user preferences and item annotations according to the semantics existing in the domain ontologies. This technique is based on Constrained Spreading Activation (CSA) strategies (Cohen & Kjeldsen, 1987; Crestani & Lee, 2000). Specifically, the weights of user preferences and item annotations are iteratively propagated through the ontology relations, generating extended versions of the user profiles and item descriptions used to provide the final personalised recommendations. The semantic propagation technique is presented in Chapter 4, and the hybrid recommendation models are explained in detail in Chapter 5. The evaluation of the models is described in Chapter 6.

- **Presentation of novel ideas for semantic context-aware and group-oriented recommendation.** In general, recommender systems are inflexible in the sense that they support a predefined and fixed set of recommendations. Most of them make use of single criterion ratings, and only recommend individual items to individual users, not dealing with aggregation of items and/or users. For these reasons, the end-user cannot customise recommendations according to his needs. The ontology-based knowledge and user profile representations proposed in this thesis enable the development of strategies that provide flexibility to existing recommendation processes. Specifically, we use an ontology query model for personalised content retrieval, we include contextualised information in the recommendations, we study mechanisms that combine several user profiles for recommending items to groups of people, and we design a technique that make use of multi-criteria ratings. Last sections of Chapter 4 describe the above recommendation mechanisms, and Chapter 6 presents experiments conducted to evaluate them in an isolated way.
- **Implementation of an ontology-based recommender system.** The recommendation models proposed in this thesis were evaluated with real users and artificial datasets created from external sources. Isolate and independently experiments showed positive results, endorsing the feasibility

of the proposals. However, we noticed the need of carrying out additional experimentation in an environment where we could integrate the above models combining their outputs, and study the difficulties arisen from the extrapolation of the models to a realistic application. For this reason, we implemented *News@hand*, a news recommender system in which text news contents are annotated with concepts (classes and instances) of a set of ontologies covering a number of different domains. When building the system, several research challenges appeared, and novel solutions have been proposed. In particular, we develop an ontology population technique (i.e., a technique for the creation of ontology instances), an automatic mechanism to annotate the news articles, and a strategy that transforms tags or keywords into existing ontology concepts. Chapter 7 describes the architecture and graphical user interface of *News@hand*, and Chapter 8 exposes the experiments performed to evaluate the system recommendation functionalities, and its semantic instance, annotation and preference creation mechanisms.

1.4 Structure of the thesis

The main objective of this thesis is the study of how models and techniques based on semantic technologies can be applied to confront some of the current limitations existing in recommender systems. The wide nature of this research area implies to treat very different fields, such as user profiling, group modelling, and personalised content retrieval. Taking into account that a very large description of state-of-the-art in all these fields at the beginning of the thesis might be unappealing for the reader, the literature review has been distributed in the different parts in which this document is structured. However, aiming to offer a preliminary overview of the context of the work, its two first chapters have been dedicated to an overall exploration of the main addressed research areas, recommender systems and semantic-based knowledge representation and retrieval, and a more detailed explanation of those approaches that can be considered as the intersection of them.

The thesis has been divided into three parts. The first part gives background knowledge and general literature surveys in recommender systems and semantic-based knowledge representation and retrieval models, identifies the current limitations of recommender systems, and describes recent approaches to confront some of these limitations using semantic-based technologies. The second part contains descriptions and evaluations of the semantic-based recommendation models proposed herein. Finally, the third and last part presents the implementation and empirical evaluation of the previous proposals in a web-based recommender system, explains the novelties and advantages of the system, and concludes with general

discussions and future research lines.

Additionally, the contents of the thesis have been distributed in individual chapters as follows:

Part I. Context and related work

- **Chapter 2** provides an overview of the state-of-the-art in recommender systems, distinguishing between content-based, collaborative filtering, and hybrid recommendations. For each of them, the strengths and weaknesses are described, and several representative applications are presented.
- **Chapter 3** motivates and defines the use of semantic technologies in knowledge representation and information retrieval models. From the existing approaches, the chapter focuses its attention in those more related to the recommender systems area. Specifically, it describes relevant state-of-the-art techniques in semantic search and personalised ontology-based content retrieval.

Part II. Recommendation models: an ontology-based proposal

- **Chapter 4** introduces the ontology-based knowledge and user profile representations underlying the proposals of the thesis. Using these representations, a basic content-based recommendation model is described in the chapter. Extensions of this model to support context-aware and group-oriented recommendations are also presented.
- **Chapter 5** explains how the ontology-based knowledge and user profile representations described in the previous chapter are used to build semantic multilayered communities of interests. The (implicit) social relationships emerged in these communities are exploited for recommendation purposes, motivating a set of hybrid recommendation models that are described at the end of the chapter.
- **Chapter 6** exposes the experiments performed to evaluate the content-based collaborative recommendation models proposed in the previous chapters. Some partial conclusions are given.

Part III. Further evaluations: an integrative experience

- **Chapter 7** describes the implementation of the proposed recommendation models in a web evaluation platform. The architecture and the graphical user interface of the prototype system are detailed in the chapter.
- **Chapter 8** presents empirical evaluations with the implemented recommender system, showing the benefits of the studied ontology-based approaches.

- **Chapter 9** finally concludes the thesis with overall discussions and future research lines to be investigated with further adaptations and extensions on the prototype system.

Each of the above chapters starts with a brief introduction of the topics addressed in it, and a paragraph describing its internal structure. The chapters that present experimental results end with their corresponding partial conclusions. The rest of the chapters, on the other hand, conclude with summary sections.

In addition to the chapters, there are several appendixes containing additional information that is relevant, but not central for the purposes of the thesis:

- **Appendix A** lists all the acronyms used in this document.
- **Appendix B** provides the API of the implemented prototype system.
- **Appendix C** contains the translation into Spanish of the *Introduction* chapter.
- **Appendix D** contains the translation into Spanish of the *Conclusions* chapter.

1.5 Publications

The basis of the proposals of this thesis arises from the ontology-based knowledge representation model introduced in (Vallet, Fernández, & Castells, 2005). This model has been exploited in different research fields such as semantic search (Castells, Fernández, & Vallet, 2007), and personalised context-aware content retrieval (Vallet, Castells, Fernández, Mylonas, & Avrithis, 2007). As novel extensions of these works, the publications that this thesis has yielded are classified in this section by the chapters and research topics they are related to.

Chapter 4

Personalised and context-aware content retrieval

The ontology-based knowledge representation and the personalised context-aware content retrieval model presented in the chapter were used for generating personalised summaries of different multimedia content sources. A description of this application is given in:

- Dolbear, C., Hobson, P., Vallet, D., Fernández, M., Cantador, I., & Castells, P. (2007). Personalised Multimedia Summaries. *Book chapter in "Semantic Multimedia and Ontologies: Theory and Applications"*, pp. 165-183. Springer-Verlag. Edited by Y. Kompatsiaris, and P. Hobson. ISBN: 978-1-84800-075-9.

In this work, the exploitation of the suggested semantic contextualisation technique consistently results in better performance with respect to simple personalisation. The described experiments show how the contextualisation

approach significantly enhances personalisation by removing out of-context user interests, and leaving the ones that are indeed relevant in the ongoing course of action.

A second application of the personalised and context-aware recommendation models for automatic adaptation in multimedia content delivery environments and infrastructures is presented in:

- Cantador, I., López, F., Bescós, J., Castells, P., & Martínez, J. M. (2008). Enhanced Descriptions for Personalized Retrieval and Automatic Adaptation of Audiovisual Content Retrieval. *Book chapter in "Personalization of Interactive Multimedia Services: A Research and Development Perspective"*. Nova Science Publishers. Edited by J. J. Pazos-Arias, C. Delgado, and M. López. ISBN: 978-1-60456-680-2.

This work focuses on a set of initiatives and achievements addressing the automatic adjustment of multimedia content to fit a wide variety of support infrastructures. The provided comprehensive view on multimedia adaptation comprises low to high-level adaptation methods from the ranking of content units according to background user interests in different scenarios (e.g., presence vs. absence of an explicit user query, single vs. multiple users, etc.) to media adaptation techniques of different usage environments (terminals, networks, codecs, players, user preferences, etc.).

Group profiling for content retrieval

Additionally to the previous applications, the proposed ontology-based user profile representation was adapted for the design of various novel group profile modelling strategies. The description and evaluation of this proposal can be found in:

- Cantador, I., Castells, P., & Vallet, D. (2006). Enriching Group Profiles with Ontologies for Knowledge-Driven Collaborative Content Retrieval. *Proceedings of the 1st International Workshop on Semantic Technologies in Collaborative Applications (STICA 2006), at the 15th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 2006)* (pp. 358-363). Manchester, UK: IEEE Computer Society Press, ISBN 0-7695-2623-3.

In this work, assuming the fact that we have a set of semantic user profiles associated to people with shared tastes and interests, we studied the feasibility of applying strategies based on social choice theory (Masthoff, 2004) for merging multiple individual preferences in a personalisation framework from a knowledge-based multimedia retrieval system. Combining several profiles with the considered group modelling strategies we sought to establish how humans recommend an optimal ranked item list for a group, and how they measure the satisfaction of a given

item list. The performed theoretical and empirical experiments demonstrate the benefits of using semantic preferences, and exhibit which user profile combination strategies could be appropriate to a collaborative environment.

Chapter 5

Social networking and Communities of Interest

Once the group modelling strategies were studied, the next step in the conducted research was the implementation of a clustering algorithm to find those sets of user profiles with similar characteristics. The approach is presented in:

- Cantador, I., & Castells, P. (2006). Building Emergent Social Networks and Group Profiles by Semantic User Preference Clustering. *Proceedings of the 2nd International Workshop on Semantic Network Analysis (SNA 2006), at the 3rd European Semantic Web Conference (ESWC 2006)*, (pp. 40-53). Budva, Montenegro.

The proposed algorithm is based on the ontological representation of the domain of discourse where user interests are defined. The ontological space takes the shape of a semantic network of interrelated domain concepts. Taking advantage of the relations between concepts, and of the weighted preferences of users for the concepts, we cluster the semantic space obtaining sets of concepts that represent common topics of interest. After this, user profiles are partitioned by projecting the concept clusters into the set of preferences of each user. The resultant user profile partitions can finally be exploited to compare the individual preferences at different semantic levels, and find several communities of users sharing interests.

Semantic multilayer hybrid recommendation

According to the different subsets of preferences obtained with our clustering algorithm, users can be compared in such a way that several, rather than just one, (weighted) links can be found between two individuals. These “multilayered” social relations were used for modelling a set of hybrid recommendation techniques in:

- Cantador, I., & Castells, P. (2006). Multi-Layered Ontology-based User Profiles and Semantic Social Networks for Recommender Systems. *Proceedings of the 2nd International Workshop on Web Personalisation, Recommender Systems and Intelligent User Interfaces (WPRSIUI 2006), at the 4th International Conference on Adaptive Hypermedia (AH 2006)*. Dublin, Ireland.

Moreover, including more relevant experiments with real user profiles, the previous content-based collaborative recommendation models were discussed in the following work:

- Cantador, I., & Castells, P. (2006). Multi-Layered Semantic Social Networks Modelling by Ontology-based User Profiles Clustering: Application to Collaborative Filtering. *Proceedings of the 15th International Conference on Knowledge Engineering and Knowledge Management – Managing Knowledge in a World of Networks (EKAW 2006)* (pp. 334-349). Pödebrady, Czech Republic: Lectures Notes in Artificial Intelligence, 4248. Springer-Verlag, ISBN 3-540-46363-1.

Chapter 6

Evaluation of the recommendation models

Following the previous works, additional evaluations of the hybrid recommendation models are exposed in:

- Cantador, I., Castells, P., & Bellogín, A. (2007). Modelling Ontology-based Multilayered Communities of Interest for Hybrid Recommendations. *Proceedings of the 1st International Workshop on Adaptation and Personalisation in Social Systems: Groups, Teams, Communities (SociUM 2007), at the 11th International Conference on User Modelling (UM 2007)*. Corfu, Greece.

In this case, instead of evaluating the models with a rather limited number of manually-defined user profiles, we automatically generated cents of user profiles merging the information of the well-known MovieLens¹ and IMDb² repositories. Specifically, we transformed the public MovieLens ratings into weighted user semantic preferences for IMDb movie characteristics. With the obtained user profiles we evaluated our recommendation models showing again the feasibility of the proposals.

All our personalised context-aware and group-oriented recommendation approaches were gathered in the following work:

- Vallet, D., Cantador, I., Fernández, M., & Castells, P. (2006). A Multi-Purpose Ontology-based Approach for Personalized Content Filtering and Retrieval. *Proceedings of the 1st International Workshop on Semantic Media Adaptation and Personalisation (SMAP 2006)* (pp 19-24). Athens, Greece.

This paper received an invitation to be extended and published in a chapter book:

- Cantador, I., Fernández, M., Vallet, D., Castells, P., Picault, J., & Ribière, M. (2007). A Multi-Purpose Ontology-based Approach for Personalised Content Filtering and Retrieval. *Book chapter in “Studies in Computational Intelligence”, vol. 93*, pp. 25-51. Springer-Verlag. Edited by M. Wallace, M. Angelides, and P. Mylonas. ISBN: 978-3-540-76359-8.

¹ MovieLens repository, GroupLens Research, <http://www.grouplens.org/>

² Internet Movie Database, IMDb, <http://www.imdb.com/>

Finally, the application of multilayered Communities of Interest to group modelling and hybrid recommendations was accepted as two journal papers:

- Cantador, I., & Castells, P. (2008). Extracting Multilayered Semantic Communities of Interest from Ontology-based User Profiles: Application to Group Modelling and Hybrid Recommendations. *Computers in Human Behaviour, special issue on Advances of Knowledge Management and Semantic Web for Social Networks*. Elsevier. In press.
- Cantador, I., Bellogín, A., & Castells, P. (2008). A Multilayer Ontology-based Hybrid Recommendation Model. *AI Communications, special issue on Recommender Systems*. IOS Press. In press.

Chapter 7

Implementation of an ontology-based recommender system

In addition to the evaluation of the recommendation models in an isolated way, we identified the need of integrating all of them in a prototype recommender system, which would be public for the research community, and would allow us to make more sophisticated and realistic experiments. The presentation of such system appears in:

- Cantador, I., Bellogín, A., Castells, P. (2008). News@hand: A Semantic Web Approach to Recommending News. *Proceedings of the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2008)*. Hannover, Germany. Lecture Notes in Computer Science, vol. 5149, pp. 279-283. Springer-Verlag. ISBN 978-3-540-70984-8.

News@hand is a news recommender system which applies our semantic-based knowledge representation and recommendation techniques to describe and relate news contents and user preferences, in order to produce enhanced personalised news suggestions.

During the development of the system, several research challenges arose: the population of the domain ontologies, the automatic semantic annotation of items, and the obtention of user preferences from social tags. The approaches to address the above problems were introduced in:

- Cantador, I., Szomszor, M., Alani, H., Fernández, M., & Castells, P. (2008) Enriching Ontological User Profiles with Tagging History for Multi-Domain Recommendations. *Proceedings of the 1st International Workshop on Collective Semantics: Collective Intelligence and the Semantic Web (CISWeb 2008), at the 5th European Semantic Web Conference (ESWC 2008)*. Tenerife, Spain. CEUR Workshop Proceedings, vol. 351, pp. 5-19, ISSN 1613-0073.

This work presents a novel strategy which filters raw collaborative tagging information (i.e., folksonomies) to incorporate it into an ontological knowledge representation. For such purpose, semantic information available on external resources such as WordNet (Miller, 1995) and Wikipedia³ is exploited. Early evaluations of the technique are also explained in the paper.

Chapter 8

Evaluations with the implemented ontology-based recommender system

Finally, experimentation with *News@band* system to evaluate the combination of the personalised recommendation models is described in:

- Cantador, I., Bellogín, A., Castells, P. (2008). Ontology-based Personalised and Context-aware Recommendations of News Items. *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2008)*. Sydney, Australia.

The combination of a model that personalises the order in which news articles are shown according to the user's long-term interest profile, and other model that reorders the news item lists taking into account the current semantic context of interests, showed significant improvements on the experimental tasks performed.

Related contributions

In parallel with the publications arising from this thesis, additional contributions have been made in related issues on recommender systems. Specifically, we have investigated 1) novel multi-criteria recommendation mechanisms, 2) semantic user profiling strategies utilising cross-folksonomy information, and 3) analysis techniques of relevant user preferences in a recommender system using machine learning algorithms. The first proposal was integrated in *News@band* system, described in Chapter 7, the second is an extension of our semantic user preference building mechanism explained in Section 8.3.2, and the third was done with user log information generated from the experiments conducted with *News@band* that are described in Section 8.4.4.

Collaborative evaluation and multi-criteria recommendations

The implementation of a tool for collaborative ontology evaluation and reuse was presented in:

³ Wikipedia, the free encyclopaedia, <http://www.wikipedia.org/>

- Fernández, M., Cantador, I., & Castells, P. (2006). CORE: A Tool for Collaborative Ontology Reuse and Evaluation. *Proceedings of the 4th International Workshop on Evaluation of Ontologies for the Web (EON 2006), at the 15th International World Wide Web Conference (WWW 2006)*. Edinburgh, UK. CEUR Workshop Proceedings, vol. 179, ISSN 1613-0073.

Among other novelties, this tool provides a collaborative recommendation mechanism based on multi-criteria ratings. Due to its own relevance for the recommender systems community, the multi-criteria recommendation algorithm was explained in detail in other publication:

- Cantador, I., Fernández, M., & Castells, P. (2006). A Collaborative Recommendation Framework for Ontology Evaluation and Reuse. *Proceedings of the International Workshop on Recommender Systems, at the 17th European Conference on Artificial Intelligence (ECAI 2006)*, (pp. 67-71). Riva del Garda, Italy.

This recommendation framework was designed to confront the challenge of evaluating those ontology features that depend on human judgements, and are by their nature more difficult for machines to address. Taking advantage of collaborative filtering techniques, the system exploits the ontology ratings and evaluations provided by users to recommend the most suitable ontologies for a given domain.

The above system was transformed into a web application, and was modified incorporating new capabilities during the collaborative problem domain definition, and ontology recommendation processes:

- Cantador, I., Fernández, M., & Castells, P. (2007). Improving Ontology Recommendation and Reuse in WebCORE by Collaborative Assessments. *Proceedings of the 1st International Workshop on Social and Collaborative Construction of Structured Knowledge (CKC 2007), at the 16th International World Wide Web Conference (WWW 2007)*. Banff, Canada. CEUR Workshop Proceedings, vol. 273, ISSN 1613-0073.

In this paper, the multi-criteria recommendation algorithm is empirically evaluated, showing relevant benefits for the application.

User modelling based on folksonomy information

We have proposed a method for the automatic consolidation of user profiles across popular social networking sites, and for the subsequent semantic modelling of their interests utilising Wikipedia as a multi-domain model:

- Szomszor, M., Cantador, I., Alani, H. (2008). Correlating User Profiles from Multiple Folksonomies. *Proceedings of the 19th ACM Conference on Hypertext and Hypermedia (Hypertext 2008)*. Pittsburgh, Pennsylvania, USA. ACM 2008. ISBN 978-1-59593-985-2.
- Szomszor, M., Alani, H., Cantador, I., O'Hara, K., Shadbolt, N. (2008). Semantic Modelling of User Interests based on Cross-Folksonomy Analysis. *Proceedings of the 7th International Semantic Web Conference (ISWC 2008)*. Karlsruhe, Germany. Lecture Notes in Computer Science. Springer-Verlag.

In these papers, we evaluate how much can be learned about the user's preferences from the combination of tag-based user profiles defined in different social networking sites, and in which domains the knowledge acquired is focussed. Results show that far richer interest profiles can be generated for users when multiple tag-clouds are combined.

Analysis of relevant preferences in recommender systems

In addition to the proposal of techniques that provide item recommendations from available preference data, or the definition of strategies for learning the latter, we have also investigated a mechanism to find out which preferences are really relevant to obtain accurate recommendations.

- Bellogín, A., Cantador, I., Castells, P., Ortigosa, A. (2008). Discovering Relevant Preferences in a Personalised Recommender System using Machine Learning Techniques. *Proceedings of the Preference Learning Workshop (PL 2008), at the 8th European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2008)*. Antwerp, Belgium.

In this work, we present a meta-evaluation methodology that applies machine learning techniques to analyse log information of *News@hand* in order to discover (and rank) the user preferences and system settings which are suitable for accurate recommendations. We also show how the proposed methodology can be used to enhance the system evaluation itself.

Part I

Context and related work

Chapter 2

Recommender systems

Recommender systems are software applications that provide personalised advice to users about products or services they might be interested in. They recommend items of interest to users based on preferences they have expressed, either explicitly or implicitly.

Based on the way in which item suggestions are estimated for different users, the following two main types of recommender systems are commonly distinguished: 1) *content-based recommender systems*, in which a user is recommended items similar to those he preferred in the past, and, 2) *collaborative filtering systems*, in which a user is recommended items that people with similar tastes and preferences liked in the past. Due to the limitations of each of the above strategies, combinations of them have been investigated in the so-called *hybrid recommender systems*, empirically demonstrating their better effectiveness.

In this chapter, we provide an overview of issues, terminology and techniques related to recommender systems. In Section 2.1, we formalise the concept of recommendation, describe the basic components of any recommender system, and introduce the existing general types of recommenders. The different recommendation approaches, content-based, collaborative filtering, and hybrid, are explained respectively in Sections 2.2, 2.3 and 2.4. For each of them, representative system examples, limitations, and statements of possible solutions are also presented. Finally, in Section 2.5, we conclude with a review of the metrics that have been proposed in the literature to evaluate recommendation approaches.

2.1 Overview of recommender systems

The recommendation problem can be formulated as follows (Adomavicius & Tuzhilin, 2005). Let $\mathcal{U} = (u_1, u_2, \dots, u_M)$ be the set of all registered users in a recommender system, and let $\mathcal{I} = (i_1, i_2, \dots, i_N)$ be the set of all possible items users have access to in the system. Let $g: \mathcal{U} \times \mathcal{I} \rightarrow \mathcal{R}$, where \mathcal{R} is a totally ordered set (e.g., non negative integers or real numbers within a certain range), be a utility function such that $g(u_m, i_n)$ measures the gain or usefulness of item i_n to user u_m . Then, for each user $u_m \in \mathcal{U}$, we want to choose an item $i^{\max, u_m} \in \mathcal{I}$, unknown to the user, which maximises the utility function g . More formally:

$$\forall u_m \in \mathcal{U}, \quad i^{\max, u_m} = \arg \max_{i_n \in \mathcal{I}} g(u_m, i_n). \quad (2.1)$$

In recommender systems, the utility of an item is usually represented by a *rating*, which measures how much a specific user is interested in the item. Depending on the application, the ratings can either be specified by the users, or be computed by the system.

Each element of the user space \mathcal{U} can be described with a *profile* that may include several demographic characteristics, such as gender, age, nationality, marital status, etc., and/or some information about the user's tastes, interests and preferences. Analogously, each element of the item space \mathcal{I} may be described with a set of characteristics or features. For example, in a movie recommender system, movies can be described not only by their titles, but also by their genres, principal actors, etc.

The way in which such user profiles and item descriptions are defined is a key point in any recommender system. However, it is not the only factor that influences the efficiency and effectiveness of the recommendation processes. For example, the mechanism to capture user preferences is critical. Users are not willing to spend time explicitly declaring their tastes and interests, and automatic preference learning strategies tend to capture general patterns of user behaviour. Further, the methods to compare and combine user profiles and item descriptions within a specific recommendation algorithm may drastically impact the resulting accuracy.

Figure 2.1 shows the basic components of a recommender system. Firstly, a user profile learning module (explicitly or implicitly) captures the preferences from the user. Once the system “knows” about the user's tastes and interests, it performs a recommendation algorithm that compares and/or combines user profiles and item descriptions. The item characteristics are stored in a database. It is important to note that depending on the recommendation strategy not all the available items are candidates to being retrieved, as we shall see. From now on we define the “choice set” as the set of items that can be recommended by the system. In subsequent

figures, the colours of the items belonging to the choice set represent particular groups of related items based on content description features. That is, items of the same colour share common content features (e.g., movie genres in a movie database).

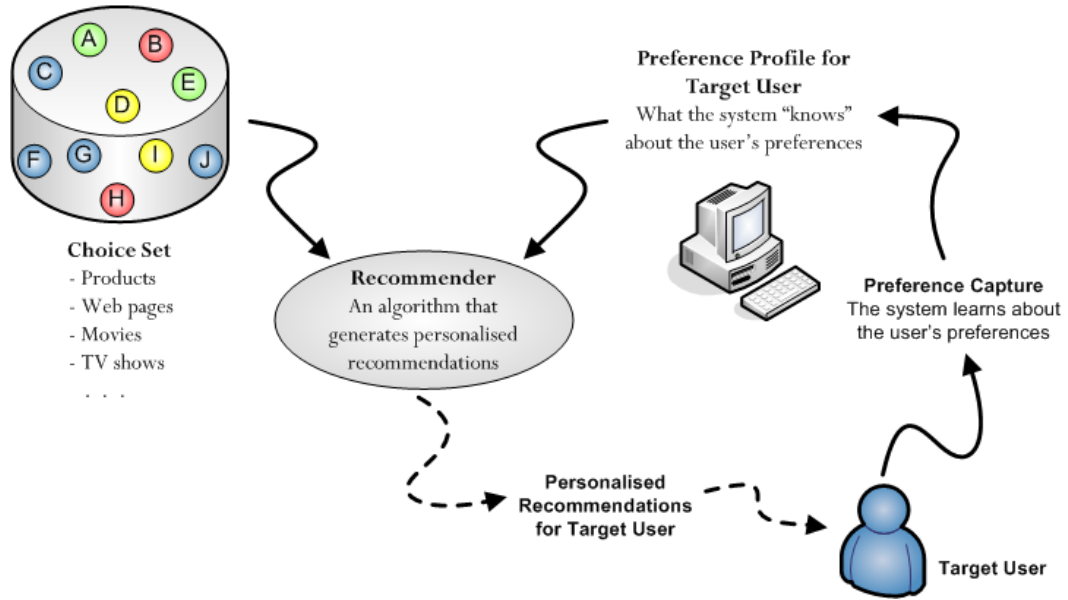


Figure 2.1 Components of a recommender system.

The main general difficulty common to all recommender systems lies in the fact that the utility function g is usually not defined on the entire $\mathcal{U} \times \mathcal{I}$ space, but only on some subset of it (the choice set), and it has to be extrapolated to the whole space. Thus, for example, in collaborative filtering systems, the utility function is defined only on the items that have been previously rated by the users.

The extrapolations from known to unknown ratings are usually done following either of the next two different approaches (Breese, Heckerman, & Kadie, 1998): 1) specifying *heuristics* that define the utility function, and empirically validating the performance of the latter, or 2) establishing *models* that estimate the utility function by optimising certain performance criteria, such as the Mean Squared Error (MSE) between known and predicted ratings. In both cases, once the unknown ratings are estimated, recommendations to a user are made by selecting the item with the highest rating, according to expression 2.1. Alternatively, the n best items can be recommended to the user.

Aside from the selected rating estimation approaches, recommender systems can be classified into the following categories, based on how recommendations are made: 1) *content-based recommender systems*, in which the user is recommended items similar to those the user preferred in the past, and, 2) *collaborative filtering systems*, in which the user is recommended items that people with similar tastes and preferences liked in the past. Due to the shortcomings proper of each of these strategies alone,

combinations of both have been investigated in the so-called *hybrid recommender systems*, empirically demonstrating their better effectiveness. Further recommendation approaches have been researched, though they cannot be considered as attempts to build long-term generalisations about the users. In this area, we may distinguish *demographic*, *knowledge-based* and *utility-based* recommender systems. In the next subsections, we present a comprehensive survey of relevant work in the field of recommender systems.

2.1.1 Heuristic-based recommender systems

According to (Breese, Heckerman, & Kadie, 1998), two main rating estimation approaches are used in recommender systems: memory-based and model-based. *Memory-based* (or *heuristic-based*) methods, such as correlation analysis and vector similarity, search the user database for user profiles that are similar to the profile of the active user that the recommendation is made for. In this type of recommender systems, it is important that the user and item databases remain in system memory during the algorithm's runtime. *Model-based* methods, such as Bayesian networks and clustering models, address the problem from a probabilistic perspective to find the best item for a given user profile, and need only keep the resulting model in memory while the algorithm is running.

Because heuristic-based approaches can make predictions based on the local neighbourhood of the active user, or can base their predictions on the similarities between items, these systems can also be classed into user-based and item-based approaches (Sarwar, Karypis, Konstan, & Riedl, 2000; Schein, Popescul, & Ungar, 2001). Sections 2.2 and 2.3 provide a survey of user-based and item-based recommender systems. For this reason, we do not enter in more details here.

2.1.2 Model-based recommender systems

In contrast to the heuristics that are based mostly on Information Retrieval (IR) methods, model-based algorithms provide item recommendation by first developing a *model* of user ratings. Algorithms in this category take a probabilistic approach and envision the recommendation process as computing the expected value of a user prediction, given his or other users' ratings on the rest of the items. The model building process is performed by different Machine Learning (ML) algorithms such as Bayesian networks (Pazzani & Billsus, 1997; Breese, Heckerman, & Kadie, 1998; Mooney, Bennett, & Roy, 1998), neural networks (Pazzani & Billsus, 1997; Breese, Heckerman, & Kadie, 1998), decision trees, clustering (Basu, Hirsh, & Cohen, 1998; Breese, Heckerman, & Kadie, 1998; Ungar & Foster, 1998), and rule-based (Sarwar, Karypis, Konstan, & Riedl, 2000) approaches. These strategies differ from IR-based approaches in that they calculate utility predictions based not on a heuristic formula,

such as a cosine similarity measure, but rather are based on a model learned from the underlying data using statistical learning techniques.

For example, based on a set of web pages that were rated as “relevant” or “irrelevant” by the user, (Pazzani & Billsus, 1997) uses the naïve Bayesian classifier (Duda, Hart, & Stork, 2001) to classify unrated web pages. More specifically, the naïve Bayesian classifier is used to estimate the following probability that page p_j belongs to a certain class C_i (e.g., relevant or irrelevant) given the set of keywords $k_{1,j}, \dots, k_{n,j}$ on that page:

$$\Pr(C_i | k_{1,j} \& \dots \& k_{n,j}).$$

Assuming that keywords are independent, the above probability is proportional to:

$$\Pr(C_i) \prod_x \Pr(k_{x,j} | C_i).$$

The keyword independence assumption does not necessarily hold in many applications. However, experimental results demonstrate that naïve Bayesian classifiers still achieve a high accuracy (Pazzani & Billsus, 1997). Furthermore, both $\Pr(C_i)$ and $\Pr(k_{x,j} | C_i)$ can be estimated from the underlying training data. Therefore, for each page p_j , the probability $\Pr(C_i | k_{1,j} \& \dots \& k_{n,j})$ is computed for each class C_i , whereupon page p_j is assigned to the class C_i having the highest probability.

Clustering models also treat recommendation as a classification problem (Basu, Hirsh, & Cohen, 1998; Ungar & Foster, 1998), by clustering similar users in the same class, estimating the probability that a particular user belongs to a particular class C_i , and thereupon computing the conditional probability of ratings.

The rule-based approach applies association rule discovery algorithms to find associations between co-purchased items, and then generates item recommendations based on the strength of the association between items (Sarwar, Karypis, Konstan, & Riedl, 2000).

Model-based approaches separate the offline tasks of creating user models from the real-time task of recommendation generation, thus improving scalability. However, this is sometimes at the cost of lower recommendation accuracy. The recommendation models proposed in this thesis follow heuristic-based strategies (see chapters 4, 5, and 6). The explicit semantic description of user preferences and item content features in our knowledge representation proposal makes it suitable to be integrated in heuristic formulas, instead of using ML techniques. Model-based strategies are not in the scope of the work presented herein, and shall therefore not be described in detail here. The reader is referred to (Adomavicius & Tuzhilin, 2005) for further reading on such methods.

2.2 Content-based recommender systems

Content-based approaches to recommendation build on the conjecture that a person likes items with features similar to those of other items he liked in the past (Terveen & Hill, 2001). Thus, the utility gain function $g(u_m, i_n)$ of item $i_n \in \mathcal{I}$ for user $u_m \in \mathcal{U}$ is estimated based on the utilities $g(u_m, i_j)$ assigned by the user u_m to items i_j that are “similar” to item i_n . For instance, in order to suggest movies to user u_m a content-based recommender system seeks to find the significant commonalities among movies user u_m has previously evaluated positively: specific genres, actors, directors, etc.

In content-based recommender systems, items are suggested according to a comparison between their content and user profiles, which contain information about the users’ tastes, interests and needs. Data structures for both of these components are created using features extracted from the content of the items. A weighting scheme is often used for providing high weights to the most discriminating features and preferences, and low weights to the less informative or characteristic ones. The profiling information can be obtained from users explicitly, for example through manual ratings, or implicitly learned from their transactional behaviour in the system over time.

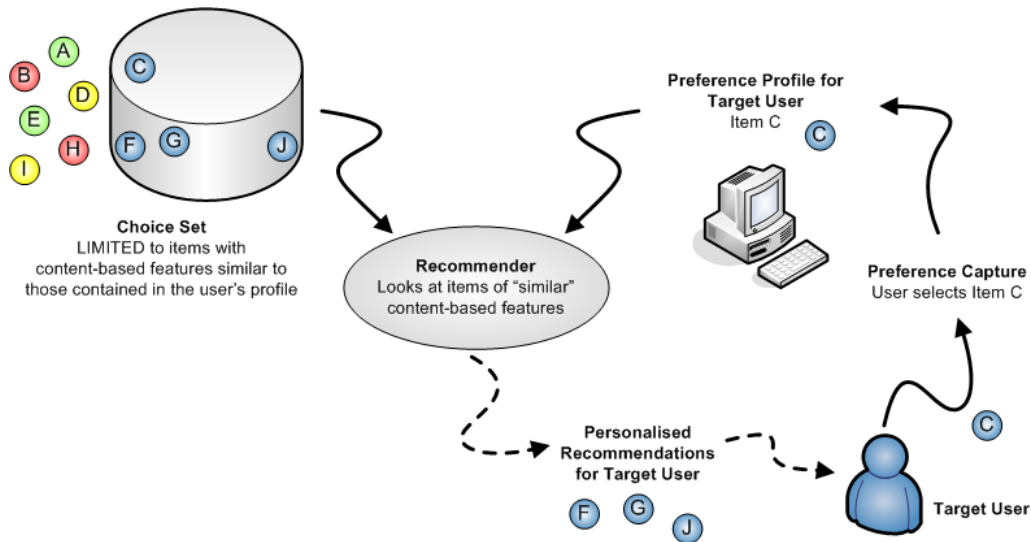


Figure 2.2 Content-based recommendations.

Figure 2.2 shows the general recommendation process followed by a content-based recommender system. Firstly, the system manually or automatically captures the target user’s preferences, building his personal profile. After this, when recommendations are to be produced, the preferences stored in this profile are compared against the features of the items stored in the system choice set, and the

items of which the features are most similar to the user's content-based preferences are retrieved and presented as recommended content to the user. Note that in this scenario only the items that share content-based features with the user profiles can be suggested, which in practice drastically reduces the set of items that can be recommended to each individual user.

More formally, and following the notation used in (Adomavicius & Tuzhilin, 2005), let $\text{Content}(i_n)$ be the content description of item $i_n \in \mathcal{I}$, i.e., the set of content features characterising i_n that are used to determine the appropriateness of the item for recommendation purposes. This description is usually represented as a vector of real numbers (weights), in which each component measures the "importance" (or "informativeness") of the corresponding feature in the item content description:

$$\text{Content}(i_n) = \mathbf{i}_n = (i_{n,1}, i_{n,2}, \dots, i_{n,K}) \in \mathbb{R}^K.$$

Since, as mentioned earlier, content-based recommender systems were mostly designed to recommend textual items, the contents of these items are usually described with *keywords*. Thus, for instance, the content-based component of the *Fab* system (Balabanovic & Shoham, 1997) represents web page contents in terms of the 128 most representative words.

Analogously, let $\text{ContentBasedUserProfile}(u_m)$ be the content-based preferences of user $u_m \in \mathcal{U}$, i.e., the weighted item content features that describe the tastes, interests and needs of the user:

$$\text{ContentBasedUserProfile}(u_m) = \mathbf{u}_m = (u_{m,1}, u_{m,2}, \dots, u_{m,K}) \in \mathbb{R}^K.$$

The utility gain of item i_n for user u_m is then computed as a score function that combines the different item description and user profile components:

$$g(u_m, i_n) = \text{score}(\text{ContentBasedUserProfile}(u_m), \text{Content}(i_n)) \in \mathcal{R}. \quad (2.2)$$

The way in which the previous expression is formulated allows distinguishing the different content-based recommendation techniques proposed in the literature. As introduced in Section 2.1, these techniques can be classified in heuristic-based and model-based approaches. The first ones calculate utility predictions based on *heuristic formulas* that are inspired mostly on information retrieval methods, such as the cosine similarity measure. The second ones, on the other hand, obtain utility predictions based on a *model* learned from the underlying data using statistical learning and machine learning models, such as Bayesian classifiers, clustering algorithms, decision trees, and artificial neural networks.

As representative examples of the above two approaches, the following main techniques are worth being mentioned:

- **Vector model.** This technique assigns the feature weights using the *term frequency/inverse document frequency* (TF-IDF) measure (Baeza-Yates & Ribeiro Neto, 1999). In a document retrieval environment, the TF-IDF measure is defined as follows.

Let N be the total number of documents that can be recommended to the users, and let N_k be the number of those documents in which the term t_k appears. Assume that $\text{freq}_{k,n}$ is the raw frequency of term t_k in the document $d_n \in \mathcal{I}$ (i.e., the number of times the term t_k is mentioned in the text of the document d_n). Then, $\text{TF}_{k,n}$, the *term frequency* (or *normalised frequency*), is given by

$$\text{TF}_{k,n} = \frac{\text{freq}_{k,n}}{\max_j \text{freq}_{j,n}}, \quad (2.3)$$

where the maximum is computed over all terms t_j which are mentioned in the text of the document d_n . If the term t_j does not appear in document d_n , then $\text{freq}_{j,n} = 0$.

The measure $\text{TF}_{k,n}$ gives more relevance to those terms that appear more times in a specific document. However, taking into account that terms which appear in many documents tend to be less useful to distinguish between a relevant document and a non relevant one, the measure $\text{TF}_{k,n}$ is usually used in combination with the so-called *inverse document frequency*, $\text{IDF}_{k,n}$:

$$\text{IDF}_{k,n} = \log \frac{N}{N_k}, \quad (2.4)$$

which assigns higher values to those terms that rarely appear in the document collection, and gives lower values to those terms that occur more frequently in the collection.

Combining equations 2.3 and 2.4, the TF-IDF weight for term t_k in document d_n is finally defined as

$$d_{n,k} = \text{TF}_{k,n} \times \text{IDF}_{k,n}. \quad (2.5)$$

With the above definitions, the vector model proposes to evaluate the utility gain of document d_n to user u_m as the correlation between the vectors $\mathbf{d}_n = \text{Content}(d_n)$ and $\mathbf{u}_m = \text{ContentBasedUserProfile}(u_m)$. This correlation can be quantified, for instance, by the cosine of the angle between the vectors:

$$g(u_m, d_n) = \cos(u_m, d_n) = \frac{u_m \cdot d_n}{\|u_m\| \times \|d_n\|} = \frac{\sum_{k=1}^K u_{m,k} d_{n,k}}{\sqrt{\sum_{k=1}^K u_{m,k}^2} \sqrt{\sum_{k=1}^K d_{n,k}^2}}. \quad (2.6)$$

- **Bayesian model.** This technique addresses the information retrieval problem within a probabilistic framework (Duda, Hart, & Stork, 2001). Its fundamental idea is the following.

Let \mathcal{I}_m be the set of items known (or initially guessed) to be relevant to user u_m . Let $\bar{\mathcal{I}}_m$ be the complement of \mathcal{I}_m , i.e., the set of items not relevant to user u_m . Let $\Pr(\mathcal{I}_m | i_n)$ be the probability that item i_n is relevant to user u_m and $\Pr(\bar{\mathcal{I}}_m | i_n)$ be the probability that item i_n is not relevant to user u_m . The utility gain on item i_n for user u_m is defined as the ratio:

$$g(u_m, i_n) = \frac{\Pr(\mathcal{I}_m | i_n)}{\Pr(\bar{\mathcal{I}}_m | i_n)}. \quad (2.7)$$

Using the Bayes' rule,

$$\Pr(\mathcal{A} | \mathcal{B}) = \frac{\Pr(\mathcal{B} | \mathcal{A}) \times \Pr(\mathcal{A})}{\Pr(\mathcal{B})},$$

equation 2.7 is transformed into:

$$g(u_m, i_n) = \frac{\Pr(i_n | \mathcal{I}_m) \times \Pr(\mathcal{I}_m)}{\Pr(i_n | \bar{\mathcal{I}}_m) \times \Pr(\bar{\mathcal{I}}_m)}. \quad (2.8)$$

The term $\Pr(i_n | \mathcal{I}_m)$ represents the probability of randomly selecting the item i_n from the set \mathcal{I}_m of items relevant to user u_m . Furthermore, $\Pr(\mathcal{I}_m)$ represents the probability that an item randomly selected from the entire item collection \mathcal{I} is relevant for user u_m . The complementary probabilities $\Pr(i_n | \bar{\mathcal{I}}_m)$ and $\Pr(\bar{\mathcal{I}}_m)$ are defined analogously.

Since $\Pr(\mathcal{I}_m)$ and $\Pr(\bar{\mathcal{I}}_m)$ are the same for all items in the collection \mathcal{I} , expression 2.8 can be rewritten as:

$$g(u_m, i_n) \sim \frac{\Pr(i_n | \mathcal{I}_m)}{\Pr(i_n | \bar{\mathcal{I}}_m)}. \quad (2.9)$$

Moreover, using the “naive” assumption that features f_k of item i_n are independent, it can be shown that the above formula is proportional to:

$$g(u_m, i_n) \sim \frac{\left(\prod_{f_k \in \text{Content}(i_n)} \Pr(f_k | \mathcal{I}_m) \right) \times \left(\prod_{f_k \notin \text{Content}(i_n)} \Pr(\bar{f}_k | \mathcal{I}_m) \right)}{\left(\prod_{f_k \in \text{Content}(i_n)} \Pr(f_k | \bar{\mathcal{I}}_m) \right) \times \left(\prod_{f_k \notin \text{Content}(i_n)} \Pr(\bar{f}_k | \bar{\mathcal{I}}_m) \right)}. \quad (2.10)$$

The term $\Pr(f_k | \mathcal{I}_m)$ represents the probability that the term f_k is present in an item randomly selected from the set \mathcal{I}_m , and $\Pr(\bar{f}_k | \mathcal{I}_m)$ represents the probability that the term f_k is not present in an item randomly selected from the set \mathcal{I}_m . The probabilities $\Pr(f_k | \bar{\mathcal{I}}_m)$ and $\Pr(\bar{f}_k | \bar{\mathcal{I}}_m)$ have meanings that are analogous to the ones just described.

While the feature independence assumption should not be applied in many applications, experimental results demonstrate that naive Bayesian classifiers still achieve a high accuracy (Pazzani & Billsus, 1997).

Finally, taking logarithms in 2.10, recalling that $\Pr(f_k | \mathcal{I}_m) + \Pr(\bar{f}_k | \mathcal{I}_m) = 1$, and ignoring factors that are constant for all items in the context of user u_m , the following expression is defined for ranking items in the Bayesian model:

$$g(u_m, i_n) \sim \sum_k u_{m,k} \times i_{n,k} \times \left(\log \frac{\Pr(f_k | \mathcal{I}_m)}{1 - \Pr(f_k | \mathcal{I}_m)} + \log \frac{1 - \Pr(f_k | \bar{\mathcal{I}}_m)}{\Pr(f_k | \bar{\mathcal{I}}_m)} \right). \quad (2.11)$$

Since the set \mathcal{I}_m is initially unknown, it is necessary to develop a method for initially computing the probabilities $\Pr(f_k | \mathcal{I}_m)$ and $\Pr(f_k | \bar{\mathcal{I}}_m)$. In (Baeza-Yates & Ribeiro Neto, 1999), some alternatives are discussed to this respect.

2.2.1 Limitations of content-based recommender systems

Content-based recommender systems have several limitations, which have been identified in the literature (Balabanovic & Shoham, 1997; Burke, 2002; Adomavicius & Tuzhilin, 2005), and are described next.

- **Restricted content analysis.** Content-based recommendations are restricted by the features that are explicitly associated with the items to be recommended. For example, content-based movie recommendations can only be based on written materials about a movie: actors' names, plot summaries, genres, etc.

The effectiveness of these techniques thus depends on the available descriptive data. Therefore, in order to have a sufficient set of features, the content should be either in a form that can be automatically parsed by a computer, or in a form in which the features can be manually extracted in an easy way. In many cases, these requirements are very difficult to fulfil. There are some domains that have an inherent difficulty for automatic feature extraction, and it is often not practical to assign features by hand. For instance, it is much harder to apply automatic feature extraction methods to multimedia data such as graphical images, video streams, and audio streams, than it is for text content.

On the other hand, if two items are represented by the same set of features, they are indistinguishable. For instance, since text documents are usually represented by their most important keywords, content-based systems cannot distinguish between a well-written text and a badly written one, if they happen to use the same terms.

- **Content over-specialisation.** Content-based recommender systems only retrieve items that score highly against a specific user profile. Tastes, interests or needs of other users that could enrich the recommendations are not taken into account. The content-based techniques cannot recommend items that are different from anything the user has seen before. Thus, for instance, a person with no experience in Spanish cuisine would never receive recommendations for even the best Spanish restaurant in town.

To overcome such limitations it may be appropriate to introduce some randomness in the recommendations, or suggest items not directly related to the user profile, for example, by considering correlated preferences of those people with similar tastes to the user (i.e., applying collaborative filtering mechanisms).

- **Portfolio effect: non diversity problem.** In certain cases, items should not be recommended if they are too similar to something the user has already seen.

To avoid this problem, the user should be presented with a diverse range of options, and not with a homogeneous set of alternatives. For example, it is not necessarily a good idea to recommend all movies by *Antonio Banderas* to a user who liked one of them in the past, or it could not be appropriate to recommend news articles describing the same event. The automatic detection of novelty and redundancy among the recommendations has already been explored and evaluated in the literature (Zhang, Callan, & Minka, 2002).

- **Cold-start: new user problem.** A user has to rate a sufficient number of items before a content-based recommender system can really grasp his preferences, and present him with reliable recommendations. A new user having none or very few ratings may not be suggested any accurate recommendations.

In Table 2.1, the recommender systems limitations identified in this section are summarised, including some general needs and solutions to address them.

		Identified problem	Needs / Possible solutions
Limitations of Content-based approaches		<i>Restricted content analysis</i>	<ul style="list-style-type: none"> • Extract content features of the items through automatic or semi-automatic processes. • Prevent the occurrence of equal content descriptions for different items. • Add additional information based not only in specific content features, but also in subjective human judgements (i.e., based on collaborative filtering features).
		<i>Content over-specialisation</i>	<ul style="list-style-type: none"> • Introduce some randomness in the recommendations. • Recommend items not directly related to the user profile, for example, considering correlated preferences of those people with similar tastes to the user (i.e., applying collaborative filtering mechanisms).
		<i>Portfolio effect: non diversity problem</i>	<ul style="list-style-type: none"> • Offer diversity in the recommendations according to related user preferences. • Filter out items not only if they are too different from the user's preferences, but also if they are too similar to something the user has seen before.
		<i>Cold-start: new user problem</i>	<ul style="list-style-type: none"> • Extend user preferences in cases where few ratings had been provided.

Table 2.1 Common limitations of content-based recommendation techniques.

2.2.2 Examples of content-based recommender systems

The roots of content-based recommendations spring from the field of Information Retrieval (Baeza-Yates & Ribeiro Neto, 1999). Because of the early and significant advances made by the IR community, and because of the importance of several text-based applications, many content-based recommender systems were focused on recommending items containing textual information. Several representative pure content-based recommender systems are presented next. Section 2.4.1 contains other content-based recommendation algorithms, but they are not described here because

they form part of hybrid recommender systems, where they are combined with collaborative filtering information.

NewsWeeder (Lang, 1995) is a Netnews filtering system that describes an article with a vector in which each component contains the number of occurrences a specific term appearing in the article. The system lets users rate their interest levels for the read articles in a 1-5 scale, and then learns their user profiles based on these ratings. Specifically, the system implements a Bayesian learning strategy based on the minimum description length principle, which takes into account a trade-off involving how to weight each term's importance, and how to decide which terms should be left out of the model for not having enough discriminating power.

Syskill & Webert (Pazzani & Billsus, 1997) is a web page recommender system designed to help users discover interesting web pages on a particular topic from a large repository. Each user has a set of profiles, one for each topic. To create a topic a user provides the system with a set of web pages considered interesting within the specific topic (see Figure 2.3). The system identifies the 128 most informative words from those web pages, which are used as Boolean features, and learns a naïve Bayesian classifier to determine the interestingness of pages. In addition, the system is enriched with background knowledge in the form of initially defined user profiles, and the use of lexical knowledge from WordNet (Miller, 1995) for feature selection.

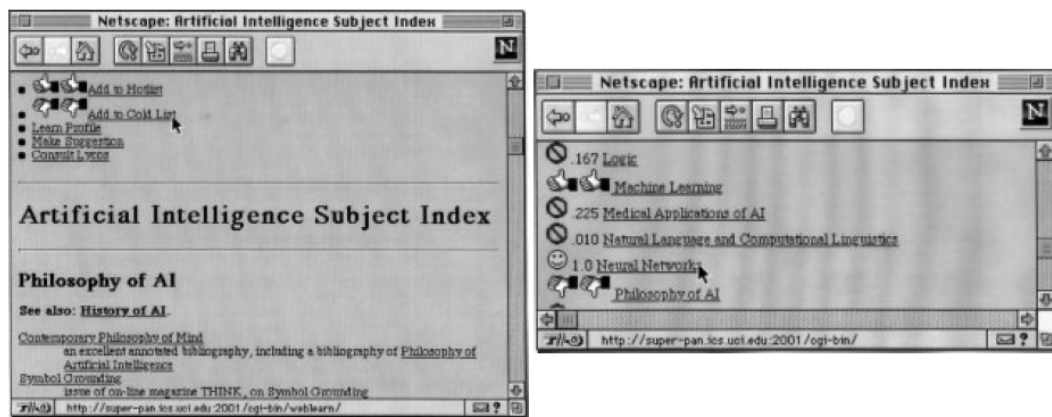


Figure 2.3 *Syskill* and *Webert* rating interface and annotation pages (Pazzani & Billsus, 1997).

InfoFinder (Krulwich & Burkey, 1997) is a content-based message recommender system that learns user information interests from sets of messages, and other on-line documents that users have classified. Specifically, the system utilises user-classified documents to build search query strings for each of their personal categories, and executes these queries to regularly recommend users those new documents that match them. In order to build such queries, *InfoFinder* extracts semantically significant topic phrases from each document using several heuristics based on visually significant features, builds a decision tree (Duda, Hart, & Stork, 2001) with the

identified phrases, and transforms the resulting decision tree into Boolean queries.

LIBRA (Mooney, Bennett, & Roy, 1998; Mooney & Roy, 2000) is a content-based book recommender system that utilises semi-structured information about books gathered from the Web. The text used to represent books is structured into fields such as author, title and subject, which are described as a set of words appearing in them. These features are used to learn a Bayesian classifier. Based on this information structure, the system has the ability to explain its recommendations by listing the features that most contribute to the highest ratings, thus favouring the readers' confidence on the system's recommendations, and providing them with insights on their own profiles.

News Dude (Billsus & Pazzani, 1999; Billsus & Pazzani, 2000) is a personal news agent that uses synthesised speech to read news stories to a user (Figure 2.4). These stories are recommended to the user according to separate models for short-term and long-term interests. User preferences are obtained by taking into account not only the user's ratings, but also the time they spent listening to the rated news readings. To determine the short-term recommendations, news stories are described in terms of TF-IDF vectors, which are compared with the cosine similarity measure, and are supplied to a learning module based on the Nearest Neighbours (NN) algorithm (Duda, Hart, & Stork, 2001). On the other hand, to establish the long-term recommendations, news stories are represented as Boolean feature vectors, where each feature indicates the presence or absence of a word, and are presented to a Bayesian learning module. According to the previous types of interests, the system also gives the user different explanations about the given recommendations.



Figure 2.4 *News Dude* user interface (Billsus & Pazzani, 1999).

2.3 Collaborative filtering systems

Collaborative filtering (CF) techniques match people with similar preferences in order to make recommendations. Unlike content-based methods, collaborative filtering systems aim to predict the utility of items for a particular user according to the items previously evaluated by other users. In other words, the utility gain function $g(u_m, i_n)$ of item $i_n \in \mathcal{I}$ for user $u_m \in \mathcal{U}$ is estimated based on the utilities $g(u_j, i_n)$ assigned to item i_n by those users u_j that are “similar” to user u_m .

The great power of the CF approaches relative to content-based ones is their “outside the box” recommendation ability (Burke, 2002), i.e., the possibility to recommend items that do not evince content features expressed in the user profiles. For example, it may occur that listeners who enjoy free jazz also enjoy avant-garde classical music, but a content-based recommender trained on the preferences of a free jazz aficionado would not be able to suggest items in the classical music realm, since none of the features (performers, instruments, repertoires) associated with items in the different categories would match. Only by looking outside the preferences of the individual such suggestions can be made.

In CF systems, users express their preferences by rating items. The ratings submitted by a user are taken as an approximate representation of his tastes, interests and needs in the application domain. These ratings are matched against ratings submitted by all other users, thereby finding the user’s set of “nearest neighbours”. Upon this, the items that were rated highly by the user’s nearest neighbours, and were not rated by the user are finally recommended. In this general setting, the way in which the user’s neighbours are determined, and the specific strategy to combine the ratings of such users characterise the different CF approaches that are commonly distinguished in the literature.

All these approaches, however, share common definitions for user profile and item description, differing from the ones used in content-based systems described in Section 2.2. Specifically, let $\text{CollaborativeUserProfile}(u_m) = \mathbf{r}_m = (r_{m,1}, r_{m,2}, \dots, r_{m,N}) \in \mathcal{R}^N$ be the collaborative profile of user u_m constituted by the set of ratings provided by the user to the N items of the system, and let $\text{Ratings}(i_n) = \mathbf{r}_n = (r_{1,n}, r_{2,n}, \dots, r_{M,n}) \in \mathcal{R}^M$ be the set of ratings $r_{m,n} \in \mathcal{R}$ assigned to item i_n by the M users registered in the system. In both of the above definitions, if user u_m has not rated item i_n , then $r_{m,n} = 0$. The utility gain of item i_n for user u_m is then computed by a score function that combines the different user profile and item description components:

$$g(u_m, i_n) = \text{score}(\text{CollaborativeUserProfile}(u_m), \text{Ratings}(i_n)) \in \mathcal{R}. \quad (2.12)$$

The way in which the previous expression is formulated allows distinguishing the different CF techniques proposed in the field. The main primary distinction is the one that classifies the techniques into *user-based* and *item-based* CF approaches. User-based CF approaches compare the active user's ratings with those of other users to identify a group of similar people in such a way that the most highly rated items of that group will be recommended to the active user. Item-based CF approaches, on the other hand, take each item of the active user's list of rated items, and recommend other items that seem to be similar to that item according to other users' ratings.

2.3.1 User-based collaborative filtering

Put in simple terms, a user-based collaborative filtering system suggests that users who chose item A will be interested in item B if other users who chose item A were also interested in item B . User-based CF techniques compare the target user's choices with those of other users to identify a group of "similar-minded" people. Once this group has been identified, those items chosen or highly rated by the group are recommended to the target user.

User-based CF algorithms make use of the entire user-item database to generate a prediction. These systems employ statistical techniques to find a set of users, known as neighbours, who have a history of agreeing with the target user (i.e., they either rate different items similarly, or they tend to rate similar types of items). Once a neighbourhood of users is formed, these systems use different algorithms to combine the preferences of neighbours to produce a prediction or top- n recommendations for the active user.

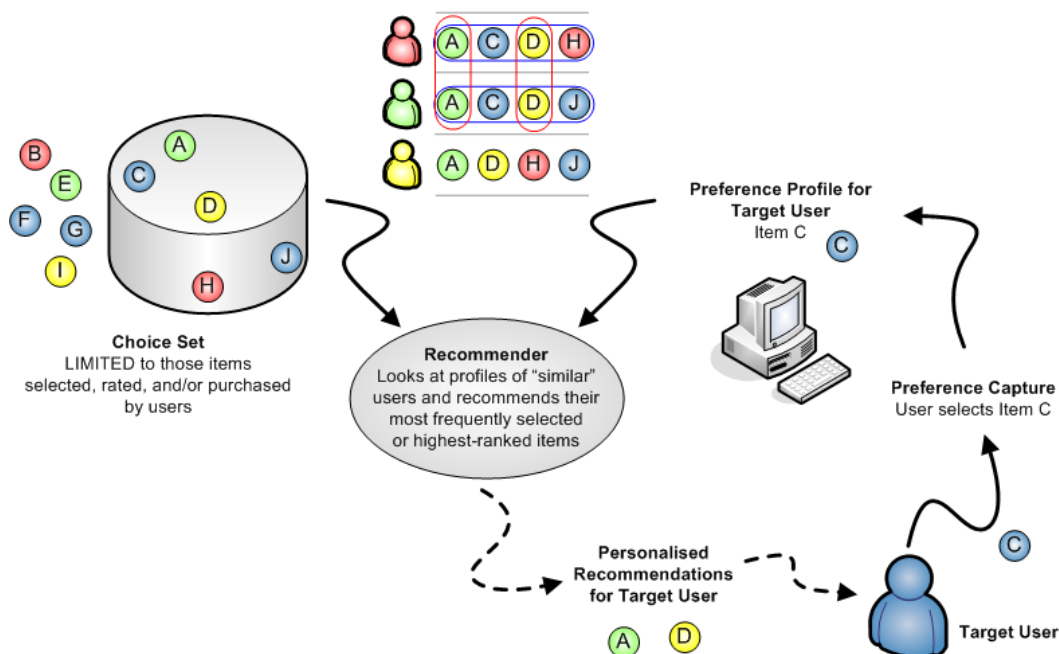


Figure 2.5 User-based collaborative filtering recommendations.

In Figure 2.5, the typical recommendation process carried out by a user-based CF system is shown. The choice set consists of the list of items that have been selected, rated and/or purchased by users. The rest of the items remain invisible to the recommender. However, note that a target user will not be recommended those items he has previously chosen. User preferences are captured by observing users' choices and/or ratings. Each choice or rating is stored in a user profile, creating histories of user in the form of action lists. To generate item suggestions, the recommendation algorithm correlates the target user's list of choices/ratings with the lists of every other user registered in the system, and selects the group of most highly correlated users (i.e., the most "similar" users). Afterwards, the system creates a list of items chosen/rated by the identified like-minded users, and ranks this list by frequency and/or by rating. The most highly items are finally recommended to the target user.

As mentioned before, user-based CF algorithms make rating predictions based on the entire set of previously rated items. Specifically and more formally, the gain utility value $g(u_m, i_n)$ of item $i_n \in \mathcal{I}$ for user $u_m \in \mathcal{U}$ is computed as an aggregate of the ratings $r_{j,n}$ of some (usually, the m^* most similar) other users u_j for the same item i_n :

$$g(u_m, i_n) = \text{aggr}_{u_j \in \hat{\mathcal{U}}_m} r_{j,n}, \quad (2.13)$$

where $\hat{\mathcal{U}}_m$ is the set of m^* most similar users to user u_m who have rated item i_n . The values of m^* can range anywhere from 1 to the number of all users registered in the system. In (Adomavicius & Tuzhilin, 2005), some examples of the above aggregation function are gathered from the literature:

$$\begin{aligned} \text{(a)} \quad g(u_m, i_n) &= \frac{1}{|\hat{\mathcal{U}}_m|} \sum_{u_j \in \hat{\mathcal{U}}_m} r_{j,n} \\ \text{(b)} \quad g(u_m, i_n) &= d \sum_{u_j \in \hat{\mathcal{U}}_m} \text{sim}(u_m, u_j) \times r_{j,n} \\ \text{(c)} \quad g(u_m, i_n) &= \bar{r}_m + d \sum_{u_j \in \hat{\mathcal{U}}_m} \text{sim}(u_m, u_j) \times (r_{j,n} - \bar{r}_j) \end{aligned} \quad (2.14)$$

where the multiplier d is a normalising factor that is usually taken as

$$d = \frac{1}{\sum_{u_j \in \hat{\mathcal{U}}_m} |\text{sim}(u_m, u_j)|},$$

and where the average rating of a user u_j , \bar{r}_j , in 2.14c is defined as

$$\bar{r}_j = \frac{1}{|\mathcal{I}_j|} \sum_{i_n \in \mathcal{I}_j} r_{j,n}, \quad \text{where } \mathcal{I}_j = \{i_n \in \mathcal{I} \mid r_{j,n} \neq 0\}.$$

In addition to the different ways in which the ratings $r_{j,n}$ are aggregated to predict the gain utility value $g(u_m, i_n)$, various approaches exist to compute the similarity $\text{sim}(u_m, u_j)$. In most of these approaches, the similarity is based on the ratings of items that both users u_m and u_j have rated. Let $\mathcal{I}_{m,j} = \{i_n \in \mathcal{I} \mid r_{m,n} \neq 0, r_{j,n} \neq 0\}$ be the set of all items co-rated by those users. The most popular approaches to compute $\text{sim}(u_m, u_j)$ are the following:

- **Cosine-based user similarity.** This measure (Breese, Heckerman, & Kadie, 1998; Sarwar, Karypis, Konstan, & Riedl, 2001) establishes the similarity between the two users u_m and u_j by computing the cosine of the angle formed by their rating vectors $\mathbf{r}_m = (r_{m,1}, \dots, r_{m,N})$ and $\mathbf{r}_j = (r_{j,1}, \dots, r_{j,N})$:

$$\text{sim}(u_m, u_j) = \cos(\mathbf{r}_m, \mathbf{r}_j) = \frac{\mathbf{r}_m \cdot \mathbf{r}_j}{\|\mathbf{r}_m\| \times \|\mathbf{r}_j\|} = \frac{\sum_{i_n \in \mathcal{I}_{m,j}} r_{m,n} \cdot r_{j,n}}{\sqrt{\sum_{i_n \in \mathcal{I}_{m,j}} r_{m,n}^2} \sqrt{\sum_{i_n \in \mathcal{I}_{m,j}} r_{j,n}^2}}. \quad (2.15)$$

- **Correlation-based user similarity.** This measure (Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994; Shardanand & Maes, 1995) establishes the similarity between the two users u_m and u_j by computing the *Pearson* correlation coefficient of their rating vectors $\mathbf{r}_m = (r_{m,1}, \dots, r_{m,N})$ and $\mathbf{r}_j = (r_{j,1}, \dots, r_{j,N})$:

$$\text{sim}(u_m, u_j) = \frac{\sum_{i_n \in \mathcal{I}_{m,j}} (r_{m,n} - \bar{r}_m) \cdot (r_{j,n} - \bar{r}_j)}{\sqrt{\sum_{i_n \in \mathcal{I}_{m,j}} (r_{m,n} - \bar{r}_m)^2} \sqrt{\sum_{i_n \in \mathcal{I}_{m,j}} (r_{j,n} - \bar{r}_j)^2}}. \quad (2.16)$$

2.3.2 Item-based collaborative filtering

An item-based collaborative filtering system suggests that a user who likes item \mathcal{A} should be recommended item \mathcal{B} if this item is found to be the most similar to item \mathcal{A} based on other users' opinions. Like user-based approaches, item-based strategies recognise patterns. However, instead of identifying patterns of similarity between user choices, they recognise patterns of similarity between the items themselves. In general terms, item-based collaborative filtering looks at each item on the target

user's list of chosen/rated items, and finds other items that seem to be “similar” to that item. The item similarity is usually defined in terms of correlations of ratings between users.

Item-based CF techniques were developed to create recommender systems with computation lower costs than those relying on user-based CF. Item-based solutions do not have to inspect databases containing millions of users in real time in order to find users with similar tastes. Instead, they can pre-score items based on their ratings and/or attributes, and then make recommendations without incurring in a high computational load. More specifically, item-based techniques first analyse the user-item matrix to identify relationships between different items, and then use these relationships to indirectly compute recommendations for users (Sarwar, Karypis, Konstan, & Riedl, 2001; Deshpande & Karypis, 2004).

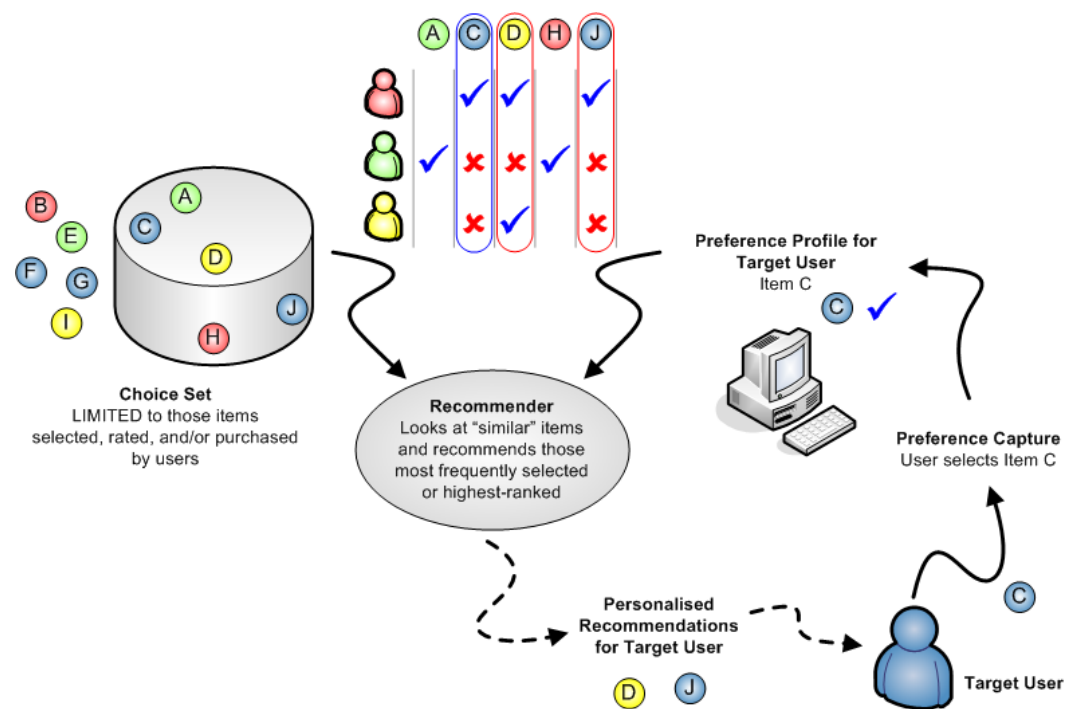


Figure 2.6 Item-based collaborative filtering recommendations.

Figure 2.6 shows the recommendation process of an item-based CF strategy. Analogously to user-based CF solutions, the choice set is limited to those items that have been selected, rated and/or purchased by users. Again, note that an item is not included in the recommendations given to the target user if it has been previously chosen by that user. User preferences are captured in the same way as in user-based CF – by observing users' choices and/or ratings, storing that information in user profiles, and creating lists of user actions. To generate recommendations, the system finds similar items to the ones listed in the target user's profile, and weights each similar item according to the ratings stored on that profile. Similar items can be

defined as those which have closely matching attributes, or which have been highly rated by users who also like the items present in the target user's profile. The items with the highest average ratings are finally recommended to the target user.

Several item-based CF approaches have been proposed to predict the gain utility function $g(u_m, i_n)$ of item i_n for user u_m . The *weighted sum* method is one of such techniques (Sarwar, Karypis, Konstan, & Riedl, 2001). This method tries to capture how the target user rates similar items. It calculates the prediction of item i_n for user u_m by computing the sum of the ratings $r_{m,j}$ given by u_m to those items i_j that are most similar to i_n . Each rating is weighted by the corresponding similarity $\text{sim}(i_n, i_j)$ between items i_n and i_j . The weighted sum is scaled by the sum of the item similarity terms in order to obtain the prediction within the predefined rating range:

$$g(u_m, i_n) = \frac{\sum_{i_j \in \mathcal{I}_n} \text{sim}(i_n, i_j) \cdot r_{m,j}}{\sum_{i_j \in \mathcal{I}_n} |\text{sim}(i_n, i_j)|}. \quad (2.17)$$

In the above expression, different ways to define the similarity between two items i_n and i_j have been proposed. Cosine-based and correlation-based approaches are two of the most popular ones, as follows. In the three formulas that follow, $\mathcal{U}_{n,j} = \{u_m \in \mathcal{U} \mid r_{m,n} \neq 0, r_{m,j} \neq 0\}$ is the set of users that have rated both items i_n and i_j .

- **Cosine-based item similarity.** Measures the similarity between two items by computing the cosine of the angle formed by their corresponding rating vectors:

$$\text{sim}(i_n, i_j) = \cos(\mathbf{r}_n, \mathbf{r}_j) = \frac{\mathbf{r}_n \cdot \mathbf{r}_j}{\|\mathbf{r}_n\| \times \|\mathbf{r}_j\|} = \frac{\sum_{u_m \in \mathcal{U}_{n,j}} r_{m,n} \cdot r_{m,j}}{\sqrt{\sum_{u_m \in \mathcal{U}_{n,j}} r_{m,n}^2} \sqrt{\sum_{u_m \in \mathcal{U}_{n,j}} r_{m,j}^2}} \quad (2.18)$$

- **Correlation-based item similarity.** Measures the item similarity computing the *Pearson* correlation coefficient of their rating vectors:

$$\text{sim}(i_n, i_j) = \frac{\sum_{u_m \in \mathcal{U}_{n,j}} (r_{m,n} - \bar{r}_n) \cdot (r_{m,j} - \bar{r}_j)}{\sqrt{\sum_{u_m \in \mathcal{U}_{n,j}} (r_{m,n} - \bar{r}_n)^2} \sqrt{\sum_{u_m \in \mathcal{U}_{n,j}} (r_{m,j} - \bar{r}_j)^2}} \quad (2.19)$$

- **Adjusted cosine item similarity.** The computation of the cosine-based item similarity (formula 2.18) has one drawback – the differences in rating scale between users are not taken into account. The adjusted cosine item similarity compensates for this by subtracting the corresponding user average rating \bar{r}_m from each co-rated pair of items:

$$\text{sim}(i_n, i_j) = \frac{\sum_{u_m \in \mathcal{U}_{n,j}} (r_{m,n} - \bar{r}_m) \cdot (r_{m,j} - \bar{r}_m)}{\sqrt{\sum_{u_m \in \mathcal{U}_{n,j}} (r_{m,n} - \bar{r}_m)^2} \sqrt{\sum_{u_m \in \mathcal{U}_{n,j}} (r_{m,j} - \bar{r}_m)^2}} \quad (2.20)$$

2.3.3 Limitations of collaborative filtering systems

Pure collaborative filtering approaches already overcome some of the weaknesses of content-based approaches. Since collaborative systems make use of other users' recommendations (ratings), they can deal with any kind of content, and recommend any items, even the ones that are dissimilar to those seen in the past. However, collaborative techniques suffer from their own limitations (Balabanovic & Shoham, 1997; Lee, 2001; Burke, 2002; Adomavicius & Tuzhilin, 2005), as described next.

- **Sparse rating problem.** In CF systems, the number of available ratings previously obtained from users is usually very small compared to the number of ratings needed to achieve reliable predictions. The estimation of new ratings from a small number of examples is thus one of the critical issues in these systems. In practice, many commercial systems, such as *Amazon.com* which recommends books, or *CDNow.com* which recommends music albums, have to evaluate very large datasets where even active users may have rated well under 1% of the existent items (Sarwar, Karypis, Konstan, & Riedl, 2001).

The success of CF recommendations depends on the availability of a critical mass of users. Collaborative systems are based on the overlap in ratings across users. They have difficulties when the space of ratings is sparse, i.e., when few users have rated the same items. There may be many items that have been rated by only a few users, and these items would be recommended very rarely, even if those few users gave them high ratings. Moreover, if the set of items changes too rapidly, old ratings will be of little value to new users, who will not be able to have their ratings compared to those of the existing users. If the set of items is large, and user interests thinly spread, then the probability of overlap with other users will be small.

Some possible solutions to the sparsity problem are:

- The use of additional non-collaborative user profile information when calculating user similarities. For example, two users could be considered similar not only if they rated the same items similarly, but also if they belong to the same demographic segment (Pazzani, 1999). Another approach is used in *GroupLens* (Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994; Konstan, Miller, Maltz, Herlocker, Gordon, & Riedl, 1997), a Netnews recommender system, where users are clustered according to existing news groups, and implicit ratings are built by measuring the time the users spend reading posts from each group.
 - The application of dimensionality reduction techniques, such as Singular Value Decomposition (SVD), to elicit underlying relations between items and users from the analysis of transitive connections (Billsus & Pazzani, 1998; Sarwar, Karypis, Konstan, & Riedl, 2000).
 - The exploitation of associative and inference rules, and related spreading activation algorithms (Crestani & Lee, 2000), to explore transitive associations among consumers and items.
- **Cold-start: new user problem.** CF strategies learn the users' preferences only from the ratings they have given. When a new user enters the system no personal ratings are available for him, and no proper recommendations can be made. Because recommendations follow from a comparison between the target user and other users, based solely on the accumulation of ratings, if few ratings are available it may become very difficult to categorise the user's interests.

Typically, two approaches are followed to address this problem:

- Use a hybrid recommendation technique that combines content-based and collaborative information (Burke, 2002).
 - Attempt to determine the best (i.e., the most informative) items for a new user to rate, using information about item popularity, item entropy, user personalisation, and combinations of the above.
- **Cold-start: new item problem.** This is the symmetric counterpart to the new user problem. CF systems only rely on users' preferences to make recommendations, and do not make use of content information of the existing items. Thus, until a new item is rated by a substantial number of users, the recommender system is not able to recommend it. Hence, a recent item that has not yet obtained many ratings cannot be easily recommended.

This problem shows up in domains such as the News, where there is a constant stream of new items, and each user only rates a few. Similarly to the new user problem, it can be addressed by hybrid recommendation approaches that consider both content-based and collaborative information during the recommendation processes.

- **Early rater problem.** In CF systems, the first person to rate an item gets little benefit from doing so. Since early ratings do not improve a user's ability to find useful matches for himself, it is necessary to provide incentives in order to encourage users to contribute their ratings, for example by taking the chance to improve their own content-based profiles as a by-product.
- **Grey sheep problem.** For the user whose tastes are unusual compared to the rest of the population, there will not be any other users who are particularly similar, leading to poor recommendations. Collaborative recommenders work best for a user who fits into a cluster with many neighbours of similar tastes. However, the techniques do not work well for the so-called "grey sheep", i.e., people who fall on the border between two cliques of users. This is also a problem for demographic systems, which attempt to categorise users according to personal characteristics.

For this kind of users, it can be beneficial to use hybrid recommendation approaches in which the content-based user profiles take more importance than collaborative aspects.

- **Portfolio effect: non diversity problem.** Since CF systems' knowledge about content is purely derived from user choices, recommendations are strongly biased toward what has been chosen (or recommended) in the past, resulting in frequent recommendations of just the most popular items. This may impoverish the potential of discovery for the end user, often failing to produce an interesting diversity of recommended content.

This fact cannot be addressed if no content-based information is available, and only users' ratings are used in the recommendation processes, so, again, the use of hybrid approaches can be a very advantageous way around this problem.

Table 2.2 gathers the CF limitations explained in this section, outlining some possible solutions.

		Identified problem	Needs / Possible solutions
Limitations of Collaborative Filtering approaches		<i>Sparsity</i>	<ul style="list-style-type: none"> • Exploit user profile information when calculating user similarities. For example, two users could be considered similar not only if they rated the same items similarly, but also if they belong to the same demographic segment. • Apply dimensionality reduction techniques, such as Singular Value Decomposition (SVD), to reduce the dimensionality of sparse ratings matrices. • Use associative and inference rules, and related spreading activation algorithms to explore transitive associations among consumers and items.
		<i>Cold-start: new user problem</i>	<ul style="list-style-type: none"> • Use a hybrid recommendation technique combining content-based and collaborative information. • Attempt to determine the best (i.e., most informative) items for a new user to rate, using information about item popularity, item entropy, user personalisation, and combinations of the above.
		<i>Cold-start: new item problem</i>	<ul style="list-style-type: none"> • Use a hybrid recommendation approach that considers both content-based and collaborative information during the recommendation processes.
		<i>Early rater problem</i>	<ul style="list-style-type: none"> • Provide incentives to encourage users to provide ratings (e.g., the possibility of improving their own content-based profiles).
		<i>Grey sheep problem</i>	<ul style="list-style-type: none"> • For this kind of users, it could be beneficial to use hybrid recommendation approaches in which the content-based user profiles take more influence than rating and collaborative aspects.
		<i>Portfolio effect: non diversity problem</i>	<ul style="list-style-type: none"> • Use a hybrid recommendation approach that exploits the content information available to confront the lack of item ratings.

Table 2.2 Common limitations of collaborative filtering techniques.

2.3.4 Examples of collaborative filtering systems

The first collaborative filtering systems reported in the literature followed a user-based approach. More recently, item-based collaborative filtering has gained momentum over the last years by virtue of computational improvements in basic prediction algorithms. For cases where the number of users is much greater than the number of items, item-based CF computational performance has been shown to be superior in practice to user-based CF (Karypis K. , 2001). Its success also extends to several commercial recommender systems, such as *Amazon.com* (Linden, Smith, & York, 2003), shown in Figure 2.7, *CDNow.com* or *MyLaunch.com*.

What do customers ultimately buy after viewing this item?

65% buy the item featured on this page: The Lord of the Rings Trilogy (Theatrical and Extended Limited Edition) DVD ~ Lord of the Rings
 ★★★★★ \$55.76

12% buy The Lord of the Rings - The Motion Picture Trilogy (Platinum Series Special Extended Edition) DVD ~ Viggo Mortensen ★★★★★
 \$63.87

Customers who bought this item also bought

The Lord of the Rings: Fellowship of the Ring - The Complete Recordings ~ Howard Shore
 The Lord of the Rings: The Two Towers - The Complete Recordings ~ Howard Shore and The London Philharmonic Orchestra
 Rome - The Complete First Season DVD ~ Kevin McKidd
 Pirates of the Caribbean - Dead Man's Chest (Two-Disc Collector's Edition) DVD ~ Orlando Bloom

Rate this item to improve your recommendations

Sign in to rate this item
 ★★★★★ I own it

Customer Reviews

Average Customer Review: ★★★★★
 Write an online review and share your thoughts with other customers.

★★★★★ A Nice Christmas Gift, November 21, 2006
 Reviewer: Dan Curtis Fan - See all my reviews
 This is a nice set. I already own the previously released sets (theatrical and extended cuts), but I'm getting two of them for friends for Christmas who don't own any of these movies yet. I guess they're not the die-hard fan that I am. It's an affordably priced set too.

Comment | Was this review helpful to you? Yes No (Report this)

Figure 2.7 Amazon.com collaborative recommendations.

Several research and commercial applications can be cited as classic examples of CF systems, as we describe next. In Section 3.5, more recent collaborative systems are described. We do not introduce them here because they are more related to social-based and ontology-based techniques.

The *GroupLens* project (Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994; Konstan, Miller, Maltz, Herlocker, Gordon, & Riedl, 1997) is one of the most referenced CF works. Based on a client/server architecture, the *GroupLens* system recommends Usenet news (Netnews) – a high volume discussion list service on the Internet (see Figure 2.8). The short lifetime of Netnews, and the underlying sparsity of the rating matrices are the two main challenges addressed by *GroupLens*. In the system, users and Netnews are clustered based on the existing news groups, and implicit ratings are computed by measuring the time the users spend reading Netnews, and using “filterbots”, i.e., programs that automatically process and rate documents.

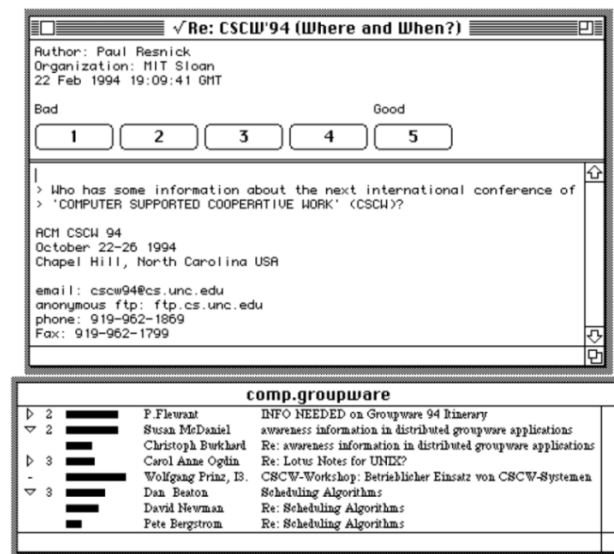


Figure 2.8 *GroupLens* rating and recommendation pages (Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994). Predicted scores are shown as bar graphs.

In (Hill, Stead, Rosenstein, & Furnas, 1995), a video recommender system is presented. Upon a client/server architecture, the system receives and sends emails to obtain user ratings and to provide video suggestions. User-based collaborative recommendations are shown to the users sorted by predicted ratings, and classified by video categories. The system also provides ranked lists with the most similar users, and gives recommendations to a group of users, instead of to individual users.

Ringo (Shardanand & Maes, 1995) is a CF system which makes recommendations of music albums and artists. One remarkable characteristic of *Ringo* is its initial user profile definition phase. When a user first enters the system, he is presented a list of 125 artists. The user rates those artists according to how much he likes listening to them. The list is formed in two parts. The first one is built on the most often rated artists, ensuring that the new user has the opportunity to rate artists which others have also rated, so that there is some commonality in people's profiles. The second one is generated upon a random selection of items from the entire database, so that all artists and albums eventually end up getting scored at some point in the initial rating phases.

Stating the CF problem as an issue of learning a binary relation between users and items, where a user is related to an item if he likes it, Nakamura and Abe (Nakamura & Abe, 1998) apply various generalisations of weighted majority prediction algorithms (Goldman & Warmuth, 1995) to provide recommendations. These methods learn weights that roughly represent the estimated affinity between users and items, and make predictions by weighted majority voting. The proposed generalisations handle the cases in which scores are not necessarily binary but many-valued, and extend the basic model to a triple user/item comparison model which is based on the idea that "a friend's friend is a friend, too".

A new combination of weighted-majority (Duda, Hart, & Stork, 2001) and memory-based algorithms is presented in (Delgado & Ishii, 1999). The authors propose to view a recommender system as a pool of independent prediction algorithms, one per each user in the system database. Each learning algorithm faces a sequence of trials with a prediction to make in each step. By defining an algorithm's individual prediction as a function of the original vote (target function) and a similarity measure between users, the authors combine both memory-based and on-line prediction. Weighted-majority is then applied for the prediction of the master algorithm for the active user, updating the weights in each trial.

Personality Diagnosis (Pennock, Horvitz, Lawrence, & Giles, 2000) is a CF method that computes the probability that a user is of the same “personality type” as another user, and, in turn, the probability that he will like non-seen items. Personality types are encoded as a vector of the user's true ratings for items in the database, and ratings are assumed to carry Gaussian noise. The probability estimations are derived by applying Bayes' rule.

2.4 Hybrid recommender systems

Hybrid recommender systems combine content-based and collaborative filtering techniques under a single framework, mitigating inherent limitations of either paradigm. Thus, hybrid recommendations are generated by taking into account both descriptive features and collaborative rating correlations.

Numerous ways for combining content-based and collaborative information are conceivable (Burke, 2002; Adomavicius & Tuzhilin, 2005). Among them, the most widely adopted is the so-called “collaborative via content” paradigm (Pazzani, 1999), where content-based profiles are built to detect similarities among users. This approach is also named meta-level hybridisation, as shown below.

Based on the taxonomy of hybridisation methods given in (Burke, 2002), hybrid recommender systems can be classified as follows:

- **Weighted hybrid recommenders.** These systems suggest items with aggregated scores that are computed by combining the results of the individual recommendation techniques to be combined. Those results are usually merged by linear combinations or vote consensus schemes.

The advantage of these methods is that the different recommendation capabilities are incorporated in the recommendation process in a straightforward way. However, they have the implicit assumption that the relative value of the different techniques is more or less uniform across the space of items – which is not always true. For example, from the discussion on the limitations of collaborative filtering given in Subsection 2.3.3, the CF approach is known to be weaker for items with a small number of ratings.

- **Switched hybrid recommenders.** These systems use some criterion to switch between recommendation techniques.

The benefit of these methods is that the suggestions can be sensitive to the strengths and weakness of the constituent recommendation techniques. However, they introduce additional complexity in the recommendation process since the switching criteria must be determined with an additional level of parameterisation.

- **Mixed hybrid recommenders.** These systems present together (e.g., combined in a single list) the suggestions given by the different recommendation techniques.

The advantage of these methods is that they directly exploit the benefits of both content-based and collaborative recommendations. However, they require ranking of items, or selection of a best suggestion, entailing the development of an item prioritisation technique.

- **Hybrid recommenders based on feature combination.** These systems merge content/collaborative suggestions by treating the collaborative information simply as additional features associated to each item, and using content-based techniques over the augmented dataset.

The benefit of these methods is that collaborative data is considered but without relying on it exclusively, thus reducing the sensitivity of the recommendations to the number of ratings.

- **Cascade hybrid recommenders.** These systems involve a staged sequential process. A first recommender produces a coarse ranking of candidates. Next, a second recommender starts from the previously filtered list as the set of candidate items, and produces a refined set of final suggestions.

The benefit of these methods is that they avoid employing the second, lower-priority technique on items that are well differentiated by the first technique, or are sufficiently poorly-rated that they will never be recommended. By doing this, cascade recommenders achieve more computationally efficient recommendations than, for example, a weighted hybrid recommender that has to apply all its techniques to all items. In addition, the cascade approach is by its nature tolerant to noise in the low-priority technique, since recommendations given by the high-priority recommender can only be refined.

- **Meta-level hybrid recommenders.** These systems combine two recommendation techniques by using the entire model generated by one (not the outputs) as the input for another.

The advantage of these methods, especially for a content-based collaborative approach, is that the learned (content-based) model is a compressed

representation of the user's interests, and the second (collaborative) recommendation step that follows can operate on this information-dense space more easily than on the initial raw data.

- **Hybrid recommenders based on feature augmentation.** These systems, similarly to cascade hybrids, involve a staged process. A first recommendation technique produces a rating or classification of each item. Afterwards, a second recommendation technique exploits the obtained information to enrich the inputs of its recommendation process. Note that these approaches are different to cascade ones, since in the latter the outputs of the first recommendation technique has no influence over the second.

The benefit of these methods is that they offer a way to improve the performance of core recommendation techniques, enriching their inputs without modifying their internal model.

2.4.1 Examples of hybrid recommender systems

Hybrid recommendation approaches have been mostly tested in experimental systems, and their success is increasingly being demonstrated in commercial applications, such as *Google* and *Yahoo!* The performance of existing search engines is often unsatisfactory in meeting users' information needs due to the enormous amount of returned information, and the fact that not all of these results are relevant or have an acceptable quality. The combination of content-based characteristics and other users' expert knowledge or search experience is a promising avenue for the implementation of a new generation of information retrieval systems. In the following, we describe several recommender systems that could be considered as first attempts to achieve the challenges of the so-called *social information retrieval*.

Fab (Balabanovic & Shoham, 1997) is a hybrid web page recommender system. In its content-based component, the text documents are represented with their most informative words, and are classified in a number of different topics. Content-based user profiles are defined according to the characteristics of the highest rated web pages for the different topics. The system uses a content-based approach in which items are rated by the user's content-based profile, and the most highly rated items are recommended to the user. This content-based approach together with a collaborative rating mechanism (Figure 2.10) allow identifying emergent Communities of Interest (CoI), whereupon social interactions between like-minded people are supported, and group as well as individual recommendations are automatically provided.

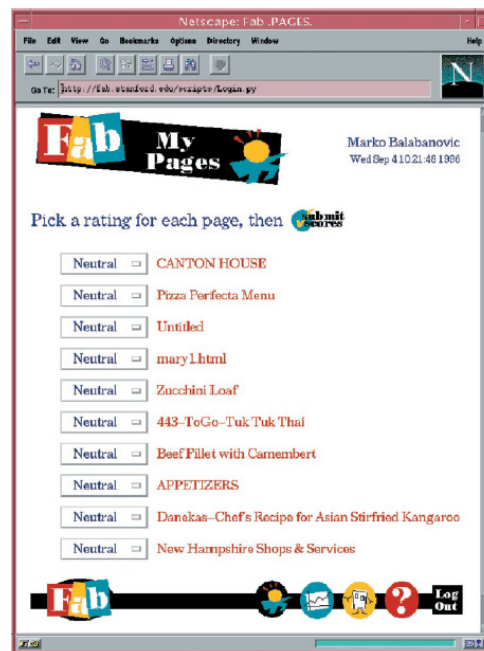


Figure 2.9 *Fab* rating page (Balabanovic & Shoham, 1997).

The work reported in (Claypool, Gokhale, Miranda, Murnikov, Netes, & Sartin, 1999) presents *P-Tango*, an online newspaper recommender system that combines content-based and collaborative filtering predictions by a weighted average. The content-based and collaborative weights are adjusted to be computed for each user and for each item according to the number of related ratings. Articles are described as a set of keywords and the newspaper sections they belong to. User profiles are divided into sections corresponding to the newspaper sections (left image in Figure 2.11). Each profile section contains a set of explicit ratings and keywords given by the user, and a list of implicit keywords which is populated by appending the keywords of the articles to which the user has given a high rating.



Figure 2.10 *P-Tango* user profile editor and on-line newspaper (Claypool, Gokhale, Miranda, Murnikov, Netes, & Sartin, 1999). The former allows the user to choose sections and keywords of interest. The latter provides a slider to enter ratings.

An alternative strategy to merge content and collaborative information is described in (Good, et al., 1999). In this case, the content information is exploited by using a set of different information filtering agents, called “filterbots” in (Konstan, Miller, Maltz, Herlocker, Gordon, & Riedl, 1997), and multiple combinations of them. The proposed types of agents are built according to several content characteristics, and information retrieval and machine learning models. The reported experiments show that using collaborative filtering to create personal combinations of a set of filterbots produces better results than either individual agents or users can produce alone.

In the context of recommending restaurants, (Pazzani, 1999) discusses two approaches to combining content-based, collaborative and demographic recommendation algorithms. One method, collaboration via content, uses collaboration among users to determine the ratings of predicted items, and uses the content-based profile only to compute similarity among users. The other method combines the results of individual algorithms seeking consensus between them. In the documented experiments, both hybrid methods obtained more precise recommendations than the individual algorithms alone.

The work published in (Tran & Cohen, 2000) presents an architecture for a hybrid recommender system, which integrates knowledge-based and collaborative filtering recommendation models as its subsystems. The authors establish conditions in the architecture for switching between the knowledge-based and the collaborative filtering styles of recommendation. These specifications take into account the current support for providing good recommendations to a particular user from the behaviour of other users, as required by the collaborative option.

In (Melville, Mooney, & Nagarajan, 2002), a framework for combining content and collaboration is presented. The framework first exploits content information of the items already rated to enrich the existing collaborative information, and second, applies a pure CF method on the enriched information. Specifically, in terms of Machine Learning, each user’s evaluations are transformed into patterns where attributes are content features of the evaluated items, and class labels are the corresponding ratings. The obtained patterns are utilised to build a naïve Bayesian classifier for each user. Once the classifiers are built, they estimate the class of all items for each user, thus generating new collaborative user profiles. These boosted collaborative user profiles are then exploited by a collaborative filtering method to make the final recommendations.

TiVo (Ali & Van Stam, 2004) is a television show recommender system (Figure 2.12). Its recommendations are provided by an item-based collaborative filtering system, and a Bayesian content-based filtering module is used to overcome the cold-start problem. The television shows are described through their genres, actors, directors and keywords. The user preferences are defined in terms of explicit user

feedback, by means of $[-3,+3]$ scale ratings, and implicit +1 value ratings, obtained from the television show records of the users.

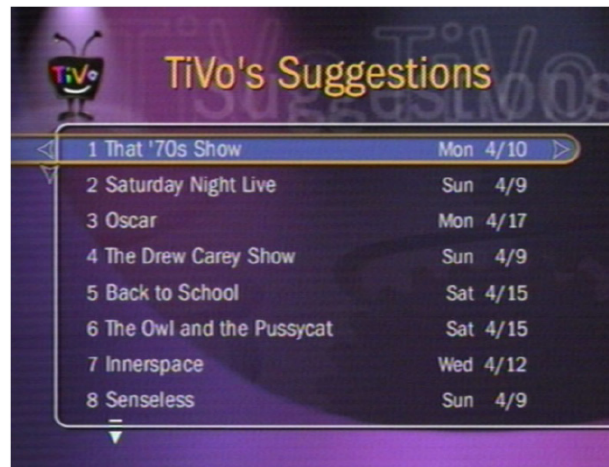


Figure 2.11 *TiVo* sorted list of recommended TV shows (Ali & Van Stam, 2004).

2.5 General limitations of recommender systems

In addition to the weaknesses specific to content-based and collaborative recommendation approaches, other limitations, common to current recommender systems in general, can be mentioned, as discussed next.

- **Poor understanding of users and items.** Most recommender systems produce ratings that are based on limited information about users and items as captured by user and item profiles, and do not take full advantage of information from users' behaviour, transactional histories, and other available data. For example, classical CF methods rely exclusively on the ratings information to make recommendations.

Since the early days of recommender systems, user and item profiles tend to be quite simple and do not utilise some of the more advanced profiling techniques. In addition to using traditional profile features such as keywords and simple demographics, more advanced profiling techniques based on data mining are progressively being used, for example finding recommendation rules, behaviour and usage patterns, etc.

- **Lack of contextual awareness.** Traditional recommenders operate on the two-dimensional Users×Items space, i.e., they make recommendations based solely on the user and item information, and do not take into consideration additional contextual information which may be crucial in some applications.

However, in many situations, the utility of a certain item to a user may largely depend on time, the people by whom the item will be consumed or shared and

under which circumstances, or the temporal purpose and changing goals of users with respect to the items. For example, a user can have significantly different preferences for the types of movies he wants to see when he is going out to a movie theatre with his girlfriend on a Saturday night, as opposed to watching a rental movie at home with his parents on a Wednesday evening.

Using multidimensional settings, the inclusion of knowledge about the user's task, goals, environment, etc. into the recommendation algorithm can lead to better recommendations (see Section 4.3 for more details).

- **Non flexible recommendations.** In general, recommendation methods are inflexible in the sense that they support a predefined, fixed way of computing recommendations. Moreover, most of them only recommend individual items to individual users, and do not deal for example with the aggregation of items and/or users. Group recommendations (Hill, Stead, Rosenstein, & Furnas, 1995) are starting to emerge as promising and very useful techniques in many real-world applications (see Section 4.4 for more details).

Therefore, the end-user cannot customise recommendation mechanisms according to his needs in real time. This problem has been identified in the literature, and the Recommendation Query Language (Adomavicius, Tuzhilin, & Zheng, 2005) has been proposed to address it, allowing the user to describe his constraints to the recommendation process by introducing SQL-like queries, as shown in Figure 2.9:

```
RECOMMEND Movie
TO User
BASED ON Rating
SHOW TOP 5
FROM MovieRecommender
WHERE Movie.genre = "comedy" AND User.city = "Madrid"
```

Figure 2.12 Example of Recommendation Query Language syntax.

In this example, the user establishes he wants to be recommended the five comedy movies that have been rated highest by people from Madrid.

- **Scalability problem.** Nearest neighbour algorithms involve a computational cost that grows exponentially with the number of users and the number of items. With millions of users and items, a typical web-based recommender system running existing algorithms suffers from serious scalability problems.

In these situations, efficient clustering techniques are thus needed to cope with this issue. A number of dimensionality reduction techniques can be applied for this purpose, such as Singular Value Decomposition (SVD), and clustering optimisation techniques, such as co-clustering.

- **Lack of support for multi-criteria ratings.** Most of the current recommender systems deal with single criterion ratings. However, it is important to be able to provide *aggregated* recommendations that suggest items based on a specific set of constraints.

In some applications, it is crucial to incorporate multi-criteria ratings into recommendation methods. Multi-criteria ratings have been extensively studied in the Operation Research community. Typical solutions to the multi-criteria optimisation problems include:

- Finding a *Pareto optimal solution*, i.e., a solution that satisfies the set of recommendation constraints, so that there is not another solution that improves the obtained satisfaction of one constrain without worsening the satisfaction of at least two of the rest constraints.
 - Taking a linear combination of multiple criteria and reducing the problem to a single-criterion problem.
 - Optimising the most important criterion and converting other criteria to constraints.
 - Consecutively optimising one criterion at a time, converting an optimal solution to constraint(s), and repeating the process for other criteria.
- **Intrusiveness.** Many recommender systems are intrusive in the sense that they require explicit feedback from the user, often to a significant degree of user involvement. Some non-intrusive methods of getting user feedback have been proposed in the field. However, non-intrusive ratings are often inaccurate and cannot fully replace explicit ratings provided by the user. Therefore, the problem of minimising intrusiveness while maintaining suitable levels of recommendation accuracy still needs to be addressed.
 - **Need for explanation.** Recommender systems should have the ability of explaining the recommendations they present to the user: causes, applied inferences on the user profile, considered constraints, etc.
 - **Lack of privacy and trustworthiness.** Recommender systems should be endowed with mechanisms that enhance the confidence and credibility levels among users, for example applying consistent privacy policies in order to protect and hide sensitive demographic and interests information of the users.
 - **Need for new evaluation methods.** Traditional accuracy measures for recommender systems do not help assess effectiveness dimensions such as “usefulness” or “quality” of the recommendations. Further research on the

definition of adequate measures and methodologies to evaluate subjective aspects of the recommendation techniques is needed.

Table 2.3 shows the general limitations of recommender systems introduced in this section. Possible solutions are also sketched.

General limitations of Recommender Systems	Identified problem	Needs / Possible solutions
	<i>Poor understanding of users and items</i>	<ul style="list-style-type: none"> • In addition to using traditional profile features such as keywords and simple demographics, more advanced profiling techniques based on data mining could be used, finding recommendation rules, behaviour and usage patterns, etc.
	<i>No incorporation of contextual information</i>	<ul style="list-style-type: none"> • Make use of multidimensional settings that enable the inclusion of knowledge about the current user's task/environment into the recommendation algorithm.
	<i>Need of flexibility</i>	<ul style="list-style-type: none"> • Provide the user mechanisms to customise the recommendations that are going to be generated, for example by expressing query inputs, constraints, etc. • Generate recommendations taking into account specific groups/segments of users and/or items.
	<i>Scalability limitations</i>	<ul style="list-style-type: none"> • Apply dimensionality reduction techniques, such as Singular Value Decomposition (SVD), and efficient clustering strategies, such as co-clustering.
	<i>No support for multi-criteria ratings</i>	<ul style="list-style-type: none"> • Adapt multi-criteria rating algorithms studied by the Operation Research community.
	<i>Non-intrusiveness</i>	<ul style="list-style-type: none"> • Explore mechanisms that minimise intrusiveness while maintain certain levels of accuracy in recommendations, for example by combining little user relevance feedback with automatic user preference learning strategies.
	<i>Need of explainability</i>	<ul style="list-style-type: none"> • Offer the ability of explaining the recommendations to the user: causes, inferences performed from the user profile, considered constraints, etc.
	<i>Trustworthiness</i>	<ul style="list-style-type: none"> • Provide mechanisms that enable to establish confidence and credibility levels among users.
	<i>Privacy</i>	<ul style="list-style-type: none"> • Provide privacy policies to protect and hide some demographic and interests information of the users.
	<i>Measuring of subjective aspects of recommendations</i>	<ul style="list-style-type: none"> • Propose novel measures and methodologies to evaluate subjective issues such as the "usefulness" and the "quality" of recommendations on items not previously presented to the users, instead of accuracy measures over already rated items.

Table 2.3 General limitations of recommendation techniques.

2.6 Evaluation of recommender systems

Recommender systems have been evaluated in many, often incomparable ways. Some evaluation metrics assess how close the ratings predicted by a recommender system are to the actual ratings provided by the users. Other evaluation strategies take into account the frequency with which a recommender system makes correct or incorrect decisions about whether an item is relevant for the user. Further, evaluation methods have been defined that quantify the ability of a recommendation algorithm to produce an ordering of the items that matches how the user would have ordered the same items according to his tastes.

An extensive and complete review of the key decisions in evaluating collaborative filtering systems is given in (Herlocker, Konstan, Terveen, & Riedl, 2004). Following that paper, the next subsections give an outline of popular metrics that have been used for the evaluation of recommender systems.

2.6.1 Accuracy metrics

Accuracy metrics have been defined for two major tasks: 1) to judge the *accuracy* of single predictions, i.e., how much predictions $p_{m,n}$ for items i_n deviate from actual ratings $r_{m,n}$, and 2) to evaluate the effectiveness of *supporting* users u_m to obtain high-quality items.

According to these tasks, accuracy metrics can be classified in the following categories:

- **Predictive accuracy metrics.** These metrics determine how close predicted ratings come to true ratings. They are particularly suited for tasks in which predictions are displayed along with the items. Two of the most popular metrics are:
 - *Mean Absolute Error (MAE).* A metric that measures the deviation of recommendations from their user-specified values. For each rating-prediction pair $\langle r_{m,n}, p_{m,n} \rangle$, this metric treats the absolute error between them (i.e., $|r_{m,n} - p_{m,n}|$) equally. The MAE is computed by first summing these absolute errors of the corresponding N rating-predictions for all the M users, and then averaging the sum by the total number of users. The lower the MAE, the more accurately the recommender predicts ratings.

$$\text{MAE} = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N |r_{m,n} - p_{m,n}|. \quad (2.21)$$

- *Root Mean Squared Error (RMSE)*. A metric that follows the same principle of MAE, but squaring the error before summing. Hence, large errors become much more pronounced than small ones.

$$\text{RMSE} = \sqrt{\frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N (r_{m,n} - p_{m,n})^2}. \quad (2.22)$$

- **Decision-support metrics.** These metrics determine how well a recommender system can make predictions of high-relevance items, i.e., items that would be highly rated (considered as “relevant”) by the user. They are particularly suitable for evaluating top- n recommendation lists: users only take care about errors for highly ranked items. Predictions errors for low-ranked items are unimportant, since users have no interest in them anyway. These metrics include classic Information Retrieval measures such as:

- *Precision*. A metric that represents the probability that an item recommended as relevant is truly relevant. It is defined as the ratio of items correctly predicted as relevant among all the items selected:

$$\text{precision} = \frac{\text{TR}}{\text{TR} + \text{FR}}, \quad (2.23)$$

where TR is the number of *true relevant* predictions, i.e., the number of items recommended as relevant that are really relevant, and FR is the number of *false relevant* predictions, i.e., the number of items recommended as relevant that are non-relevant.

- *Recall*. A metric that represents the probability that a relevant item will be recommended as relevant. It is defined as the ratio of items correctly predicted as relevant among all the items known to be relevant:

$$\text{recall} = \frac{\text{TR}}{\text{TR} + \text{FN}}, \quad (2.24)$$

where TR is the number of *true relevant* predictions, i.e., the number of items recommended as relevant that are really relevant, and FN is the number of *false non-relevant* predictions, i.e., the number of items recommended as non-relevant that are relevant.

- *F-measure*. A metric defined as the harmonic mean of the *precision* and *recall* metrics (Lewis & Gale, 1994):

$$F_{\beta} = \frac{(1 + \beta^2) \cdot \text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}, \quad (2.25)$$

where parameter $\beta \in [0,1]$ determines the relative influence of both metrics (the value $\beta = 1$ is commonly used).

- *Receiver Operating Characteristic (ROC) curve* (Swets, 1988). A metric that is used to measure the compromise between presenting the user a high number of relevant items, and recommending him a low number of non-relevant ones. They show the percentage of correctly predicted relevant items $TR/(TR+FN)$ with respect to the percentage of wrongly predicted non-relevant items $FR/(FR+TN)$. The number of correct relevant predictions can be increased at the expense of increasing the number of non-relevant predictions (and vice versa). The *Area Under the Curve (AUC)* is in fact one of the best accepted metrics by the Machine Learning community. Figure 2.13 shows three different ROC curves. The area under these curves characterises all of them, and allows measuring their different levels of goodness.

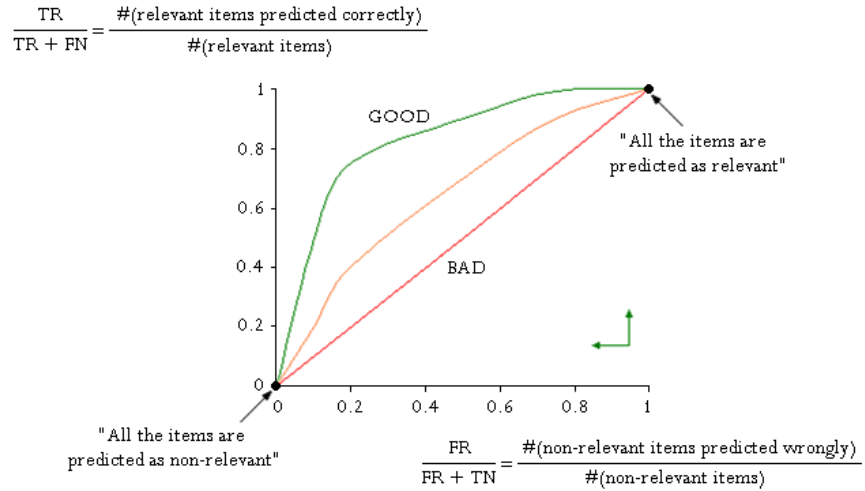


Figure 2.13 Three ROC curves with different levels of goodness according to their AUC.

2.6.2 Non-accuracy metrics

Although popular, accuracy metrics have a number of limitations. They are typically applied to test items that users chose to rate. However, items that users choose to rate are likely to constitute a skewed sample. For instance, users may rate mostly the items that they like. In other words, the empirical evaluation results typically show only how accurate the system is on most popular items, whereas the ability of the

system to properly evaluate a random item is not tested. Moreover, they often do not adequately capture “usefulness” and “quality” of recommendations. For example, a system that recommends obvious items that a user will buy or select (e.g., bread or milk in a supermarket) produces high accuracy rates; but it will not be very helpful to the user.

To overcome the previous limitations, a number of additional evaluation metrics have been proposed in the literature.

- **Coverage.** This metric is defined as the percentage of items for which a recommender system is capable of making predictions (Sarwar, Konstan, Borchers, Herlocker, Miller, & Riedl, 1998; Good, et al., 1999; Herlocker, Konstan, Borchers, & Riedl, 1999). Its value can be given in terms of a percentage on either the total number of items, or the number of items in which a user may have some interest. Systems with lower coverage may be less valuable for users, since they are limited in the decisions they are able to help with.
- **Novelty and serendipity.** These metrics measure the “non-obviousness” of the recommendations (Sarwar, Karypis, Konstan, & Riedl, 2001). To provide an example of the difference between both metrics, consider a recommender system that simply suggests books that were written by the user’s favourite writer. If the system suggests books that a user was not aware of, the recommendations will be novel, but probably not serendipitous, since the user would have likely discovered it on his own. On the other hand, a system that suggests books by new writers is more likely to provide serendipitous recommendations.
- **Learning rate.** This metric approximates how quickly an algorithm can produce good recommendations, and how well the system can help users make more effective decisions according to the data currently available. The performance of recommender systems varies depending on the amount of learning data. As the quantity of learning data increases, so should the quality of the recommendations. This issue is particularly geared to cold-start situations (Schein, Popescul, & Ungar, 2001).
- **Confidence.** This metric measures how certain the recommender system is about whether its recommendations are accurate. To help users make effective choices based on the recommendations, recommender systems should allow users to navigate along both rating prediction and confidence axes. In (Herlocker, Konstan, & Riedl, 2000) a wide range of different confidence displays are explored, to study which ones are most influential in users making the right decisions.

2.7 Summary

The success of recommender systems in overcoming the information overload in leisure, cultural and commercial applications can be already considered a reality in our days. In this chapter, we have revised a number of approaches, techniques and systems that have been proposed to provide personal recommendations for products of quite different kinds, such as books, web pages, news articles, movies, etc.

The problem of recommending items from some fixed repository has been studied extensively, and two main paradigms have emerged. Content-based recommender systems suggest items similar to those a given user liked in the past, whereas collaborative filtering systems identify users whose tastes are similar to those of the given user, and recommend items they liked.

The combination of content-based and collaborative filtering approaches, in the so-called hybrid recommender systems, has been demonstrated to be effective in limiting the impact of own weaknesses of each other. However, general limitations of recommender systems remain that have not been solved yet, and are still open research problems, such as the poor understanding and explainability of recommendations, the need for contextualisation, the lack of flexibility (e.g., query-driven or group-oriented approaches), or the usual sparsity of rating and user profile information.

Chapter 3

Semantic-based information representation and retrieval

Recommender systems generally suggest items to a user based on collaborative rating patterns, or the content similarity of these items to others already rated by the user. These approaches, however, are incapable of capturing more complex properties of, or relationships among, items at a deeper semantic level. This thesis explores the incorporation of a structured semantic layer between such spaces as a means to enable a better understanding about the underlying factors that determine whether a user is interested or not in particular items.

Few recent approaches are exploiting semantic capabilities in making recommendations. Nonetheless, a number of semantic-based techniques have been proposed long ago in the Knowledge Representation, Information Retrieval, and User Modelling fields, which can be considered as the pillars of any recommender system. For that reason, in this chapter, we revisit not only the semantic-based techniques applied to recommender systems, but also summarise relevant work in semantic-based information representation and retrieval.

More specifically, Section 3.1 describes the origins of the use of conceptual knowledge representations in information retrieval systems. Section 3.2 focuses on those semantic knowledge representations which are based on ontologies. Section 3.3 focuses deeper into the issues related to ontological engineering in the scope of the Semantic Web initiative. Finally, Sections 3.4 and 3.5 present a state-of-the-art in techniques that respectively exploit ontological structures in Information Retrieval and Recommender Systems.

3.1 Conceptual knowledge representation in Information Retrieval

Any Information Retrieval (IR) system is based on a logic representation of user information needs, and the information supplied by the objects in the search space, in such a way that the comparison between queries and potential answers takes place in the ideal model.

The various logic representations proposed in the area (Salton & McGill, 1986) respond, on the one hand, to the requirement of being efficiently processable by an IR system, and necessarily entail some information loss. This is clear, for instance, in the representation of information needs by a simple list of keywords, as is the case in currently dominant technology in both research and industry. On the other hand, an underlying goal to any IR system is that the observations performed in the ideal model correlate as frequently as possible with equivalent observations by real users. In this aim, it is natural to consider the idea of reducing the distance between the logic representation in the system and the real one in the user's mind, with regards to the formulation of queries and the understanding of documents. The problem is complex due to the involvement of diverse, difficult to capture, if not define, aspects related to human cognition, and even the definition of reality, truth and meaning. Among other reasons, this can account for the fact that the widely adopted representation in the IR field is the so-called bag of words (for text content), by which the comparison between queries and answers is mainly based on literal coincidences between queries and document passages.

Nonetheless, efforts are many that have explored the possibility to elaborate the representational level beyond the literary of character strings, towards more abstract models that approximate a conceptual representation of sought and available information, in order to enhance the response accuracy and coverage for certain types of queries. In fact, we are assisting to a renewed interest today towards the introduction of semantic capabilities in current search engines (Taylor, 2007).

The elaboration of conceptual frameworks and their introduction in IR models has wide precedents. The following quotation from a work by W. B. Croft published more than twenty years ago (Croft, 1986) could well serve today as an introduction to the topic at hand:

“The systems that have been developed, such as those based on probabilistic models of relevance (Van Rijsbergen, 1979), capture ‘domain knowledge’ purely in the statistics of occurrence of individual words (or stems) in the documents and in statistical dependencies that exist between words. We define domain knowledge to mean information about the important topics or concepts in a particular domain and how they relate to each other. The statistical approach has many advantages and can

achieve a reasonable level of effectiveness with techniques that are very efficient. However, it appears that to achieve significant improvements in retrieval effectiveness compared to current techniques, systems must be de-signed to acquire and use explicit domain knowledge.”

Starting from this point of view, in the representation proposed by Croft, the domain is modelled as a **thesaurus** of concepts, each one of which has a name, relations to other concepts, and a list of more or less ad-hoc rules to recognise the concept in a textual passage. The considered relations between concepts include synonymy, hyponymy and instantiation, meronymy and similarity. This semantic knowledge is used to expand both queries and the document indexing entries through the relations between concepts. Aware of the cost of producing domain knowledge, Croft suggests using such knowledge as an enabler of incremental improvement over purely statistic methods, in such a way that the performance of the latter is retained in the absence or incompleteness of the former. Moreover, and to further address the incompleteness problem, Croft proposes the acquisition of domain knowledge by means of dialogs with the user, which can be seen as a far precedent of current proposals in the area of “folksonomies” (Gruber, 2008).

Croft’s work is representative of a trend which, by that same period, explores the enhancement of IR systems’ performance through the enrichment of the representation of meanings by introducing an explicit conceptual abstraction. In this line, works proliferate in the eighties which investigate the use of **semantic networks** to enrich the representation of the indexing terms. See for instance (Shoval, 1981; Cohen & Kjeldsen, 1987; Rau, 1987). The introduction of a conceptual model of this kind is motivated and developed in an even more explicit way in later works, such as the ones by Agosti and Crestani (Agosti, Crestani, Gradenigo, & Mattiello, 1990; Agosti, Melucci, & Crestani, 1995; Crestani, 1997) in which semantic relations are used in relevance propagation and assisted navigation strategies, in addition to query formulation. It is also interesting, and seminal of posterior works, the explicit distinction in the latter works of three representational levels (*documents*, *words*, and *concepts*), with relations within and between such levels.

The idea of **augmenting the semantic representation of a document** beyond a set of plain words is in fact present in earlier works to those decades, such as Karen Spärck Jones’ PhD thesis itself as early as 1964 (Spärck Jones, 1964). In it, the author reflects on the flexible correspondence between words and meanings, and the role of relations between words (synonymy, antonymy, hyponymy, entailment and others) in the description of meanings. Her work considers the notion of predefined semantic primitives, consisting in essence of (domain-specific or general) concepts taken from a thesaurus (the Roget’s), which are automatically extended with emergent semantic entities, observable in the analysis of a text corpus.

Considerable research followed in which several authors have kept progressing on conceptual approaches to IR based on domain knowledge, seeking a fuller development, an improvement of results, or their application to different scenarios (the Web, etc.), with own characteristics and problems (scale, heterogeneity levels, user typology, etc.), addressing pending or new difficulties, and exploring the new opportunities brought by the evolution of technology.

One of the pursued lines in this direction is the one based on *linguistic approaches*, among which the use of resources like WordNet (Miller, 1995) is particularly representative of the use of explicit conceptual descriptions (Madala, Takenobu, & Hozumi, 1998; Vorhees, 2004). Although WordNet is a resource with domain-independence leaning, it can be said that in a way it captures generic knowledge of a wide variety of domains.

Beyond WordNet, or complementarily to its use, many works have researched the use of thesauri with a higher or lower specialisation level, to introduce enhancements in search effectiveness (Salton & Lesk, 1971; Hersh & Greenes, 1990; Paice, 1991; Hersh, Hickam, & Leone, 1992; Harbourt, Syed, Hole, & Kingsland, 1993; Jones, 1993; Yang & Chute, 1993; Järvelin, Kekäläinen, & Niemi, 2001). A thesaurus consists of a set of terms (words or titles) plus an arbitrary set of binary relations of different kinds (hierarchic, association, etc.), defined over the set of terms. In IR, thesauri represent an approximation to the representation of conceptual spaces, where the thesaural terms approximate concepts of the domain for which the thesaurus is built. One of the most common uses of thesauri in this context is the expansion of query terms, based on the mapping of query words to thesauri elements, and the extension of the latter through their relations to other terms in the thesaurus. It is common to use weights associated to the relations in the expansion, where the weights represent degrees of intensity in the relations, under different interpretations (certainty, similarity, etc.) and obtention methods (manual, statistic correlation, position in concept graphs, etc.) for such weights.

Both the use of manually created thesauri and the automatic generation of the latter have been researched in the IR field. In the first case, they are usually built by domain experts in the subjects to which the thesauri belong. There is a multitude of specialised thesauri nowadays for the access to information in fields such as health, law, economy, arts, cultural heritage, education, different scientific areas, etc., which have been used in diverse works in this line. Given the cost involved in the construction and maintenance of a thesaurus, and the importance of the unified use of this type of resource, it is usual that thesauri undergo consensus and standardisation for shared use. On its side, the automatic creation or extension of thesauri is generally based on the statistic analysis of the co-occurrence of thesaurus terms in passages from a text corpus, based on which relations between terms are inferred (Crouch, 1990; Chen & Lynch, 1992).

The studies on the effectiveness of using thesauri yield uneven results, which to much extent depend on aspects such as the quality and degree of automation of the thesaurus construction, the use or not of relevance judgments provided by experts or users, the proximity between the corpus from which a thesaurus is generated, the final search environment where it is applied, and other details such as the thesaurus term spotting techniques in text fragments. Although results have not been favourable in all cases (Hersh, Hickam, & Leone, 1992), there seems to be evidence or even consensus that it is possible to achieve improvements at least in relative terms (in some aspects, under certain conditions, etc.) by the use of thesauri (Yang & Chute, 1993).

From a very different starting point, the idea to raise IR techniques to a higher conceptual level is also explicitly present in *Latent Semantic Analysis* (LSA) techniques, widely studied and applied in diverse domains (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990). Differently from thesauri-oriented techniques, concepts emerge in LSA by means of algebraic methods, based on the frequency of words in documents of a corpus. The method has the considerable advantage of not requiring the introduction of external knowledge to the corpus whatsoever. On the other hand, the resulting concepts from LSA are intangible in that they do not have any textual or intuitive expression of their own, but they are defined by vectors that relate them to words of the initial vocabulary. Concepts are thus mathematical abstractions here, which manifest themselves in the effect obtained from them when comparing queries and documents, documents between them, or words to other words. Related to this, and through such manifestations, researchers have investigated the potential similarity between the pseudo-concepts found by LSA and the corresponding linguistic or cognitive phenomena, observable for instance in the detection of synonymy and antonymy relations, text classification, etc., by a person (Landauer, Foltz, & Laham, 1998). Although some correlation has been observed between the semantic associations obtained by LSA and human comprehension of meanings, the results in this realm cannot be considered conclusive, which limits the applicability of the product of LSA by itself to other contexts, as an explicit, reusable semantic resource or representation. Evidence has nonetheless been provided on the potential of this technique in terms of performance improvements in IR tasks (Dumais, 1994; Ledsche & Berry, 1997).

3.2 Ontologies for domain knowledge representation

After the initiatives described in the previous section, but sharing many of their premises and goals, ontology-based semantic technologies positively uphold the intensive use of domain knowledge with diverse purposes. The introduction of ontologies to move beyond the capabilities of current technologies has been an often

portrayed scenario in the area of semantic-based technologies since the late nineties (Luke, Spector, & Rager, 1996). Though Gruber's definition (Gruber, 1993) is pervasively cited, the notion of ontology has been fairly versatile in practice. In practical terms (e.g., from the standpoint of an IR researcher), ontologies are commonly handled as hierarchies of concepts with attributes and relations, which establish a terminology to define semantic networks of interrelated concepts and instances, describing domain-specific knowledge that is stored in a knowledge base (KB). In many ways, an ontology is similar to a thesaurus. However, fundamental and practical differences can be noted. While a thesaurus usually has a pre-established set of relation types, ontologies tend to be *more flexible*, typically open to arbitrary relation types, more diverse and domain-specific, which can be potentially extended anytime. In this sense, it is generally considered that a thesaurus is a particular case of ontology, the latter bearing a considerably higher expressive power.

On the other hand, ontological KBs tend to be *oriented* (though not always) *to storing large amounts of knowledge*, with a much finer level of detail than is usually envisioned in a thesaurus. We might say that, in a way (leaving aside the variety of cases, which can be considerably wide) these KBs are conceived with an intermediate perspective between a database and a thesaurus. The potential of a resource of this kind is clear although the development and maintenance costs are considerable, and proportional to the level of detail and coverage.

Compared to what is usual in thesauri, the *emphasis on formalisation is much higher* in ontologies, which seek to describe the world (or at least a domain) on the basis of a descriptive logic which axiomatises the classes, their relations, and the properties of both (symmetry, transitivity, equivalences, etc.), in suitable terms to be formally reasoned upon. This results in important advantages for the development of powerful query and inference mechanisms. In exchange, the involved problems in the approach are well-known, as the difficulty to formalise natural knowledge, even in the smaller bits, is considerable.

On the other hand, the extensive development support technologies produced since the late nineties in the semantic-based field (standards, methodologies, editors, APIs, reasoners, etc.) to facilitate the construction, exploitation and maintenance of ontologies and KBs, draw on a clear additional advantage, and the *standardisation* of infrastructures can be an important step towards the reuse of technologies and resources as the ones related to the use of thesauri in IR.

Table 3.1 shows different classifications for ontologies that have been proposed in the literature, and help better understand the differences between thesauri and ontologies. Guarino (Guarino, 1998) proposes a classification based on the generality of the ontologies. McGuinness (McGuinness, 2003) proposes a classification based on the internal structure and contents of the ontologies, and following a line where ontologies range from lightweight to heavyweight, depending on the complexity and

sophistication of the elements they contain. Finally, Gómez-Pérez et al. (Gómez-Pérez, Fernández-López, & Corcho, 2003) propose a classification that uses the type of information represented by the ontology as the main classification criterion.

Classification criterion	Categories
<p><i>According to their generality (Guarino, 1998)</i></p>	<ul style="list-style-type: none"> • Upper level ontologies. Ontologies that describe generic concepts, such as space, time and events. They are, in principle, domain independent and can be reused to construct new ontologies. • Domain ontologies. Ontologies that describe the vocabulary of a given domain, by specialising concepts provided by upper-level ontologies. • Task ontologies. Ontologies that describe the vocabulary required to perform generic tasks or activities, again by specialising concepts of upper-level ontologies. • Application ontologies. Ontologies that describe the vocabulary of a specific application, corresponding, in general, to the roles performed by entities in a given domain while performing some task or activity.
<p><i>According to the complexity of the elements they contain (McGuinness, 2003)</i></p>	<ul style="list-style-type: none"> • Controlled vocabularies. Finite list of terms. • Glossaries. Lists of terms whose meaning is described in natural language. The format of a glossary is similar to a dictionary, where terms are organised in alphabetical order, followed by their definitions. • Thesauri. Lists of terms and definitions that standardise words for indexing purposes. Besides definitions, a thesaurus also provides the hierarchical, associative, and equivalence (synonym) relationships between terms. • Informal is-a hierarchies. Hierarchies that use generalisation (type-of) relationships in an informal way, i.e., related concepts can be aggregated into a category even if they do not respect the generalisation relationship. For example, “car” and “hotel”, strictly speaking, are not “types-of-travel”, but they could appear under “travel”, in an informal “is-a” hierarchy. • Formal is-a hierarchy. Hierarchies that fully respect the generalisation relationship. • Frames. Models that include <i>classes</i> (or <i>frames</i>) that contain properties/attributes (or <i>slots</i>). Slots do not have global scope, but they apply only to the classes for which they have been defined. • Ontologies that express value restrictions. Ontologies that provide constraints to the values their class properties can assume. • Ontologies that express logical restrictions. Ontologies that allow first-order logic restrictions to be expressed.

<p><i>According to the information they represent</i> (Gómez-Pérez, Fernández-López, & Corcho, 2003)</p>	<ul style="list-style-type: none"> • Knowledge representation ontologies. They offer the modelling constructs used in frame-based representations, such as classes, subclasses, values, attributes, and axioms. • Generic and common use ontologies. They represent common-sense knowledge that can be used in different domains, typically including a vocabulary that relates classes, events, space, causality, and behaviour. • Upper ontologies. They describe general concepts. • Domain ontologies. They offer concepts that can be reused within a specific domain. • Task ontologies. They describe the vocabulary related to a task or activity. • Domain-task ontologies. They are task ontologies that can be reused in one specific domain, but not generally in similar domains. • Method ontologies. They provide definitions for concepts and relationships relevant to a process. • Application ontologies. They contain all the necessary concepts to model the application in question. They are used to specialise and extend domain or task ontologies for a specific application.
--	---

Table 3.1 Different ontology classification schemas.

3.3 Ontologies and the Semantic Web vision

Almost twenty years have passed since Tim Berners-Lee proposed the World Wide Web (WWW) project, while working at the European Organisation for Nuclear Research (CERN). At that time, CERN's staff needed to share documents located on their main computers. Berners-Lee had previously built several systems for this purpose, and with this background knowledge he conceived the WWW.

Berners-Lee wanted anyone to be able to put information on a computer, and make that information accessible to anyone else, anywhere. Without any doubt, that vision has been made reality. Nowadays, the Web provides perhaps the simplest way to share information. Literally everyone can create web pages with the help of authoring tools, and a large number of organisations disseminate data coded in web pages. As of October 2008, the indexed Web is estimated to contain over 27.6 billions of web pages⁴.

The Hypertext Markup Language (HTML) is the basic language used to encode rendering information (font size, colour, position on screen, etc.) and hyperlinks to web pages or resources on the Web (texts, multimedia files, e-mail addresses, etc.). In this scenario, computers carry out the information presentation, and the

⁴ The size of the World Wide Web, <http://www.worldwidewebsize.com/>

interpretation and identification of relevant information are delegated to human beings.

It takes great effort to evaluate, classify and select relevant information manually. Because the volume of data available on the Web is growing at an exponential rate, it is impossible for human beings to manage the whole complexity and volume of such information in complete ways. This situation puts limits to the exploitation of today's Web. It is thus natural to ask whether computers can do this job for us.

To answer this question, let us think about any of the current commercial web search engines. Suppose we want to know the history of the *Firefox* web browser, and we launch the query “firefox history”. Most of the obtained top search results would be related to tools for managing *Firefox* bookmarks, but few of them would tell us the origin and evolution of the browser. Similarly, if we introduce the term “java” in order to find information about the Indonesian island, we would obtain many links to software applications, development tools, tutorials and forums about the programming language, before obtaining the searched results. If we look for “books about García Márquez” we would find dozens of books *by* García Márquez, but probably any of them talking about the writer. Analogously, if we ask for XML standards *for* teaching (“XML teaching”), the majority of the results would refer to the teaching of XML.

In all the previous examples the limitations of the current Web reside in the fact that web pages do not contain information about themselves, i.e., about their contents, and the subjects they refer to. In other words, today's web technologies are not able to capture (i.e., formally represent) the “semantics” of the presented contents.

The Web has evolved as a medium for information exchange among people, rather than machines. As a consequence, the semantic content, i.e., the meaning of the information in a web page, is coded in such a way that it is only accessible to human beings. Figure 3.1 exemplifies this situation with a simplified version of a web page of forecast information (Catells, 2003).

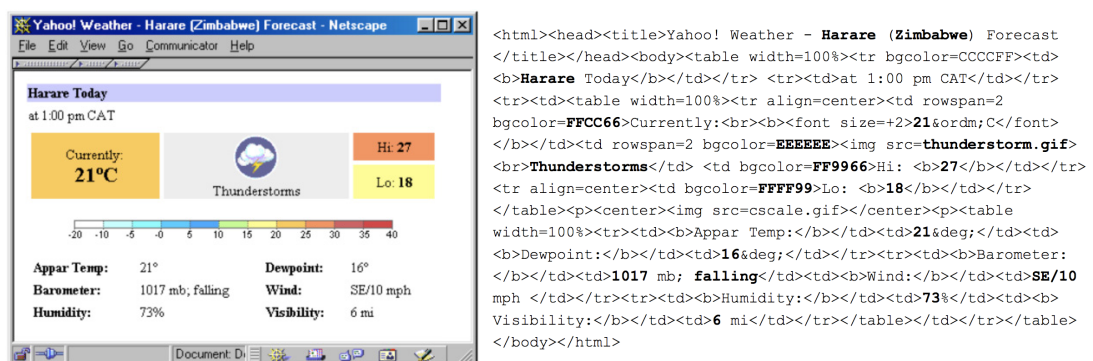


Figure 3.1 The current Web is oriented to human beings (Catells, 2003).

In the example of the figure, whilst the presentation of the data in the browser is easily interpretable by humans, it is nearly impossible to be automatically processed by a computer when the temperature, the sky conditions, and other semantics of the document have to be understood. This is due to the fact that *semantics* and style format tags are interspersed.

The word *semantics* implies meaning or, as WordNet (Miller, 1995) defines it, “of or relating to the study of meaning and changes of meaning”. For the Semantic Web, *semantic* indicates that the meaning of the data on the Web can be discovered – not just by people, but also by computers (Passin, 2004). In contrast, most meaning on the Web today is inferred by people who read web pages and hyperlinks labels, and by other people who write specialised software to work with the data. The concept *the Semantic Web* stands for a vision in which computers – software applications – as well as people can find, read, understand and use data over the World Wide Web to accomplish useful goals for users.

Of course, we already use software to accomplish things on the Web, but the distinction lies in the words *we use*. *People* surf the Web, buy things on websites, work their way through search pages, read the labels on hyperlinks, and decide which links to follow. It would be much more efficient and less-time consuming if a person could launch a process that would then proceed on its own, perhaps checking with the person from time to time as the work progresses. The business of the Semantic Web is to bring such capabilities into widespread use.

“I have always imagined the information space as something to which everyone has immediate and intuitive access, and not just to browse but to create... Machines become capable of analysing all the data on the Web – the content, links, and transactions between people and computers.

... when [the Semantic Web] does [emerge], the day-to-day mechanism of trade, bureaucracy, and our daily lives will be handled by machines talking to machines, leaving people to provide the inspiration and intuition.” (Berners-Lee, 2000)

In 2001, Berners-Lee, Hendler and Lassila published a revolutionary article in the magazine *Scientific American*, entitled “The Semantic Web: A New Form of the Web Content that is Meaningful to Computers Will Unleash a Revolution of New Possibilities” (Berners-Lee, Hendler, & Lassila, 2001). In this article, the authors describe the future scenarios in which the Semantic Web will have a fundamental role in the day-to-day life of individuals.

In one of the scenarios, Lucy needs to schedule a series of medical consultations for her mother. A series of restrictions applies to this scenario. Lucy’s tight schedule, geographical location constraints, doctor’s qualifications, and adherence to their Social Security plan. To help Lucy find a solution, there is a software agent, capable of negotiating among different parties: the doctor, Lucy’s agenda and medical service directory, among others. The point is that, although each party codes its information

in a different way, because of a semantic layer, they are able to interact and exchange data in a meaningful way. The enabling technology that will bring this scenario forward is what the authors called the Semantic Web.

The authors emphasised the important point that most the actions described in the scenarios can be achieved in today's Web, but not without considerable effort and many comes-and-goes between different websites. The promise of the Semantic Web is that it will unburden users from cumbersome and time-consuming tasks.

Confronting the implicit semantics, the chaotic growth of resources, and the absence of a clear organisation of the current Web, the Semantic Web advocates classify, provide structure, and annotate the resources with explicit semantics processable by machines. Figure 3.2 illustrates this proposal. Currently, the Web can be seen as a graph formed by nodes of a single type (HTML pages), and edges (hyperlinks) equally undifferentiated. Hence, for example, there is no distinction between a personal web page of a painter, and the website of an on-line art store, and links to the lecture pages of a professor are not differentiated with links to his publications. On the contrary, in the Semantic Web every node (resource) has assigned a specific type/class/category (professor, store, painter, book, etc.), and edges represent relations explicitly differentiated (painter – painting, professor – department, book – editorial, etc.).

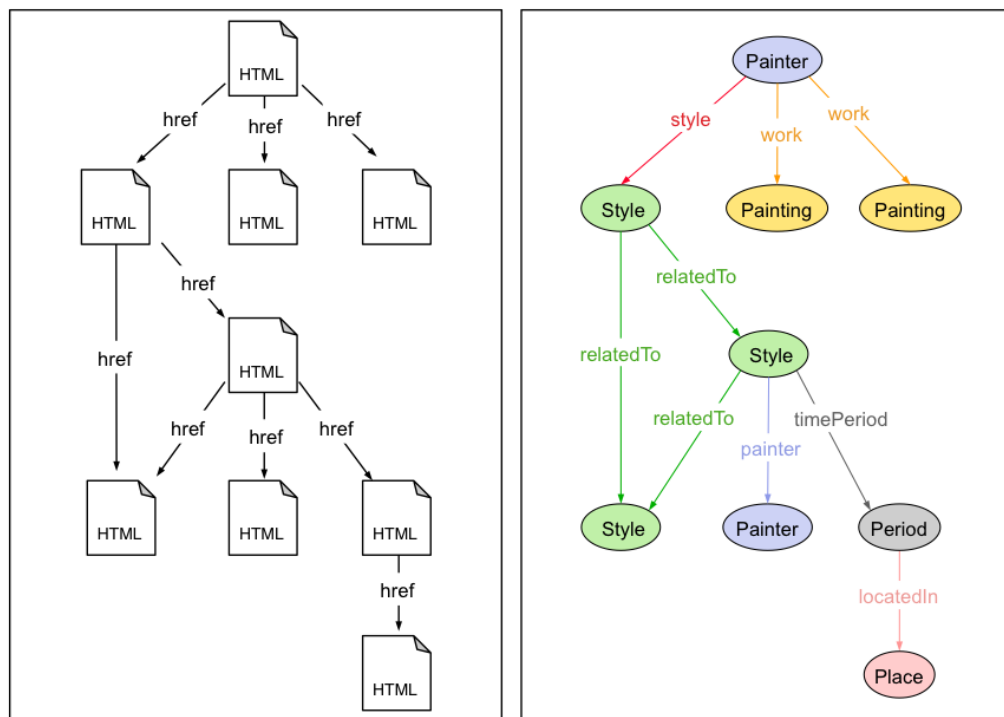


Figure 3.2 Content as it is structured in the current Web (left image) vs. the same content as it might be structured in the Semantic Web (right image).

The Semantic Web maintains the pillars that have provided the success of the current Web, such as the principles of decentralisation, portability, easy access and contribution, or the openness to growth and uses not expected beforehand. In this context, a key problem is to achieve an understanding among the parties: users, software developers, and computer programs with very different profiles.

For that purpose, the Semantic Web rescues the notion of “ontology” from the Artificial Intelligence (AI) field. Gruber defines ontology as a “*formal explicit specification of a shared conceptualisation*” (Gruber, 1993). An ontology is a hierarchy of concepts with attributes and relations that defines an agreed terminology to describe semantic networks of interrelated information units. It provides a vocabulary of classes and properties to describe a domain, emphasizing the sharing of knowledge and the consensus about its representation. For instance, an ontology about *Art* could include classes such as *Painter*, *Painting*, *Art Style*, or *Museum*, and properties (relations) like *author of a picture*, *painters belonging to an artistic style*, or *paintings shown in a museum*.

The objective is then to build a Web formed by a network of nodes typified and interconnected through properties existing in shared ontologies. Thus, for example, once an ontology about paintings had been created, a virtual museum could organise its contents defining instances of painters, paintings, art styles, etc., interrelating and making them available in the Semantic Web. The adoption of common ontologies is a key point to guarantee that all participants of the Semantic Web, providing or consuming resources, could satisfactorily work together or in an autonomous way. Continuing with the previous example, several museums could collaborate to create a great meta-museum, integrating the contents of all of them. A software agent browsing a network like that might recognise the different information units, obtain specific data or reason about complex relations. At that point, we could distinguish between a painting *painted by* an artist, and a portrait *of* an artist.

Finally, the Web not only provides access to contents, but also offers interaction and services (buying a movie, booking a flight, making a bank transfer, etc.). The Semantic Web services are an important research line in the Semantic Web, which proposes the definition of ontologies describing functionalities and procedures to describe web services: their inputs and outputs, the constraints to satisfy for their execution, the effects that they produce, or the steps to follow when dealing with complex services. These machine-processable descriptions would allow the automation, discovering, composition, and execution of services, as well as the communications among them.

3.3.1 Indexing and retrieving information

We all have had the experience of admitting defeat in the struggle to find information. At some point, everyone has been frustrated and annoyed by how hard is to locate things, especially when you are not sure what to ask for. In the Web, this situation is a daily fact, and search engines are our allies to face it.

Digital libraries, on-line stores, virtual museums, or any system that provides contents on the Internet, internally manage *index structures* that link keywords to information resources, allowing the user to very quickly find items related to his interests and goals, expressed in the form of keyword-based queries.

Focusing on searching by queries composed of words and the retrieval of documents where these words appear, an obvious approach is to scan the texts sequentially. Sequential or online text searching involves finding the occurrences of a word set pattern in a text. This strategy is appropriate when the text is small (i.e., a few megabytes), and the only choice if the collection is very volatile (i.e., undergoes modifications very frequently).

A second approach is to build data structures over the texts (called *indices*) to speed up the search. There are many approaches to build indices. Information Retrieval (IR) researchers have been (and keep) investigating indexing structures and retrieval mechanisms for the last fifty years. Excellent explanations of the most successful approaches can be found in (Baeza-Yates & Ribeiro Neto, 1999).

Figure 3.3 shows one of the simplest but most used indexing schemes called inverted files (or *inverted indices*). These structures are composed of two elements: the *vocabulary* and the *occurrences*. The vocabulary is the set of different words in all the text documents. For each of such words a list of all the text positions where the word appears is stored.

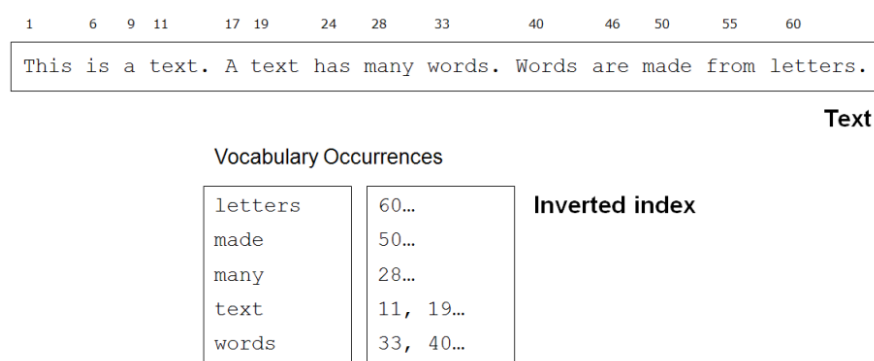


Figure 3.3 Inverted index structure (Baeza-Yates & Ribeiro Neto, 1999).

With these data structures, a search algorithm usually follows two general steps. Firstly, the words and/or word patterns present in the query are separately searched in the vocabulary. Secondly, the lists of all the found word occurrences are merged and returned.

Of course, the search process is much more complex than these two steps. Weighting mechanisms of the indexed words, compression of the data structures, morphologic/syntactic processing of the queries, or manipulation and ranking of occurrences, are some examples of difficult tasks that have to be performed. Here, we do not go into details because IR techniques are not in the scope of the thesis. Nevertheless, fundamental bibliography is suggested to the reader (Salton & McGill, 1986; Baeza-Yates & Ribeiro Neto, 1999).

In practice, it is worthwhile to build and maintain an index when the text collection is large and semi-static. Semi-static collections can be updated at reasonably regular intervals (e.g., daily), and their indices do not change very much. This is the case for most real text databases, not only dictionaries or other documental resources of slow growing pace. For instance, it is the case for web pages and journal archives.

Nowadays, the most successful techniques for medium-size databases combine online and indexed searching. The use of higher-level semantic resources, beyond index keywords, is common as well, to let users search by concepts and categories. Most systems that use conceptual information to retrieve content maintain their own concept hierarchies, and attempt to identify the recorded concepts in the documents they index. There is ample work of different scope (research and commercial) to automatically extract concepts from a document, with varying success (Alfonseca, Moreno-Sandoval, Guirao, & Ruiz-Casado, 2006). It is a more recent research goal to allow open, arbitrary definitions of vocabularies and concept sets, and to identify where these concepts are being used.

This is the essence of *metadata* information.

3.3.2 Metadata

Metadata is data about other data. For example, the ISBN number and the author's name are metadata about a book. The data types describing the data in a database also fall into the category of metadata. It is even possible to have meta-metadata, i.e., statements about the origin of a piece of metadata since after all, metadata is still data. The distinction lies in the intended use of the data, and in the subject that the metadata describes.

The origin of the notion of metadata dates back from antiquity. The Greek philosopher Aristotle provided the first known solution to organise the knowledge with his category system. He proposed that all knowledge should be structured in categories, organised under supertypes (genus) and subtypes (species). Table 3.2 shows an example of the knowledge categorisation mechanism proposed by Aristotle.

Category	Examples
<i>Substance</i>	Cat
<i>Quality</i>	The cat is black
<i>Quantity</i>	The cat is one foot long
<i>Relationship</i>	The cat is half the size of a cocker spaniel
<i>Where</i>	The cat is at home
<i>When</i>	The cat came back last night
<i>Position</i>	The cat is sitting
<i>Possession</i>	The cat has a toy
<i>Action</i>	The cat is jumping
<i>Emotion</i>	The cat likes milk

Table 3.2 Categorisation scheme of metadata about the concept “cat” proposed by Aristotle (Breitman, Casanova, & Truszkowski, 2007).

Traditional use of metadata has been often focused on specific sectorial domains, such as libraries, museums, finance, healthcare, biology, commerce, etc. The use of metadata in the context of the Semantic Web is similar, except for the fact that the environments in which the vocabularies are defined, shared and used are orders of magnitude more open and uncontrolled.

It is well known that the outstanding success of the Web is due to the freedom and decentralisation it affords. Its contents range from very sophisticated websites designed by specialists to personal web pages created by people with little computer expertise. Furthermore, in general there is little censorship or restrictions to the quality of the information in the Web. It virtually depends on the web page owners. Scientific papers cohabit in harmony with commercial websites, personal blogs, or collaborative wiki-style web pages. In this scenario of significant anarchy, it seems very hard to have a single organisation model that could prevail.

The Semantic Web should be decentralised as possible (Berners-Lee, Hendler, & Lassila, 2001). However, the fact that there should be no central control requires many compromises; the most important to provide a consistency ideal. James Hendler, one of the founding fathers of the Semantic Web, stated that, in the future, instead of a single information organisation model, there will exist a series of parallel models (Hendler, 2001), as illustrated in Figure 3.4.

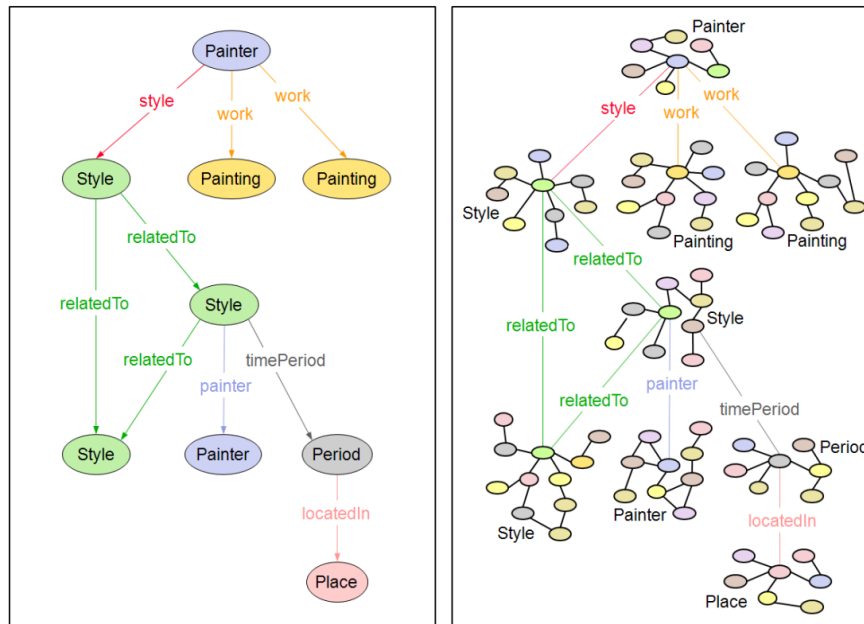


Figure 3.4 Vision of parallel semantic networks in the Semantic Web.

3.3.3 Annotations

All the meanings and information conveyed by content in unstructured form (such as text or audiovisual content) cannot in general be fully translated to a clear and formal semantic representation, for both pragmatic (cost) and intrinsic (problems for the formalisation of the world) reasons. However, it is possible to formally describe parts of the conveyed information, albeit to an incomplete extent, as metadata. For the same reason that it is generally useful to keep both parts of the information (data and metadata) in the system, it is also relevant to have a link that connects the two of them, commonly known as annotation.

Different syntactic supports and standards have been proposed for the representation of metadata and annotations. Markup languages like HTML and XML are widespread nowadays, but they have limitations in their expressiveness and shareability (Passin, 2004). Ontology-based technologies have been developed in the last few years to address and overcome some of these limitations.

In order to illustrate the subtleties in the difference between alternative metadata representation approaches and levels, we provide a simplified example of how basic metadata annotations of a web page (Figure 3.5) with the biography of the painter *Vicent Van Gogh* can be progressively structured until obtaining a formal representation of semantic concept networks.



Figure 3.5 Example of a web page about the life of the painter Vincent Van Gogh.

The most basic approach to annotate the text is the extraction of its keywords, i.e., those terms with special significance, usually occurring with salient frequency. Hence, for example, from Van Gogh’s biography, we might retrieve keywords like “painter”, “Vicent Van Gogh”, “style”, or “Post Impressionism” (see left image of Figure 3.6).

Plain keywords would hardly qualify as metadata (although as an extreme case, they could). They form part of the text itself and do not provide any additional information about the meaning (semantics) of the contents. In the given example, only having the representative keywords, we know that the document mentions a painter, and mentions Vincent Van Gogh, but we would not know that Vincent Van Gogh is a painter, or Post Impressionism is a painting style.

As a first level of metadata annotation, we may assign some of the found keywords to a number of predefined, not-interconnected named data elements or descriptors. Following the example, in the context of painters, we could have data elements such as “painting”, “style”, “name”, or “nationality”, and prior knowledge of specific painters, works, and so forth. The keywords extracted from the text could then be assigned to those categories, as shown in the right image of Figure 3.6. Thus, the example web page would mention the painter *Vincent Van Gogh*, the *Dutch* nationality, the painting style *Post Impressionism*, and the paintings *The Starry Night* and *Irises*.

At this point, we already have metadata, but we still miss semantic information. We already know that “Vicent Van Gogh” is a painter, “Dutch” is a nationality, and “The Starry Night” is a painting. However, we are not able to state that *Vicent Van Gogh* was *Dutch* and painted *The Starry Night*. The metadata is not structured, and no description about how categories are related is provided.

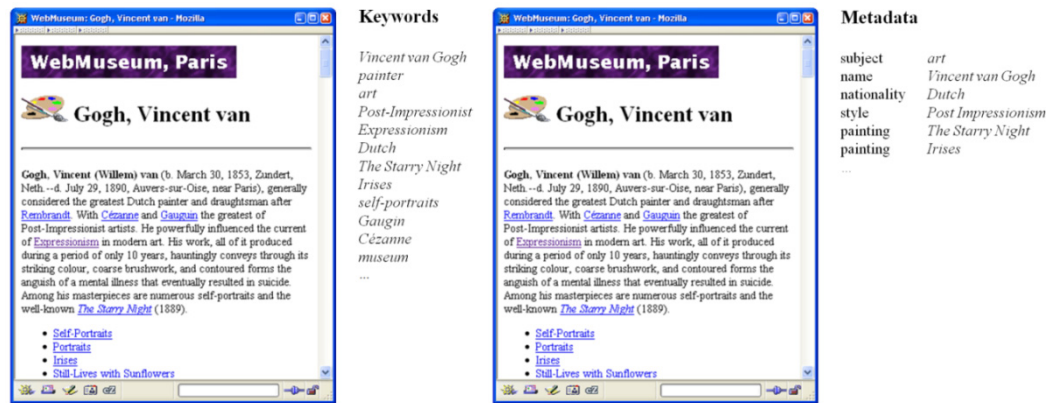


Figure 3.6 Example of a first annotation level in a web page, where content keywords are identified and assigned to a set of raw categories.

The previous approach to annotation is the simplest one, and can be supported for example by simple HTML meta-tags or document-oriented XML tags, as shown in Figure 3.7.

```
<html lang="en">

<head> ... </head>

<body>
  ...
  <p> <b><name>Gogh, Vincent (Willem) van</name></b> (<birth> March 30,
    1853, Zundert, Neth.</birth> -- <death>July 29, 1890, Auvers-sur-Oise
    </death>), generally considered the greatest <nationality>Dutch
    </nationality> <role>painter</role> and draughtsman after
    <a href="/auth/rembrandt/"> <painter>Rembrandt</painter></a>.

    <p> Among his masterpieces are numerous self-portraits and the well-known
    <a href="/starry-night/"><painting>The Starry Night</painting></a> (1889).
    ...
</body>
</html>
```

```
<html lang="en">

<head>
  <meta http-equiv="Content-Type"
    content="text/html; charset=iso-8859-1">
  <meta name="subject" content="Art">
  <meta name="name" content="Vincent van Gogh">
  <meta name="nationality" content="Dutch">
  <meta name="style" content="Post Impressionism">
  <meta name="painting" content="The Starry Night">
  <meta name="painting" content="Irises">
  <title> WebMuseum: Gogh, Vincent van </title>
  <link rel="stylesheet" type="text/css" href="/wm/home.css">
</head>


<body> ... </body>

</html>
```

Figure 3.7 Simple annotation of content keywords using HTML meta-tags and document-oriented XML tags.

An easy way of adding structure to metadata is to declare category properties (attributes) whose values would be specified for each of the different instances. Thus, as shown in Figure 3.8, the category *Painter* might be assigned properties such as

“name”, “nationality”, “style” or “works”, and the annotation “Vicent Van Gogh” would have the type *Painter*, the name *Vicent Van Gogh*, the *Dutch* nationality, the style *Post Impressionism*, and several works like *The Starry Night* or *Irises*. It is very important to note that the values of all the previous properties are strings, and do not reference to other annotations. Indeed, the next step in the annotation process should be the incorporation of relations between annotation entities.



Structured metadata

```

vincentVanGogh
type      Painter
name      Vincent van Gogh
nationality Dutch
style     Post Impressionism
works     The Starry Night
           Irises
           ...

starryNight
title     The Starry Night
date      1889
...

postImpressionism
type      PaintingStyle
name      Post Impressionism
painters  Vincent van Gogh
           Gauguin
           ...

```

Figure 3.8 Example of basic metadata structure where each semantic annotation contains string-valued properties.

Such structure can be supported by the structure-oriented side of XML. Figure 3.9 shows how XML could be used to structure the metadata of the annotations associated to Vicent Van Gogh’s biography.

```

<Painter>
  <name>Gogh, Vincent (Willem) van</name>
  <birth>March 30, 1853, Zundert, Neth.</birth>
  <death>July 29, 1890, Auvers-sur-Oise</death>
  <nationality>Dutch</nationality>
  <role>painter</role>
  <style>Post Impressionism</style>
  <painting>The Starry Night</painting>
  <painting>Irises</painting>
  ...
</Painter>

<Painting>
  <title>The Starry Night</title>
  <date>1889</date>
  ...
</Painting>

<PaintingStyle>
  <name>Post Impressionism</name>
  <painter>Vincent van Gogh</painter>
  <painter>Gauguin</painter>
  ...
</PaintingStyle>

```

Figure 3.9 XML-based structured annotations.

The incorporation of relations between entities is a natural step to maintain rich descriptions of the semantics underlying the web contents. Doing that, the values of the properties would not have to be just strings, but could also refer to instances of other categories in the document. Thus, for example, the property “works” of a *Painter* would target a specific instance of the *Painting* category. Figure 3.10 depicts this idea. The instance “Vicent Van Gogh” of type *Painter* contains several properties “work” linking to the painter’s works (e.g., “Starry Night”), which belong to the *Painting* category.

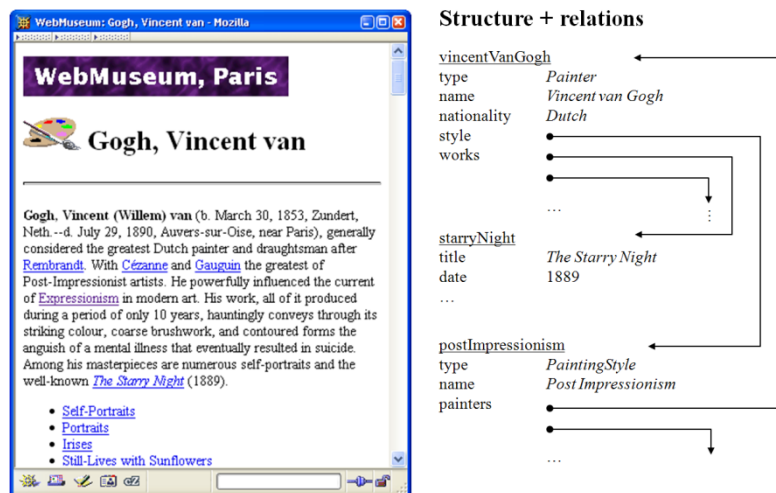


Figure 3.10 Example of structured and interrelated metadata.

So far, for the purpose of annotation needs, we have exemplified how metadata can be structured through the declaration of arbitrary categories and properties. However, this solution does not take into consideration other requirements, such as portability and modularity. For instance, we may want to reference semantic concepts of external resources, or allow for extensions of the available structures. Such needs can be addressed by an ontology-oriented approach. Specifically, structures can be transformed into ontology classes with well-defined syntax and hierarchical links, and properties would be defined as class attributes with specific type and cardinality restrictions. The resulting hierarchy of interrelated concepts, which provides a vocabulary to describe a domain and maintains a consensus about its representation, is basically an *ontology*. Figure 3.11 depicts a simplified representation of the ontology associated to the example web page. Note how formal syntax is used to declare classes (“class” reserved word), class inheritance (“extends” reserved word), primitive data types (e.g., “String”), or cardinality constraints (such as “multiple” attribute).

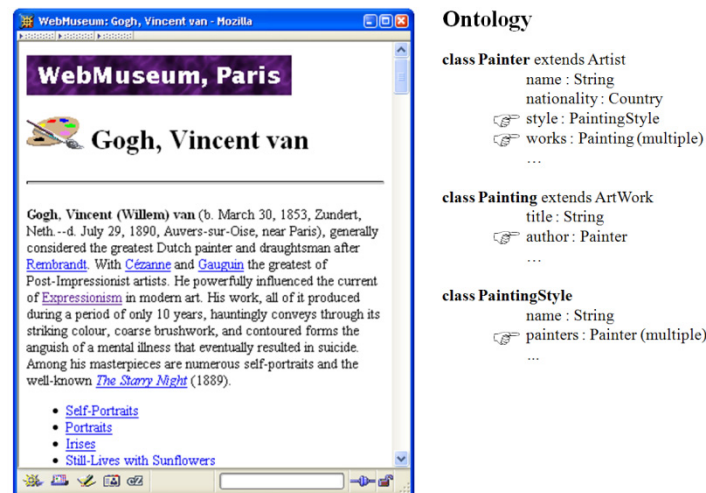


Figure 3.11 Example of metadata represented in the form of an ontology.

As it is explained in the next subsections, several XML-based *languages for the description of ontologies* have been proposed in the last few years. These languages are the common pillars of Semantic Web applications.

3.3.4 Ontology description languages

Ontology description languages have received considerable attention since the end of the nineties, boosted by the emergence of the Semantic Web. The diagram in Figure 3.12 shows the layered technologies of the Semantic Web, where the layers from RDF Schema downwards are standardized by the World Wide Web Consortium (W3C).

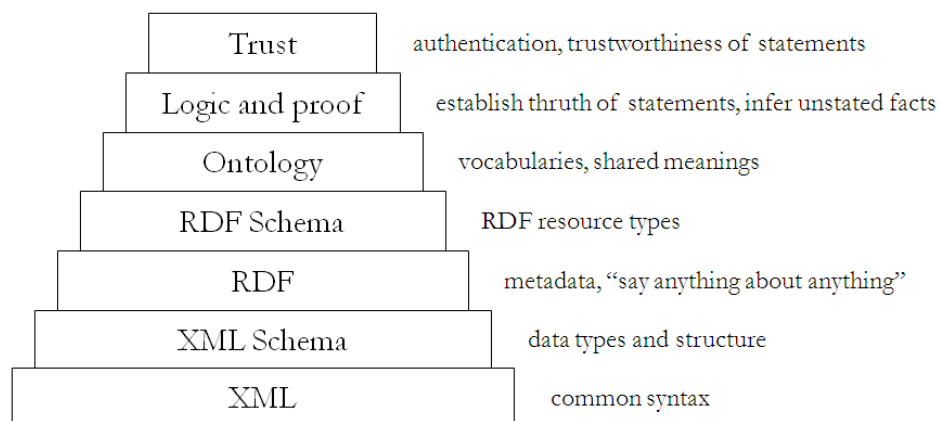


Figure 3.12 The layered technologies of the Semantic Web (Passin, 2004).

In this layered model, the relationships among resource and ontology description languages are shown. Each layer is seen as building on the layer below. At the base, most data is expected to be created in XML formats. Each layer is progressively more specialised and also tends to be more complex than the layers below it. A lower

layer does not depend on any higher layers. Thus, the layers can be developed and made operational relatively independently.

- **XML** (eXtensible Markup Language) is the language framework that, since the end of the nineties, has been used to define most new languages that are used to interchange data over the Web.
- **XML Schema** is a language to define the structure of specific XML-based vocabularies.
- **RDF** (Resource Description Languages) is a flexible language with a graph-based data model supporting the definition of ontological metadata in the form of arbitrary resources interlinked by semantic relations.
- **RDF Schema** (RDFS) is a complement of RDF conceived to type resources with classes, associate relations with classes, and build class hierarchies.
- **Ontology** is a layer containing languages for the definition of vocabularies and conditions on the usage of words and terms in the context of a specific vocabulary. OWL (Web Ontology Language) is one of such languages.
- **Logic and proof** is a layer where logic reasoning is used to check the consistency and correctness of datasets, and to infer new knowledge that is not explicitly stated but is required by, or consistent with, a known set of data.
- **Trust** is a layer to provide authentication of identity and evidence of the trustworthiness of data, services, and agents.

The reader should realise that the above diagram is the one upheld by the W3C view. There are potential alternatives for some of the layers. Among others, alternative schemas exist for XML documents, besides a large number of alternative efforts to develop ontology systems.

In the following, we very briefly describe XML, XML Schema, RDF and OWL, giving some examples to highlight the relationships and extensions among them. The reason for explaining these languages is two-fold. Firstly, after being released as W3C recommendations, these languages are being extensively exploited by scientific and some commercial semantic applications (Benjamins, et al., 2008). Secondly, the ontology-based recommendation models presented in this dissertation use these languages. The explanation of the latter shall introduce here some concepts that will be needed in later chapters.

XML represents a first approach to a web-based ontology support. XML allows structuring data and documents in the form of trees of tags with attributes, while XML Schema is used to provide the specification of those trees, and the definition of primitive and extended data types.

Since the advent of XML in 1998, a number of standards have been defined for modelling information in very specific domains, such as finance (Coates, 2001) (XBRL, RIXML, RbXML, ebXML, etc.), news (e.g., NewsML, XMLNews, PRISM), teaching (SCORM, IEEE LOM, etc.), or healthcare (Dudek, 2001) (NLM Medline, SCIPHOX, CDA, etc.), among many other fields. XML is a first step to support an explicit data representation, and well-defined structure of web contents, separated from (or embedded in) their presentation in HTML. However, the representational support procured by XML is mostly syntactic, with limited semantic expressiveness. The XML data model consists of a tree structure in which there is no distinction between objects and relations, nor is a proper support provided for class hierarchies.

The first version of **RDF** was published in 1999. Being a language for the definition of ontologies and metadata in the Web, RDF is today one of the most popular and widespread standard in the Semantic Web community. The basic unit of representation in RDF is the “triple” or sentence, which consists of two nodes (subject and object) linked by a directed edge (predicate). The nodes represent resources, and the edge represents a property that relates the two nodes. For example, a sentence could describe the fact that the author (predicate) of “Starry Night” painting (subject) was Vicent Van Gogh (object), as shown in Figure 3.13. Linking several of these triples, semantic graphs or networks are built.

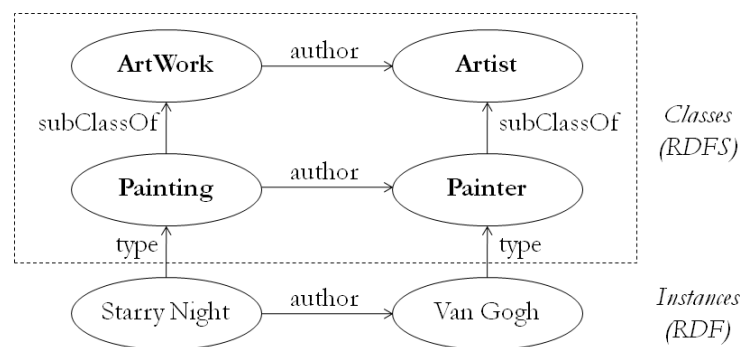


Figure 3.13 Example of RDF(S) graph.

RDF Schema (RDFS) is used to declare the class hierarchies, and the allowed properties and relations of the available resources (see Figure 3.13). In RDF, classes, relations, and the sentences themselves are also resources, so they can be reached as part of the graph. Several syntactic ways to formulate RDF have been proposed, but perhaps the most widely adopted is the XML-based. This is the reason for usually considering RDF as an extension of XML. Figure 3.14 shows a simplified example of RDF(S) syntax.

<pre> <Painter about="vangogh" name="Vicent van Gogh" birth="1853" death="1890" nationality="Dutch"> ... </Painter> <Painting about="starrynight" ...> <author resource="vangogh"> ... </Painting> </pre>	Instances

<pre> <Class about="Painter"> <subClassOf resource="Artist"/> </Class> <Class about="Painting"> <subClassOf resource="ArtWork"/> </Class> <Class about="Artist"/> <Class about="ArtWork"/> </pre>	Classes

```

<Property about="author">
    <domain resource="ArtWork"/>
    <range resource="Artist"/>
</Property>

```

Figure 3.14 Example of RDF(S) syntax.

RDF and RDFS are accompanied by the definition of query languages similar to the well-know SQL for database management. These languages support complex queries on an RDF graph using a simple declarative syntax. Failing to reach agreement on a single standard, various particular initiatives have been consolidating as de-facto RDF query languages, such as SPARQL, an W3C recommendation, RDF Query Language (RDQL) from Hewlett-Packard company, RDF Schema Query Language (Karvounarakis, Alexaki, Christophides, Plexousakis, & Scholl, 2002) (RQL), or Sesame RDF Query Language (SeRQL), developed by the Dutch company Administrator. As a representative example, the SPARQL query given in Figure 3.15 would return all European painters.

```

PREFIX ns:<http://example.com/artOntology#>
SELECT ?painterName ?country
WHERE {
    ?x ns:name ?painterName ;
        ns:nationality ?y .
    ?y ns:countryName ?country ;
        ns:isInContinent ns:Europe .
}

```

Figure 3.15 Example of RDQL query.

After RDF and RDFS, two ontology description language proposals were put forward: OIL (Ontology Inference Language), which was developed in Europe, and DAML (DARPA Agent Markup Language), which was developed in the USA. These two languages were very similar, and they finally merged into a single one: DAML+OIL. From this union, aiming to leverage the advantages of DAML+OIL and improve its limitations, a new language called **OWL** (Web Ontology Language) was defined. OWL can be formulated in RDF format, so it is usually considered as an extension of the latter. OWL includes all the expressive capabilities of RDF(S) and extends them with the possibility of using logical expressions. OWL allows, for example, the definition of classes by the declaration of constraints over their properties (e.g., the class of paintings from Spanish painters), by the combination of several classes using Boolean and Set operators (e.g., the class of Spanish and Post-Impressionist painters), or by the enumeration of the instances belonging to the classes. Further, OWL allows assigning features to the semantic properties, such as cardinality, transitivity or inverse relations. A few examples are shown in Figure 3.16. Besides RDF(S) and OWL, which can be considered the most widespread ontology description languages, a number of other interesting initiatives have been developed, such as OCML or WebODE.

```
...
<owl:Class rdf:ID="SpanishPostImpressionistPainter">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:resource="#SpanishArtist"/>
    <owl:Class rdf:resource="#PostImpressionistPainter"/>
  </owl:intersectionOf>
</owl:Class>
...
<owl:Restriction>
  <owl:onProperty rdf:resource="#author"/>
  <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:Cardinality>
</owl:Restriction>
...
<owl:Class rdf:ID="Continent">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#Europe">
    <owl:Thing rdf:about="#Africa">
    <owl:Thing rdf:about="#Asia">
    <owl:Thing rdf:about="#NorthAmerica">
    <owl:Thing rdf:about="#SouthAmerica">
    <owl:Thing rdf:about="#Australia">
    <owl:Thing rdf:about="#Atarctica">
  </owl:oneOf>
</owl:Class>
...
```

Figure 3.16 Example of OWL expressivity capabilities.

3.4 Semantics in Information Retrieval

The most common way in which *semantic information retrieval* has been understood and addressed from the area of semantic-oriented technologies, especially in their beginnings in the late nineties, consists of the construction of a query engine that receives requests in an ontology query language (such as SPARQL today), executes them on a KB, and returns tuples of ontology values from the ontology which satisfy the conditions in the query. These techniques use thus Boolean search models, based on an ideal vision of the information space, as consisting of formal ontological knowledge units, devoid of ambiguity or redundancy. Under such perspective, the IR problem is reduced to a data retrieval task. A knowledge unit is an either correct or incorrect answer to a given information request, whereby the search results are assumed to be 100% precise, and there is no notion of approximate answer to an information need. This view can be framed as an issue of Question Answering (QA), a long researched topic in Natural Language Processing (Burger, et al., 2001), also converging to the IR field (Vorhess, 2001).

The so-called semantic portals (Maedche, Staab, Stojanovic, Studer, & Sure, 2003; Castells, Foncillas, Lara, Rico, & Alonso, 2004; Contreras, et al., 2004) are a good example of this approach. These portals typically provide simple search functionalities which may be better classed in the spectrum of *semantic data retrieval*, rather than semantic information retrieval. Searches return ontology instances or values, rather than documents, and no ranking method is usually provided. In some systems, links to documents that reference the instances are added in the user interface, next to each returned instance in the query answer (Contreras, et al., 2004), but neither the instances nor the documents are sorted by relevance. Maedche et al. do provide a criterion for query result ranking in the SEAL Portal (Maedche, Staab, Stojanovic, Studer, & Sure, 2003), but the principles on which the method is based – a similarity measure between query results and the original KB without axioms – are not clearly justified, and no experimental validation is provided.

In contrast to the purely Boolean approach, some works in this context do explicitly consider keeping, along with the domain ontologies and KBs, the original documents in the retrieval model, as a fundamental part of the search (and answer) space, where the relation between ontologies and documents is established by *annotation relations*. In this line, KIM (Kiryakov, Popov, Terziev, Manov, & Ognyanoff, 2004; Popov, Kiryakov, Ognyanoff, Manov, & Kirilov, 2004) and TAP (Guha, McCool, & Miller, 2003) are examples of wide-ranging achievements on the construction of high-quality KBs, and the automatic annotation of documents on a large scale. Rather than the search itself, KIM focuses on the automatic population of ontologies from text corpora, along with the annotation of the latter. In one of the latest account of progress of this project (Kiryakov, Popov, Terziev, Manov, & Ognyanoff, 2004), a ranking model for retrieval is hinted at but is not been

developed in detail and evaluated. In fact, KIM relies on the Lucene⁵ keyword-based IR engine for this purpose (indexing, retrieval and ranking).

On its side, TAP presents a view of the search space (specifically the Web) where documents and concepts are nodes alike in a semantic network (Guha, McCool, & Miller, 2003), whereby the separation of contents and metadata is somewhat blurred. The research in TAP gave wide attention to infrastructural aspects (e.g., deployment support for KBs and distributed queries on the Web), and the presentation of results. With regards to the retrieval models themselves, the expressive power of the query language in TAP is fairly limited compared to languages such as SPARQL. Specifically, the supported capabilities are limited to keyword search within the “title properties” (marked as such in the ontology) of instances, and no ranking is provided.

Another work in this line is the one by Mayfield and Finin, which combines ontology-based techniques and text-based retrieval in sequence, in a blind relevance feedback iteration (Mayfield & Finin, 2003). Inference over class hierarchies and rules is used for query expansion, and the extension of semantic annotations. Documents are annotated with RDF triples, and ontology-based queries are reduced to Boolean string search, based on matching RDF statements with wildcards, at the expense of the expressive power for queries. It is interesting nonetheless how inference is used in this work to complete missing knowledge, ultimately relying on keyword-based search wherever the knowledge coverage by ontologies and metadata falls short.

The *ranking problem* has been taken up again in (Stojanovic, Studer, & Stojanovic, 2003), and more recently (Rocha, Schwabe, & de Aragão, 2004; Castells, Fernández, & Vallet, 2007). Rocha et al. propose the expansion of query results through arbitrary ontology relations starting from the initial query answer, where the distance to the initial results is used to compute a similarity measure for ranking (Rocha, Schwabe, & de Aragão, 2004). This method has the advantage of allowing the user to express information needs with simpler, keyword-based queries but in exchange, it is not possible to define more precise (structured) query conditions taking advantage of the vocabulary and semantic relations defined by the ontology. To confront that limitation, the ranking of documents is addressed in (Castells, Fernández, & Vallet, 2007) by combining semantic search with conventional keyword-based retrieval to achieve tolerance to knowledge base incompleteness. On their side, Stojanovic et al. propose a ranking scheme for ontology triples, based on the number of times an instance appears as a term in a relation type, and the derivation tree by which a sentence is inferred (Stojanovic, Studer, & Stojanovic, 2003). These three works are thus concerned with ranking formal answers to ontology-based queries, and therefore address a complementary problem to that of ranking the documents that are annotated by these answers.

⁵ Lucene information retrieval library, <http://lucene.apache.org/>

3.5 Semantics in Recommender Systems

Social systems build and keep a profile of each user, which is mainly composed of his relationships with others, and possible additional information about these relationships: reliability, frequency, context, etc. Connected to one another, users form graphs of social links, named in the literature as **social networks** (Wasserman & Faust, 1994). In these graphs, users' relationships with others may be explicitly described by users themselves in the system, or can be indirectly discovered from different sources of information, such as address books, IRC contact lists, or e-mail message boxes. For example, co-authorship or co-citation of people in scientific publications, web pages, etc., can be used to build a social network. Text classification techniques can be applied to e-mails in order to contextualise and define the topic of relationships, and so forth. In fact, approaches have been recently proposed that automatically collect the above and other types of social network information from the Web in order to apply methods of Semantic Network Analysis (SNA) for the study of online communities (Mika, *Ontologies are Us: A Unified Model of Social Networks and Semantics*, 2005).

To model the social profile of a user, the relationships between users can also be formalised using ontologies. The Friend-Of-A-Friend (FOAF) ontology is one of the most popular in this area. It aims to create a network of machine-readable pages describing people, the links between them and the things they create and do. FOAF is a technology that makes it easier to share and use information about people, their activities and their resources (e.g., photos, calendars, web blogs), to transfer information between websites, and to automatically extend, merge and reuse it online.

Flink (Mika, *Flink: Semantic Web Technology for the Extraction and Analysis of Social Networks*, 2005) is a system for the extraction, aggregation and visualisation of online social networks. It employs semantic technologies for reasoning with personal information extracted from a number of electronic information sources including web pages, emails, publication archives, and FOAF profiles. Extending the traditional bipartite model of ontologies (concepts and instances) with the social dimension leads to a tripartite model of the Semantic Web, namely the layer of communities and their relations (users), the layer of semantics (ontologies and their relations) and the layer of content items and their relations (the hypertext Web). The application of this representation is demonstrated in (Mika, *Social Networks and the Semantic Web: The Next Challenge*, 2005) showing how community-based semantics emerges from this model through a process of graph transformation.

ONTOCOPI (Alani, O'Hara, & Shadbolt, 2002) is a tool for discovering **Communities of Practice** (Wenger, 2000), CoP, by analysing ontologies of a given relevant domain of discourse. It aims to disclose informal CoP relations by

identifying patterns in the relations represented in ontologies, and traversing the ontology from instance to instance via selected relations. Performing experiments to determine particular CoP from an academic ontology, the authors show how the alteration of the weights applied to the ontology's relations affect the structure of the identified CoP.

Up to date, one of the most significant uses of social relations and CoP is the implementation of ***social collaborative filtering*** strategies. The most popular collaborative filtering implementations require either a critical mass of referenced resources or a lot of active users. Recent collaborative recommendation solutions are based on finding referrals with expertise on the given domain of discourse. *FOAFRealm* (Kruk & Decker, 2005) is a distributed user profile management system based on the FOAF metadata. It enables the collaboration among people in order to develop effective information retrieval. In the system, users' managed collections are exploited to provide a collaborative filtering strategy that makes use of the social network maintained by the users themselves. Apart from the explicit FOAF friendship relations, the framework controls the access to personal resources, giving different weights to votes during negotiations and specifying the maximum length of the path between different people.

In (Golbeck & Mannes, 2006), a novel approach for inferring relationships using provenance information and trust annotations in Semantic Web-based social networks is presented. A recommender application, *FilmTrust* (Golbeck & Hendler, 2006), combines the computed trust values with the provenance of other annotations to personalise the website. The *FilmTrust* system uses trust to compute personalised recommended movie ratings, and to order reviews. The results obtained with *FilmTrust* illustrate the success that can be achieved using the proposed method. The authors show that the obtained recommendations are more accurate than other techniques when the user's opinions about a film are divergent from the average, thus addressing the grey sheep problem.

In addition to explicit social relations, recent researches focus their attention on finding ***implicit relations among people***, according to personal tastes, interests and preferences. Hence, for example, the work (Liu, Maes, & Davenport, 2006) presents a theory and implementation of "taste fabrics", a semantic mining approach to the modelling and computation of personal tastes for different topics of interests. The taste fabric affords a flexible representation of a user in taste-space, enabling a keyword-based profile to be 'relaxed' by a spreading activation (Cohen & Kjeldsen, 1987; Crestani & Lee, 2000) pattern on the taste fabric. An evaluation of taste-based recommendation using the taste fabric implementation shows that it compares favourably to classic collaborative filtering recommendation methods, and whereas collaborative filtering is an opaque mechanism, recommendation using taste fabrics can be effectively visualised, thus enhancing transparency and user trust.

In addition to the explicit and implicit definition of social relations (and the subsequent discovery of social communities) to be exploited by recommender systems, other works have focused on incorporating ***semantic-based knowledge representations*** to describe user and/or item profiles, and making enhanced, more understandable recommendations.

An adaptation of the item-based collaborative filtering method (see Section 2.3.2) integrating semantic similarities for items with rating- or usage-based similarities is presented in (Mobasher, Jin, & Zhou, 2004). The authors propose to modify the item similarity formulas 2.18, 2.19 and 2.20 by adding a component that computes semantic content-based similarities between items. The reported experimental results demonstrate that the integrated approach yields significant advantages both in terms of improving accuracy, as in dealing with sparse datasets or new items (cold-start problem). Moreover, in (Jin & Mobasher, 2003), the previous combined item similarity is also used to fill the original rating matrix, showing again that the proposed method helps to alleviate the sparsity problem.

An approach to ontological user profiling in a recommender system is presented in (Middleton, Roure, & Shadbolt, 2004). Working on the problem of recommending on-line academic research papers, the authors present two systems, *Quickstep* and *Foxtrot*, which create user profiles monitoring the behaviour of the users, and gathering relevance feedback from them. The obtained profiles are represented in terms of a research topic ontology. Research papers are classified using ontological classes, and the proposed collaborative recommendation algorithms suggest documents seen by similar people on their current topics of interest. In this scenario, ontological inference is shown to ease user profiling, external ontological knowledge seems to successfully improve the recommendations, and the profile visualisation is used to enhance profiling accuracy.

More recently, Anand and Mobasher take up again the issue that most currently available recommender systems still tend to use very simplistic user models to generate recommendations (Anand & Mobasher, 2007). For example, in user-based collaborative filtering, as more ratings are provided by the user, they are simply added to the existing set of ratings, and all ratings are used in discovering the active user's neighbourhood. Similarly, content-based techniques tend to just update the bag-of-words or probabilities as new items are rated. The authors contend for a fundamental shift in terms of how a user is modelled in a recommender system. Specifically, they distinguish between a user's long term and short term memories, and propose a recommendation process that uses these two memories. Context-based retrieval cues are obtained to retrieve relevant preference information stored in the long term memory, and the identified relevant preferences are used in conjunction with the information stored in the short term memory to make recommendations. The paper introduces three types of contextual cues: collaborative, behavioural and semantic,

and provides empirical evidence that the approach improves recommendation quality.

An implementation of the semantic contextualisation proposed in the previous work is described in (Sieg, Mobasher, & Burke, 2007). In this case, the authors present a strategy for personalised search that involves building models of user contexts as ontological profiles by assessing implicitly derived interest scores to concepts defined in a domain ontology. A spreading activation algorithm is used to maintain the interest scores based on the user's ongoing behaviour. The conducted experiments show that re-ranking the search results based on the interest scores and the semantic evidence in an ontological user profile are effective in presenting the most relevant results to the user.

Finally, (Shoval, Maidel, & Shapira, 2008) proposes the incorporation of a common ontology which enables describing both the users' and the items' profiles with concepts taken from the same vocabulary. Based on this representation approach, and utilising the ontology hierarchy, the authors present a content-based method for filtering items for a given user. The active user's profile is compared with the item profiles using a similarity measure that takes into account the occurrence of common concepts in both profiles, as well as the existence of "related" items according to their position in the ontology hierarchy. Based on the computed similarities, items are ranked for the user. At the time of this writing, the method is being implemented in *ePaper*, a personalised electronic newspaper, using an ontology that mirrors the two first levels of the IPTC⁶ news taxonomy, which was specifically designed for classification of news items.

Our semantic-based knowledge representation and recommendation proposals, covered by Chapters 4, 5 and 6, and their integrated implementation in a news recommender system, described in Chapters 7 and 8, share many characteristics with the recent and on-going works outlined in this section. The following are some of these commonalities:

- **Ontology-based knowledge representation.** Similarly to (Mika, Flink: Semantic Web Technology for the Extraction and Analysis of Social Networks, 2005), we base and focus our research on a tripartite knowledge model, where user and item spaces are connected through a semantic one. As done in (Shoval, Maidel, & Shapira, 2008), we propose to build this layer in terms of concepts available in domain ontologies. See Section 4.1 for more details.
- **Spreading of semantic preferences.** The extension of ontology-based user profiles through the semantic relations of the domain ontologies (Sieg, Mobasher, & Burke, 2007) is also present in our work (Section 4.1). As

⁶ International Press Telecommunications Council, <http://www.iptc.org/>

concluded in (Jin & Mobasher, 2003; Mobasher, Jin, & Zhou, 2004), we show that this strategy is beneficial to mitigate the sparsity and cold-start problems.

- **Semantic personalised and context-aware recommendations.** Personalisation (Anand & Mobasher, 2007) and contextualisation (Sieg, Mobasher, & Burke, 2007) of content retrieval exploiting an ontological knowledge representation are described in Sections 4.2 and 4.3.
- **Implicit communities of interest.** Like (Liu, Maes, & Davenport, 2006), and opposed to (Golbeck & Mannes, 2006), where explicit user relations are exploited for recommendation purposes, we discover implicit user relations (communities) from the similarities existing among semantic user preferences. In our case, the identification of such communities is carried out at different semantic interest layers, laying the ground for building what we shall call multilayered Communities of Interest (Sections 5.1 and 5.2).
- **Semantic content-based collaborative recommendations.** Explicit item-based collaborative recommendation from ontological user profiles was presented in (Middleton, Roure, & Shadbolt, 2004). Here, we propose the exploitation of the underlying multilayered communities found by our approach, for making group-oriented and hybrid recommendations (Sections 4.4 and 5.3). Experimental results of these recommendation techniques are given in Chapter 6.
- **A prototype recommender system.** The integration and evaluation of our content-based and collaborative recommendation strategies in a news recommender system is reported in the final part of the thesis (see Chapters 7 and 8). Similarly to *ePaper* system (Shoval, Maidel, & Shapira, 2008), our prototype will make use of the IPTC news codes ontology to describe both user and item profiles.

3.6 Summary

The enhancement of the semantic dimension to describe both user preferences and item content features is an emerging research trend in recommender systems. For that purpose, the wide experience in semantic-based techniques in related areas such as Information Retrieval and User Modelling provide a wealth to leverage from.

Recent works are expanding on the benefits that can be reaped by adding semantic capabilities to recommendation strategies. Spreading related semantic preferences in user profiles enable strategies to address sparsity and cold-start effects. Describing user and item profiles in terms of unambiguous semantic concepts enables a finer, more precise knowledge of the user tastes and relations and the item

content by the recommender systems. Moreover, the consideration of semantic short-term preferences according to recent rating and behaviour patterns of the user facilitates the design and development of context-aware recommendation models.

On the other hand, not only the advantages of using semantic-based approaches are inherited for recommender systems, but also its problems. Open challenges arise, bringing the opportunity for further research. Problems such as semantic preference learning, semantic item annotation, or domain ontology population get thus brought into the research agenda of recommender systems.

Part II

Recommendation models: an ontology-based proposal

Chapter 4

Content-based recommendation: a semantic-intensive approach

Content-based recommender systems provide suggestions based on an analysis of the content features of the items a user has searched for, rated or purchased in the past. In many domains, the definition and automatic capture of such features are very complex tasks (e.g., video and audio signal processing). In fact, as mentioned in Section 2.2, content-based recommenders have been mostly designed to recommend textual items, in which the text contents are usually described as a bag of keywords.

Alternatively to such approaches, in this chapter, we propose a knowledge representation based on ontologies. User preferences and item features are defined in terms of concepts (classes or instances) belonging to a set of domain ontologies. As we describe in Section 4.1, the semantic relations between concepts defined by such ontologies enable not only a better, more detailed “machine understanding” of the contents, but also the definition of a semantic spreading strategy which extends and enriches the user profiles, providing means for the mitigation of the sparsity problem.

Building upon this knowledge representation approach three recommendation models are proposed. The first one, explained in Section 4.2, suggests items to a single user considering only the semantic preferences expressed in his profile. The second, described in Section 4.3, incorporates semantic contextual information into the recommendation process according to the concepts occurring in those items the user has recently browsed, evaluated or rated as relevant. These concepts do not really belong to the actual profile, but are assumed to relate, to some extent, to the current (short-term) preferences of the user for a specific task or goal. Finally, in Section 4.4, we introduce several strategies that merge a number of user profiles in order to provide group-oriented recommendations.

4.1 Semantic user profiles and preference extension

Ontology-based knowledge representation

In contrast to other approaches in personalised content retrieval and recommendation, our general recommendation approach makes use of explicit user profiles (as opposed to for example sets of preferred documents). Working within an ontology-based personalisation framework (Vallet, Castells, Fernández, Mylonas, & Avrithis, 2007), user preferences are represented as vectors $\mathbf{u}_m = (u_{m,1}, u_{m,2}, \dots, u_{m,K}) \in [-1, +1]^K$ where the weight $u_{m,k} \in [-1, +1]$ measures the intensity of the interest of user $u_m \in \mathcal{U}$ for concept $c_k \in \mathcal{O}$ (a class or an instance) in a domain ontology \mathcal{O} , K being the total number of concepts in the ontology. A positive preference value indicates that the user is interested in the concept, while a negative one reflects a user dislike for the concept. Similarly, the items $d_n \in \mathcal{I}$ in the retrieval space are assumed to be described (annotated) by vectors⁷ $\mathbf{d}_n = (d_{n,1}, d_{n,2}, \dots, d_{n,K}) \in [0, 1]^K$ of concept weights, in the same vector-space as user preferences. Based on this common logical representation, measures of user interest for content items can be computed by comparing preference and annotation vectors, and these measures can be used to prioritise, filter and rank contents (a collection, a catalogue, a search result) in a personal way. Figure 4.1 shows our twofold-space ontology-based knowledge representation, in which M and N are respectively the number of users and items registered in the system.

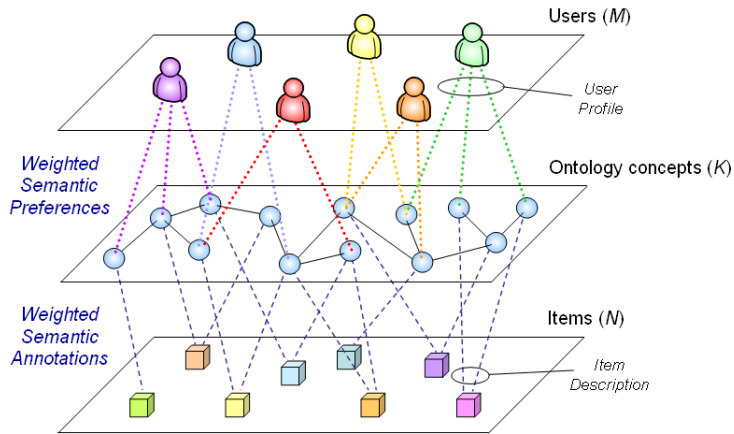


Figure 4.1 Ontology-based user profiles and item descriptions.

⁷ From now on, the notation for information items changes from i_n , which was introduced in Chapter 2, to d_n , reinforcing the idea that we usually have documents as information items. A person used to read Information Retrieval works will find this notation more natural in this chapter. Formulas presented in Chapters 5 and 6 will also be more easy-readable.

The ontology-based representation is richer 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 sports, science fiction movies, or stock markets, 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 dynamic, context-dependent relevance. An ontology provides further formal, computer-processable meaning on the concepts (who is coaching a team, an actor's filmography, financial data on a stock), and makes it available for the personalisation system to take advantage of.

The main benefits of using a concept-based user profile representation in contrast to common keyword-based approaches would then be the following:

- **Semantic richness.** Ontology concept-based preferences are more precise, and reduce the effect of the ambiguity caused by simple keyword terms. For instance, if a user states an interest for the keyword “java”, the system does not have further information to distinguish *Java, the programming language*, from *Java, the Pacific island*. A preference stated as “ProgrammingLanguage:Java” (this is read as the instance Java from the Programming Language class) lets the system understand unambiguously the preference of the user, and also allows the exploitation of more appropriate related semantics (e.g., synonym, hypernym, subsumption, etc.). This, together with disambiguation techniques, might lead to the effective recommendation of text-annotated items.
- **Hierarchical representation.** Ontology concepts are represented in a hierarchical way, through different hierarchy properties, such as *subClassOf*, *instanceOf* or *partOf*. Parents, ancestors, children and descendants of a concept give valuable information about the semantics of the concept. For instance, the concept *leisure* might be highly enriched by the semantics of each leisure activity, which would be described by the hypothetical taxonomy that the concept could subsume.
- **Inference.** Ontology standards introduced in Section 3.3.4, such as RDF and OWL, support inference mechanisms that can be used to enhance recommendation, so that, for instance, a user interested in *animals* (superclass of *dog*) is also recommended items about *dogs*. Inversely, a user interested in *skiing*, *snowboarding* and *ice hockey* can be inferred with a certain confidence to be globally interested in *winter sports*. Also, a user keen on *Spain* can be assumed to like *Madrid*, through the *locatedIn* transitive relation, assuming that this relation had been seen as relevant for inferring previous underlying user's interests.

Figure 4.2 shows an example of conceptualised preferences. Having a set of three domain ontologies with information about art works, institutions and regions,

suppose a user indicates an interest for the topic “visual art works”, which is represented in the ontologies as a class *Visual Art Work* inheriting from the main class *Art Work*. The system is then able to infer preferences for *Visual Art Work* subtopics (through the general property *subClassOf*), obtaining finer grain details about the user preferences, such as potential interests in paintings and photographs. Note that original and more specific preferences will prevail over the system’s inference. In this case, as highlighted in the figure, the user is not interested in the concept *movie*, whose negative weight prevails over the higher-level topic inference.

Apart from hierarchical properties, other arbitrary semantic relations can be exploited for preference extension. Assuming the user has an additional preference for “Madrid” – an instance of the class *City* in the region ontology – the properties *exhibitedAt* and *locatedIn* could be exploited in order to infer new interests. Firstly, assuming a sufficient degree of interest for *Painting*, the system could use the *exhibitedAt* property in order to infer that the user could be interested in “museums” in general. Secondly, given that the user is interested in *Madrid*, the system could determine the inferred interests for museums in that city, thanks to the use of *locatedIn* and *instanceOf* properties. Thus, for instance, the system could find out a potential interest for “El Prado” museum. Recommendations about paintings exhibited in El Prado could then be suggested to the user, although no explicit preferences for such elements had been previously declared.

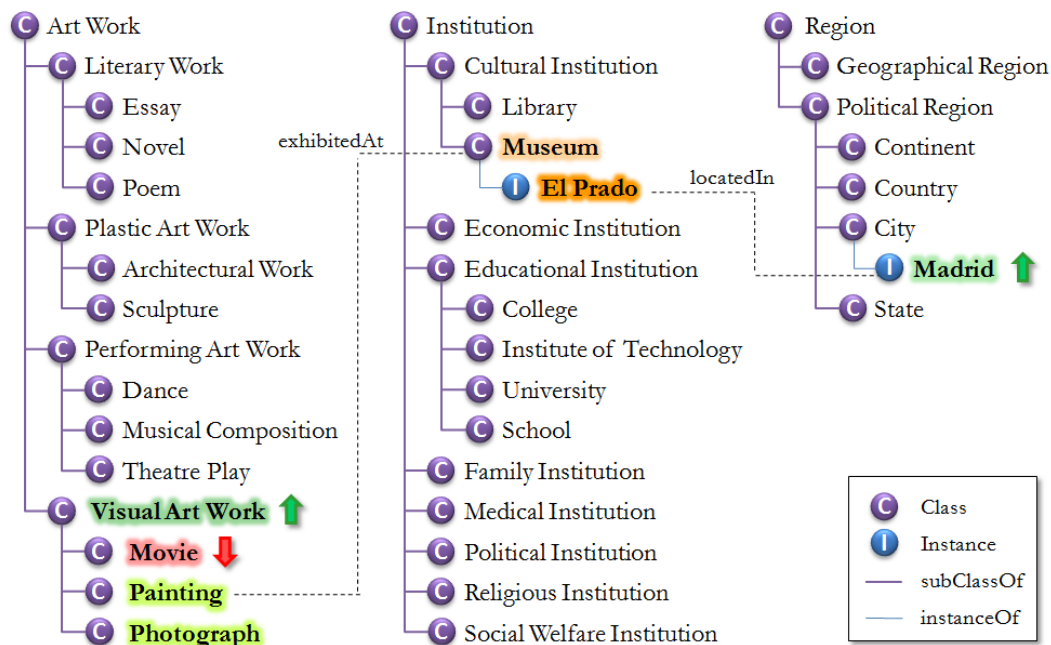


Figure 4.2 Representation of user preferences as concepts of domain ontologies.

In addition to the above benefits, this kind of knowledge representation provides additional advantages thanks to the use of Semantic Web technologies.

- **Portability.** Based on XML standards, the domain knowledge, item annotation, and user profile information could be easily distributed, adapted and integrated in different recommender systems for different applications. The machine-processable nature of such standards also would allow the automatic transformation of the available metadata into a visual representation easily understandable by humans (e.g., by using HTML documents).
- **Domain independency.** Using an ontology-based knowledge representation, content retrieval and recommendation algorithms can be designed independently from the domain of discourse. Ontology hierarchies, concepts (in the form of classes and instances), and relations are the elements to be taken into consideration for the definition of new models. In principle, no domain-dependent restrictions would affect the implementation and reuse of such models. This is not feasible for example in model-based recommender systems, where probabilistic models are built from the available data, and cannot be used in different domains, unless the entire model is rebuilt with new data.
- **Multi-source annotation.** Assuming the existence of manual or automatic mechanisms to semantically annotate any type of content (text, video, audio, etc.), ontology-based recommender systems could suggest items from multiple different sources without the need of changing their inner recommendation algorithms.

Carrying further domain knowledge than simple keyword terms, ontology concepts and their semantic relations will be exploited by the recommendation models presented in this work. Introduced in the example of Figure 4.2, a key point of these models will be the extension of user preferences and item annotations through the ontology properties that relate all of them. In the following, we describe the developed algorithm to spread semantic concepts of user and item profiles.

Semantic extension of user preferences

In real scenarios, user profiles tend to be very scattered, especially in those applications where user profiles have to be manually defined. 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. On the other hand, applications where an automatic preference learning algorithm is applied tend to recognise the main characteristics of user preferences, thus yielding profiles that may entail a lack of expressivity.

To overcome this problem, (Vallet, Castells, Fernández, Mylonas, & Avrithis, 2007) proposes a semantic preference spreading mechanism which expands the initial set of preferences stored in user profiles through explicit semantic relations with

other concepts in the ontology (Figure 4.3). The approach is based on the so called Constrained Spreading Activation (CSA) strategy (Cohen & Kjeldsen, 1987; Crestani, 1997; Crestani & Lee, 2000). The expansion is self-controlled by applying a decay factor to the intensity of preference each time a relation is traversed, and taking into account constraints (threshold weights) during the spreading process.

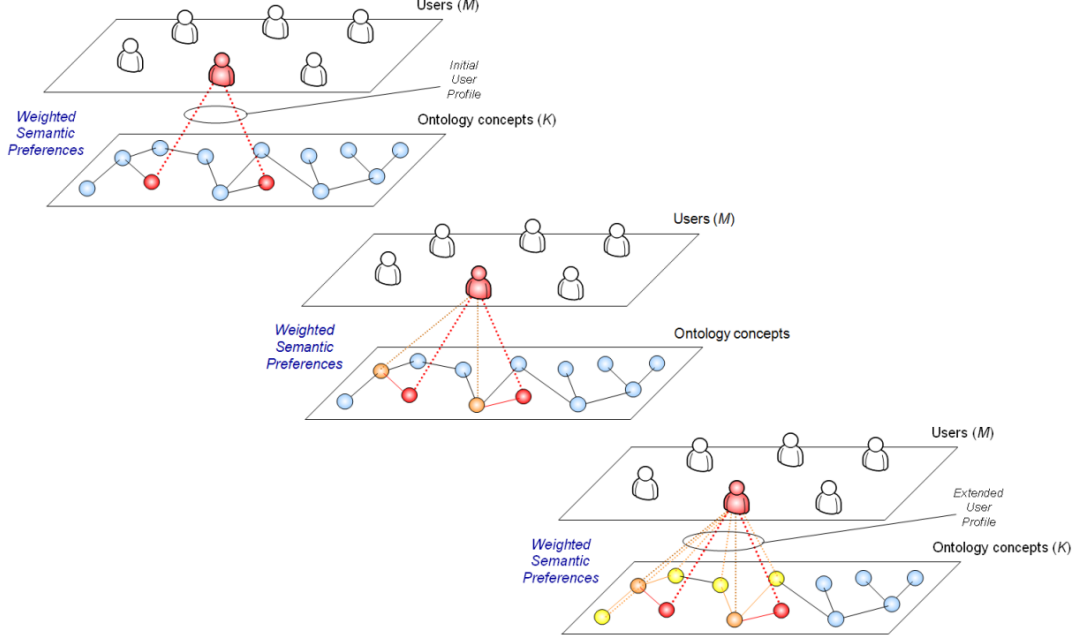


Figure 4.3 Semantic preference extension.

The activation of user preferences is based on an approximation to conditional probabilities. Let $p_u(c_x) = u_x \in [-1, +1]$ be the preference (interest/dislike) of the user $u \in \mathcal{U}$ for the ontology concept $c_x \in \mathcal{O}$. The probability that c_x is relevant for the user can be expressed in terms of the probability that c_x and each concept c_y directly related to c_x in the ontology belong to the same topic, and the probability that c_y is relevant for the user. A similar formulation could be given for non-relevant concepts. With this definition, the relevance of c_x for the user can be computed by a CSA algorithm, starting with the initial set of semantic concepts \mathbf{P}_u in the user profile, i.e., $\mathbf{P}_u = \{c_k \in \mathcal{O} | p_u(c_k) \neq 0\}$.

Let \mathcal{R} be the set of all relations in \mathcal{O} . The spreading strategy is based on weighting each semantic relation $r \in \mathcal{R}$ with a measure $w(r, c_x, c_y)$ that represents the probability that given the fact that $r(c_x, c_y)$ holds, c_x and c_y belong to the same topic. This is used for estimating the relevance of c_y when c_x is relevant for the user. The weight $w(r, c_x, c_y)$ is interpreted as the probability that c_y is relevant for the user if we know that the concept c_x is relevant for the user, and $r(c_x, c_y)$ holds.

With this measure, concepts are expanded through the semantic relations of the ontology, using a constrained spreading activation mechanism over the semantic network defined by these relations. As a result, the initial set of concepts \mathbf{P}_u is extended to a larger vector \mathbf{EP}_u , where $\mathbf{EP}_u[c_k] \geq \mathbf{P}_u[c_k]$ for all $c_k \in \mathcal{O}$.

Let \mathcal{R}^{-1} be the set of all inverse relations of \mathcal{R} , i.e., a concept c_x has an inverse relation $r^{-1}(c_x, c_y) \Leftrightarrow \exists r(c_y, c_x) \mid r \in \mathcal{R}$. Let $\widehat{\mathcal{R}} = \mathcal{R} \cup \mathcal{R}^{-1} = \mathcal{R} \cup \{r^{-1} \mid r \in \mathcal{R}\}$, and $w : \widehat{\mathcal{R}} \rightarrow [0, 1]$. The extended concept vector \mathbf{EP}_u is computed by:

$$\mathbf{EP}_u[c_y] = \begin{cases} \mathbf{P}_u[c_y] & \text{if } \mathbf{P}_u[c_y] > 0 \\ R\left(\left\{\mathbf{EP}_u[c_x] \cdot w(r, c_x, c_y) \cdot \text{power}(c_x)\right\}_{c_x \in \mathcal{O}, r \in \widehat{\mathcal{R}}, r(c_x, c_y)}\right) & \text{otherwise} \end{cases}, \quad (4.1)$$

where $\text{power}(c_x) \in [0, 1]$ is a propagation power assigned to each concept c_x (by default, $\text{power} = 1$), and

$$R(\mathbf{X}) = \sum_{S \subseteq \mathbb{N}_n} \left\{ (-1)^{|S|+1} \times \prod_{i \in S} x_i \right\},$$

having $\mathbf{X} = \{x_i\}_{i=0}^n, x_i \in [0, 1]$.

For further details about the previous formula, the reader is referenced to (Crestani, 1997). Figure 4.4 shows a simple example of the 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 shows how a third preference is inferred, accumulating the evidence of relevance from the original two preferences.

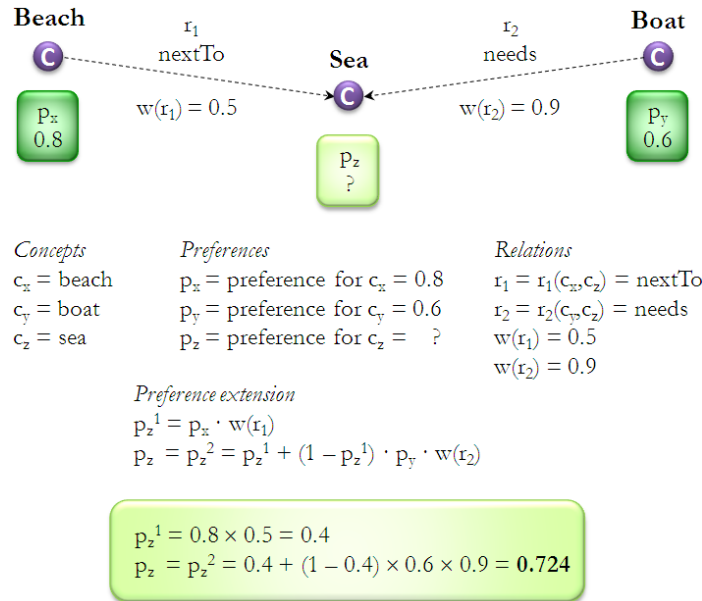


Figure 4.4 Example of semantic preference extension computation.

The pseudocode of the entire expansion algorithm is presented in Figure 4.5. Before, in Table 4.1, we describe a set of parameters that have been included in the algorithm to avoid cases of excessive semantic propagation.

Parameter	Description
ε	<p>The minimum threshold weight a concept has to have in order to expand its weight to related concepts.</p> <p>A high threshold value improves the performance of the spreading algorithm, as less expansion actions are made. However, higher threshold values exploit less the underlying semantics of the KB, thus resulting in poorer propagation inferences.</p>
n_e	<p>The maximum number of expansion steps to be performed by the spreading algorithm.</p> <p>Similarly to the ε threshold, the parameter n_e has to be set as a trade-off between performance and inference quality.</p>
n_h	<p>The maximum number of times a concept can be generalised.</p> <p>This parameter is equivalent to n_e applied to hierarchical relations, like <i>subClassOf</i>. Once a concept has been expanded up to n_h hierarchical levels, it would be convenient not to expand it more. The intention of this constrain is to not generalise a preference (semantically) too much, as this type of expansion is a risky assumption with the original user's preferences. For instance, in the example given at the beginning of this section, where the user likes <i>skiing</i>, <i>snowboarding</i> and <i>ice hockey</i>, the system can infer quite safely that is likely the user will be interested in other <i>winter sports</i>, but it could be self-defeating to infer a preference for any kind of <i>sport</i> in general.</p>
n_f	<p>The maximum fan-out (i.e., number of output properties) a concept can have to be expanded.</p> <p>The aim of this constrain is to reduce the “hub effect” in concepts with many relations to other concepts. For instance, if a user likes a group of companies that trade on the NASDAQ stock market and belong to the Telecommunication sector, a correct inference is that the user might be interested in other companies with these two features. Nonetheless, it could be considered incorrect to propagate that preference to the concept <i>Company</i>, and expand it to hundreds of other companies vaguely related to the original set.</p>
$\text{power}(c_x)$	<p>The propagation intensity (strength) of a concept.</p> <p>This factor multiplies the effect of propagating the concept weight. By default, it is set to 1.</p>
$w(r, c_x, c_y)$ $w(r)$	<p>The propagation decay of a relation between two given concepts.</p> <p>This parameter approximates the probability that a concept c_y is relevant given that c_x is relevant and relation $r(x, y)$ holds. It can be seen as the propagation power of the relation $r \in \mathcal{R}$ for concepts c_x and c_y.</p> <p>The definition of the values of this parameter for each relation might be critical, and is very difficult to decide. In the experiments conducted in this work, these values were empirically fixed for each ontology property, not taking into account the involved concepts of the relation, so they can be expressed as $w(r)$ instead of $w(r, c_x, c_y)$.</p>

Table 4.1 Parameters of the semantic spreading algorithm.

```

function expand(P, EP, w) {
    // Init the expanded concept weights with the input ones
    for (  $c_x \in \mathcal{O}$  ) {
        EP[ $c_x$ ] = P[ $c_x$ ]
    }

    // Create a priority queue based on concept weights (initially null)
    Q ← buildPriorityQueue( $\mathcal{O} \times \{\text{prev}=0, \text{hierarchyLevel}=0, \text{expansionLevel}=0\}$ )

    while ( Q.isEmpty() == false ) {
        // Extract the next concept to expand
        ( $c_x$ ,  $\text{prev}_x$ , hierarchyLevel, expansionLevel) ← Q.pop()

        // Check the minimum concept weight constraint
        if ( EP[ $c_x$ ] <  $\epsilon$  ) {
            exit // The remaining concept weights are also below  $\epsilon$ 
        }

        // Check the maximum expansion constrain
        if (expansionLevel ≥  $n_e$ ) {
            goto while
        }

        // Expand the neighbourhood of the current concept
        for ( $\{r, c_y\} \in c_x.\text{getNeighbourhood}()$ ) {
             $\text{prev}_y$  = EP[ $c_y$ ]

            // Check the hierarchical level expansion constrain
            if (EP[ $c_y$ ] = 1 OR ( $r.\text{isHierarchical}()$  AND hierarchyLevel ≥  $n_h$ )) {
                goto for
            }

            // "Undo" the last update from  $c_x$ 
            EP[ $c_y$ ] ← (EP[ $c_y$ ] -  $w(r, c_x, c_y) * \text{power}(c_x) * \text{prev}_x$ ) /
                (1 - EP[ $c_y$ ] *  $w(r, c_x, c_y) * w_f(c_x, n_f) * \text{power}(c_x) * \text{prev}_x$ )

            // Do the propagation taking into account the fan-out factor
            EP[ $c_y$ ] ← EP[ $c_y$ ] + (1 - EP[ $c_y$ ]) *  $w(r, c_x, c_y) * w_f(c_x, n_f) * \text{power}(c_x) * \text{EP}[c_x]$ 

            if ( $r.\text{isHierarchical}()$ ) {
                hierarchyLevel++;
            }

            Q.push( $c_y, \text{prev}_y, \text{hierarchyLevel}, \text{expansionLevel}$ )
        }

        expansionLevel++
    }
}

```

Figure 4.5 Pseudocode of the semantic spreading algorithm.

A recommender system could output ranked lists of content items taking into account not only the initial user profiles, but also the semantic extension of user preferences and item annotations. In Chapter 6, we present experiments showing that the performance of our recommendation models is considerably poorer when the spreading mechanism is not enabled. Typically, the basic user profiles without expansion are too simple. They provide a good representative sample of user preferences, but do not reflect the real extent of user interests, which results in low overlaps between the preferences of different users. Moreover, the preference extension is not only important for the performance of personalised recommendations, but is essential for the clustering strategy of the collaborative models described in Chapter 5. Before showing that, in the rest of this chapter, we focus on the basis of our content-based models, i.e., the proposed ontology-based personalised, context-aware, and group-oriented recommendation techniques.

4.2 Semantic personalised content retrieval

Once a rich representation of user interests is available, we propose to relate it to content semantics in order to predict the relevance of content items, considering not only a specific user request but the overall needs of the user. Our content retrieval framework assumes the availability of a corpus \mathcal{I} of items (texts, multimedia documents, etc.), annotated by domain concepts (instances or classes) from an ontology-based knowledge base \mathcal{O} . That is, each item $d_n \in \mathcal{I}$ is associated to a set of semantic annotations $\mathbf{d}_n = (d_{n,1}, d_{n,2}, \dots, d_{n,K})$, where $d_{n,k} \in [0,1]$ indicates the degree to which the concept $c_k \in \mathcal{O}$ is important in the meaning of d_n , and $K = |\mathcal{O}|$ is the number of concepts in the KB. Based on these annotations a semantic index (see Section 3.3.3) is built, associating the contents to weighted semantic metadata that describe the meaning carried by the items in terms of the domain ontology \mathcal{O} . In Sections 8.1 and 8.2, when building and evaluating a ontology-based recommender system, we shall come back to this issue, and present a novel approach to automatically annotate textual items.

Through the ontology layer there is a fuzzy relationship between users and the indexed content of the system. Although the use of this ontology layer is transparent to the user, the system can take advantage of its unambiguous, richer relations, and inference capabilities, as explained in the previous section. Based on preference weights, measures of user interest for content units can be computed, with which it is possible to discriminate, filter and rank contents (a search result, a collection, a catalogue) in personalised and collaborative ways.

Our first content retrieval model (wrapped by the ‘Item Retrieval’ component in Figure 4.6) makes personalised item recommendations for a single user, and works in

two phases. In the first one, a formal ontology-based query (e.g., in RDQL) is issued by some form of query interface (e.g., NLP-based) which formalises a user information need. The query is processed against the knowledge base using any desired inference or query execution tool, outputting a set of ontology concept tuples⁸ that satisfy the query. From this point, the second retrieval phase is based on an adaptation of the classic vector-space IR model (Section 2.2), where the axes of the vector space are the concepts of \mathcal{O} , instead of text keywords. Like in the classic model, in ours the query and each item are represented by vectors \mathbf{q} and \mathbf{d} , so that the degree of satisfaction of a query by an item can be computed by the cosine measure:

$$\text{sim}(\mathbf{d}, \mathbf{q}) = \cos(\mathbf{d}, \mathbf{q}) = \frac{\mathbf{d} \cdot \mathbf{q}}{\|\mathbf{d}\| \times \|\mathbf{q}\|} = \frac{\sum_{k=1}^K d_k q_k}{\sqrt{\sum_{k=1}^K d_k^2} \sqrt{\sum_{k=1}^K q_k^2}}$$

Note that the dimension of these vectors, as formally defined above, is K , but since the number of non-zero coordinate values is in practice orders of magnitude lower than K , the computation of the previous and subsequent formulas based on the cosine measure can be fast and easily optimised.

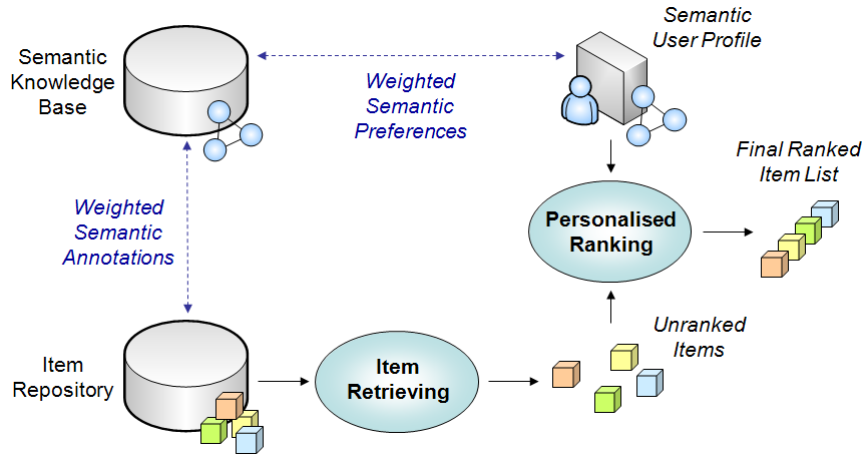


Figure 4.6 Personalised ontology-based content retrieval.

The problem, however, is how to build the \mathbf{q} and \mathbf{d} vectors. We do not address this issue here, and we rely on the state of the art on this subject, as the obtention of query and item vectors is not in the focus of this thesis. We shall not deal with

⁸ As defined in Section 3.3.4, a tuple (triple or sentence) is the basic unit of representation in RDF, which consists of two nodes (subject and object) linked by a directed edge (predicate). The nodes represent resources, and the edge represents a property that relates the two nodes. For example, a tuple could describe the fact that the author (predicate) of “Starry Night” painting (subject) was Vicent Van Gogh (object).

semantic search or query-driven recommendation approaches either. For possible ways to provide such functionalities, see (Vallet, Castells, Fernández, Mylonas, & Avrithis, 2007). Instead, we continue explaining our content retrieval process with its personalisation phase (component ‘Personalised Ranking’ in Figure 4.6).

Our personalised recommendation strategy is built as an extension of the ontology-based knowledge representation model presented in the previous section. It shares the expressiveness of ontologies to define user interests on the basis of the same concept space that is used to describe contents. Assuming a semantic profile of user preferences has been obtained, either automatically or manually, our notion of personalised content retrieval is based on the definition of a matching algorithm that provides a personal relevance measure $\text{pref}(\mathbf{d}, \mathbf{u})$ of an item \mathbf{d} for a user \mathbf{u} . This measure is set according to the semantic preferences of the user, and the semantic annotations of the item. The procedure for matching \mathbf{d} and \mathbf{u} is based again on a cosine function for vector similarity computation:

$$\text{pref}(\mathbf{d}, \mathbf{u}) = \cos(\mathbf{d}, \mathbf{u}) = \frac{\mathbf{d} \cdot \mathbf{u}}{\|\mathbf{d}\| \times \|\mathbf{u}\|} = \frac{\sum_{k=1}^K d_k u_k}{\sqrt{\sum_{k=1}^K d_k^2} \sqrt{\sum_{k=1}^K u_k^2}}. \quad (4.2)$$

The formula matches two weighted-concept vectors and produces a value in $[-1, +1]$. Values close to -1 are obtained when the two vectors are dissimilar, and indicate that user preferences negatively match the content metadata. On the other hand, values close to $+1$ indicate that user preferences significantly match the content metadata, which means a potential interest of the user for the item. Since the annotated content is considered an external resource by our model, we assume that the annotation may lack weights, or even a clear weighting criterion. In such situations, the personalisation function assigns a weight of $+1$ by default to all metadata.

If no query is executed to limit the items to which formula 4.2 has to be applied, the personalisation strategy can be used to filter and order lists of items while browsing, which is in essence the purpose of any recommendation technique. From this point, this is the content retrieval scenario we assume for all the presented recommendation models.

Figure 4.7 shows an example of the computation of the preference value, in a simplified setting where $\mathcal{O} = \{\text{Building}, \text{Flower}, \text{Sea}, \text{Sky}, \text{Tree}\}$ is the set of all domain ontology concepts (classes and instances). The user is interested in “Mountain”, “Sea” and “Sky”, with different positive intensity, and has a negative preference for “Flower” and “Tree”. Hence, the preference vector for this user is $\mathbf{u} = (0.0, -0.3, 0.9, 0.7, 0.5, -0.1)$. Similarly, an image is annotated with the concepts “Building”, “Sea” and “Sky”, therefore the corresponding metadata vector is

$\mathbf{d} = (0.8, 0.0, 0.0, 0.6, 0.4, 0.0)$.

$\mathcal{O} = \{\text{Building, Flower, Mountain, Sea, Sky, Tree}\}$



User preferences	
Class	Weight
Flower	-0.3
Mountain	0.9
Sea	0.7
Sky	0.5
Tree	-0.1

Content metadata	
Class	Weight
Building	0.8
Sea	0.6
Sky	0.4

(Building, Flower, Mountain, Sea, Sky, Tree)

$\mathbf{u} = (0.0, -0.3, 0.9, 0.7, 0.5, -0.1)$

(Building, Flower, Mountain, Sea, Sky, Tree)

$\mathbf{d} = (0.8, 0.0, 0.0, 0.6, 0.4, 0.0)$

Figure 4.7 Example of user and item weighted-concept vectors.

The preference value $\text{pref}(\mathbf{d}, \mathbf{u})$ of item \mathbf{d} for user \mathbf{u} is computed with formula 4.2 as follows:

$$\text{pref}(\mathbf{d}, \mathbf{u}) = \frac{(0.8 \times 0.0 + 0.0 \times (-0.3) + 0.0 \times 0.9 + 0.6 \times 0.7 + 0.4 \times 0.5 + 0.0 \times (-0.1))}{\sqrt{0.8^2 + 0.0^2 + 0.0^2 + 0.6^2 + 0.4^2 + 0.0^2} \times \sqrt{0.0^2 + (-0.3)^2 + 0.9^2 + 0.7^2 + 0.5^2 + (-0.1)^2}} \simeq 0.45$$

Personalisation of content retrieval must be handled carefully. An excessive personal bias may drive results too far from the user's current goals. In order to bias the result of a search (the ranking) to the preferences of the user, the above measure could be combined with a query-based score without personalisation, such as the measure $\text{sim}(\mathbf{d}, \mathbf{q})$ defined previously, to produce a combined ranking (Vallet, Castells, Fernández, Mylonas, & Avrithis, 2007). On the other hand, personalisation should combine long-term preferences, based on past usage history, with shorter-term predictions based on current user activities, as well as reactions to (implicit or explicit) user feedback to personalisation output, in order to correct the system assumptions when needed. The incorporation of contextualised semantic preferences into the presented ontology-based personalised recommendation model is indeed the purpose of the work presented in the next section.

4.3 Semantic contextualisation of user preferences

Context is a difficult notion to capture in a software system, and the elements that can be considered under the notion of context are manifold: user tasks/goals, recently browsed/rated items, computing platforms and network conditions, social environment, physical environment and location, time, external events, text around a word, visual content of a graphic region, etc. As representative examples, the reader is referenced to (Billsus & Pazzani, 2000; Middleton, Roure, & Shadbolt, 2004; Sujiyama, Hatano, & Yoshikawa, 2004; Räck, Arbanowski, & Steglich, 2006; Ahn, Brusilovsky, Grady, He, & Syn, 2007).

Complementarily to the ones mentioned, we propose a particular notion useful in semantic content retrieval: that of semantic runtime context, which we define as the background topics \mathbf{C}_u^t under which activities of a user u occur within a given unit of time t . A runtime context is represented in our approach as a set of weighted concepts from the domain ontologies \mathcal{O} . This set is obtained by collecting the concepts that have been involved in the interaction of the user (e.g., accessed items) during a session. Similarly to (Middleton, Shadbolt, & Roure, 2004; Castells, Fernández, Vallet, Mylonas, & Avrithis, 2005), the context is built in such a way that the importance of concepts $c_k \in \mathcal{O}$ fades away with time (number of accesses back when the concept was referenced) by a decay factor $\xi \in [0, 1]$:

$$\mathbf{C}_u^t[c_k] = \xi \cdot \mathbf{C}_u^{t-1}[c_k] + (1 - \xi) \cdot \mathbf{Req}_u^t[c_k],$$

where $\mathbf{Req}_u^t \in [0, 1]^{|\mathcal{O}|}$ is a vector whose components measure the degree in which the concepts c_k are involved in the user's request at time t . This vector can be defined in multiple ways, depending on the application: a query concept-vector (if a request is expressed in term of a concept-based search query), a concept vector containing the most relevant concepts in a document (if a request is a “view document” request), the average concept-vector corresponding to a set of items marked as relevant by the user (if a request is a *relevance feedback* step), etc. The decay factor ξ establishes the number of action units in which a concept is considered as in the current semantic context, i.e., how fast a concept is “forgotten” by the system when recommendations have to be made. This may seem similar to pseudo-relevance feedback. However, it is not used to reformulate a query, but to focus the user's preference vector as follows.

Once the context is built, a contextual activation of preferences is achieved by finding semantic paths linking preferences to context. These paths are made of existing relations between concepts in the ontologies, following the CSA technique explained in Section 4.1. This process can be understood as finding an intersection between user preferences and the semantic context, where the final computed weight

of each concept represents the degree to which it belongs to each set (Figure 4.8). The perceived effect of contextualisation is that user interests that are out of focus, under a given context, are disregarded, and those that are in the semantic scope of the ongoing user activity are more considered for recommendation.

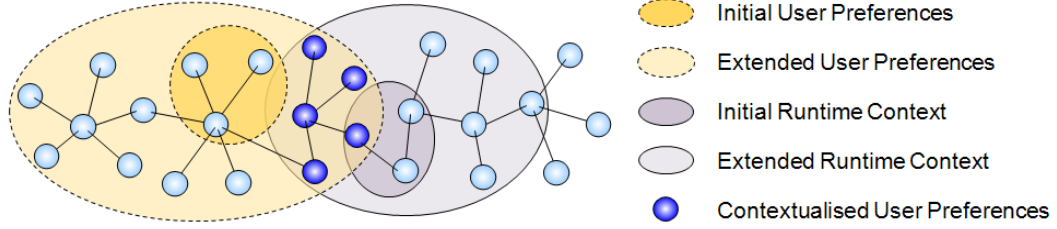


Figure 4.8 Contextualised semantic user preferences.

After the semantic user profile \mathbf{P}_u^t and context \mathbf{C}_u^t are propagated through the ontology relations, a combination of their expanded versions \mathbf{EP}_u^t and \mathbf{EC}_u^t is exploited for making context-aware personalised recommendations using the following expression:

$$\begin{aligned} \text{pref}_c(d, u) &= \lambda \cdot \text{pref}(d, \mathbf{EP}_u^t) + (1 - \lambda) \cdot \text{pref}(d, \mathbf{EC}_u^t) \\ &= \lambda \cdot \cos(\mathbf{d}, \mathbf{EP}_u^t) + (1 - \lambda) \cdot \cos(\mathbf{d}, \mathbf{EC}_u^t) \end{aligned} \quad (4.3)$$

where $\lambda \in [0, 1]$ measures the strength of the personalisation component with respect to the current context. This parameter could be manually established by the user, or dynamically adapted by the system according to multiple factors, such as the current size of the context, the automatic detection of a change in the user's search focus, etc. In the last part of this thesis, we present a recommender system which includes the semantic context-aware recommendation model. In Section 8.4.4, we describe experiments conducted with that recommender system to evaluate the impact of formula 4.3.

4.4 Semantic group profiles for content retrieval

Group-oriented recommendations

During the last few years, a number of domains have been identified in which personalisation has a great potential impact, such as news, education, advertising, tourism or e-commerce. User modelling may encompass large range of personal characteristics. Among them, user interest for topics or concepts (directly observed, or indirectly, via user behaviour monitoring followed by system inference) is one of the most useful in many domains, and widely studied in the user modelling and personalisation research community. While the creation and exploitation of individual models of user preferences and interests have been largely explored in this

field, group modelling – combining individual user models to model a group – has not received the same attention (Ardissono, Goy, Petrone, Segnan, & Torasso, 2003; McCarthy & Anagnost, 1998; O'Connor, Cosley, Konstan, & Riedl, 2001).

It is very often the case that users do not work in isolation. Hence, the proliferation of virtual communities, computer-supported social networks, and collective interaction (e.g., several users in front of a set-top box), call for further research on group modelling, opening new problems and complexities. An increasingly important type of personalised content retrieval and recommender systems comprises those that generate suggestions for groups rather than for individuals. In this context, the decision of a group member whether or not to accept a given recommendation can depend not only on his own evaluation of the content of the recommendation, but also on his beliefs about the evaluations of the other group members, and about their motivation.

Collaborative applications should be able to adapt to groups of people who interact with the system. These groups may be quite heterogeneous, in terms of age, gender, intelligence and personality influence on the perception and complacency with the system outputs each member of the groups may have. Of course, the question that arises is how a system can adapt itself to a group of users, in such a way that each individual enjoys or even benefits from the results.

In this section, we review relevant works on group modelling and recommendation exposed in the literature, and present an approach to group profiling and content retrieval based on merging user preferences contained in individual ontology-based user profiles.

Many studies have examined systems that support **group formation**. The groups can be built intentionally (by explicit definition from the users) or non-intentionally (by automatic identification from the system).

Kansas (Smith, Hixon, & Horan, 1998) is a virtual world in which a user can explicitly join a group by moving towards other users, who share a specific virtual spatial region to work collaboratively in a common task. Inside a group, the users can play different roles according to their current capabilities, which are defined by system treatments of user inputs and outputs. These capabilities can be manually acquired and dropped, or can be transferred by one user to another. The authors explain how direct manipulation and control, the “desktop metaphor”, might be an interesting approach for human computer interaction in cooperative environments.

MusicFX (McCarthy & Anagnost, 1998) enables automatic group formation by selecting music in a corporate gym according to the musical preferences of people working out at a given time. Thus, performing as a group preference arbitration system, *MusicFX* allows users to influence, but not directly control, the selection of music in the fitness centre. Specifically, each user specifies his preference for each musical genre. An individual preference rating for a genre is presented by a number

ranging from -2 to $+2$. The group preference for that genre is then computed by the sum of the current users' individual preferences. The system uses a weighted random selection policy for selecting one of the group top N music genres. One interesting anecdote the authors found with the system was the fact that people began modifying their workout times to arrive at the gym with other people, often strangers, who shared their music tastes.

The group modelling problem has also been addressed *merging similar individual user profiles*. In this scenario, user profiles are represented as sets of weighted preferences or as sets of personal scores assigned by the users to the existing items.

INTRIGUE (Ardissono, Goy, Petrone, Segnan, & Torasso, 2003) is a tourist information server that presents information about the area around Torino (Italy). The system recommends sightseeing destinations and itineraries by taking into account the preferences of heterogeneous tourist groups, explains the recommendations by addressing the group members' requirements, and provides an interactive agenda for scheduling a tour. For each individual attraction, a record in a database stores characteristics and properties as a set of feature/value pairs, some of them related to geographical information and others used for matching preferences and interests of the users. Group recommendations are conducted in three steps. Firstly, the group is modelled as a set partitioned into a number of homogeneous subgroups, whose members have similar characteristics and preferences, and are assigned different degrees of influence on the estimation of the group preferences. Next, items are separately ranked by taking the preferences of each subgroup into account. Finally, subgroup-related rankings are merged to obtain the ranking suitable for the whole group.

In (Masthoff, 2004), the author discusses several strategies based on social choice theory for merging individual user models to adapt to groups (e.g., plurality voting, additive and multiplicative utilities, "Borda count" and "Copeland rule", approval voting, least misery and most pleasure strategies, etc.). Considering a list of TV programs, a group of viewers represent their interests with sets of personal 1-10 rating for the different TV programs. Masthoff investigates how humans select a sequence of items for the group to watch, how satisfied people believe they would be with the sequence chosen by the different strategies, and how their satisfactions correspond with that predicted by a number of satisfaction functions. These evaluation functions are modified in (Masthoff, 2005), where satisfaction is modelled as a mood, and assimilation and decline of emotions with time is incorporated.

A more sophisticated strategy to merge various individual user profiles based on total distance minimisation is presented in (Yu, Zhou, Hao, & Gu, 2004). The minimisation of the total distance between user profiles guarantees that the merged result could be close to most users' preferences. The shown experimental results

prove that the resultant group profile actually reflects most members' preferences of the group. The practical application and evaluation of the above strategy is described in (Yu, Zhou, Hao, & Gu, 2006), where a TV program recommender system for multiple viewers is presented.

In addition to group modelling, there exist several approaches that have been applied to the problem of making accurate and efficient recommendations for groups of people under the framework of **collaborative filtering**. In collaborative filtering systems, a user provides ratings to items, and these ratings are used to suggest him ranked lists with other items according to the overall preferences of those people with similar rating patterns.

In (Hill, Stead, Rosenstein, & Furnas, 1995), a video recommender system is presented. Under a client/server architecture, the system receives and sends emails to obtain user ratings and to provide video suggestions. The recommendations are shown to the users sorted by predicted ratings and classified by video categories. The system also provides ranked lists from the most similar users, giving thus recommendations to a group of users (virtual community), instead of to individual users. The authors obtained open ended feedback from users indicating interest in establishing direct social contacts within their virtual community.

PolyLens (O'Connor, Cosley, Konstan, & Riedl, 2001) is a collaborative filtering system that suggests movies to groups of people with similar interests, which are expressed through personal five-start scale ratings from the well-known *MovieLens* recommender system (Herlocker, Konstan, & Riedl, 2000). In *PolyLens*, groups of people are explicitly created by users. For each member of a group, a ranked list of movies is obtained from a classic collaborative filtering mechanism. The individual ranked lists are merged according to the least misery principle, i.e., using a social value function where the group's happiness is the minimum of the individual members' happiness scores. Experimenting with *PolyLens*, the authors analysed primary design issues for group recommenders, such as the nature of the groups (in terms of persistency and privacy), the rights of group members, the social value functions for groups, and the interfaces for displaying group recommendations. They found that users not only valued group recommendations, but also were willing to yield some privacy to get the benefits of such recommendations, and extend the recommender system to enable them to invite non-members to participate, via email.

Finally, instead of applying an automatic group modelling algorithm, there exist approaches that make use of **consensus mechanisms** to achieve a final content recommendation policy agreed by the different members of a group.

Travel Decision Forum (Jameson, Baldes, & Kleinbauer, 2003), *TDF*, proposes a manual user interest aggregation method for group modelling by 1) allowing the current member optionally to view (and perhaps copy) the preferences already specified by other members, and 2) mediating user negotiations offering the users

proposals and adaptations of their preferences. This method has several advantages, such as saving of effort, learning from other members, and encouraging assimilation to facilitate the reaching of agreement. In this system, neither user profile merging nor recommendation is used.

Collaborative Advisory Travel System (McCarthy, Salamo, McGinty, & Smyth, 2006), *CATS*, is a cooperative group travel recommender system which aims to help a group of users arrive at a consensus when they need to plan skiing holidays together; each having their own needs and preferences with respect to what constitutes as an ideal holiday for them. *CATS* system makes use of visual cues to create emphasis and help users locate relevant information, as well as enhance group awareness of each other's preferences and motivational orientations. Individual user models are defined as set of critiques, i.e., restrictions on vacation features that should be satisfied. The system constructs a reliable group-preference model measuring the quality of each vacation package in terms of its compatibility with the restrictions declared by the members of the group.

Table 4.2 shows a summary of the group recommendation approaches explained in this section, giving brief descriptions and the referenced representative examples of all of them.

Approach	Description	Representative examples
<i>Group formation</i>	Explicit or implicit group modelling to achieve a democratic content retrieval according to individual preferences.	(Smith, Hixon, & Horan, 1998) (McCarthy & Anagnost, 1998)
<i>User profile merging</i>	Merging of individual preferences to obtain a unique group profile to be used in the content retrieval process.	(Ardissono, Goy, Petrone, Segnan, & Torasso, 2003) (Masthoff, 2004) (Yu, Zhou, Hao, & Gu, 2006)
<i>Collaborative filtering</i>	Application of collaborative strategies to retrieve contents which are novel for the user, but related to him based on the preferences of similar users, and combination of the resultant individual recommendations.	(Hill, Stead, Rosenstein, & Furnas, 1995) (O'Connor, Cosley, Konstan, & Riedl, 2001)
<i>Cooperative consensus</i>	Application of a consensus mechanism by users in order to cooperatively define a shared content retrieval policy.	(Jameson, Baldes, & Kleinbauer, 2003) (McCarthy, Salamo, McGinty, & Smyth, 2006)

Table 4.2 Categorisation of group recommendation approaches and examples.

Social choice strategies

Though the previous approaches have addressed group preference modelling explicitly to a rather limited extent, or in an indirect way in prior work in the

computing field, the related issue of *social choice* (also called group decision making, i.e., deciding what is best for a group given the opinions of individuals) has been studied extensively in economics, politics, sociology, and mathematics (Pattanaik, 2001; Taylor, Mathematics and Politics: Strategy, Voting, Power and Proof, 1995). The models for the construction of a social welfare function in these works are similar to the group modelling problem we put forward here.

Other areas in which social choice theory has been studied are meta-search, collaborative filtering, and multi-agent systems. In meta-search, the ranking lists produced by multiple search engines need to be combined into one single list, forming the well-known problem of rank aggregation in IR (Baeza-Yates & Ribeiro Neto, 1999). In CF, preferences of a group of individuals have to be aggregated to produce a predicted preference for somebody outside the group. In multi-agent systems, agents need to take decisions that are not only rational from an individual's point of view, but also from a social point of view.

In this work, we study the feasibility of applying strategies, based on social choice theory (Masthoff, 2004), for combining multiple individual preferences in a personalisation framework from a knowledge-based multimedia retrieval system (Vallet, Castells, Fernández, Mylonas, & Avrithis, 2007) which makes use of the ontology-based knowledge representation explained at the beginning of this chapter. In the framework, user preferences are gathered in ontology semantic concept-based user profiles. Using these profiles, the framework retrieves personalised ranked lists of items, and shows them in a graphical interface (Figure 4.9).

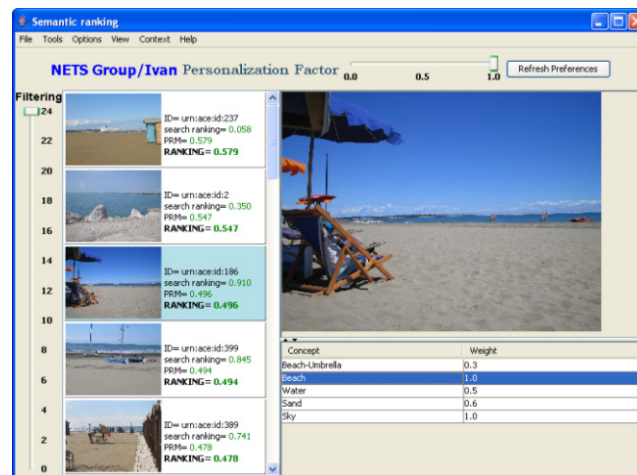


Figure 4.9 Screenshot of the personalisation framework used to evaluate ontology-based group modelling strategies.

In the following, we explain the investigated strategies. We assume a user has a preference (utility) for each item represented in the form of a numeric rating. In all the cases, the greater the rating value, the most useful the item is for the user.

- **Additive utilitarian strategy.** Preference values from all the users of the group are added, and the larger the sum the more influential the item is for the group (Figure 4.10). Note that the resulting group ranking will be exactly the same as that obtained taking the average of the individual preference values. A potential problem of this strategy is that individuals' opinions tend to be less significant as larger the group is.

This strategy could also use a weighted schema, where a weight is attached to individual preferences depending on multiple criteria for single or multiple users. For example, in *INTRIGUE* (Ardissono, Goy, Petrone, Segnan, & Torasso, 2003), weights are assigned depending on the number of people in a group and the group's relevance (children and disabled have a higher relevance).

	Item									
User	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
u_1	10	4	3	6	10	9	6	8	10	8
u_2	1	9	8	9	7	9	6	9	3	8
u_3	10	5	2	7	9	8	5	6	7	6
group	21	18	13	22	26	26	17	23	20	22

Figure 4.10 Group formation following the additive utilitarian strategy. The ranked list of items for the group would be (d_5 - d_6 , d_8 , d_4 - d_{10} , d_1 , d_9 , d_2 , d_7 , d_3).

- **Multiplicative utilitarian strategy.** Instead of adding the preference ratings, they are multiplied, and the larger the product the more influential the item is for the group (Figure 4.11).

This strategy could be self-defeating: in a small group, the opinion of each individual will have too much large impact on the product.

	Item									
User	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
u_1	10	4	3	6	10	9	6	8	10	8
u_2	1	9	8	9	7	9	6	9	3	8
u_3	10	5	2	7	9	8	5	6	7	6
group	100	180	48	378	630	648	180	432	210	384

Figure 4.11 Group formation following the multiplicative utilitarian strategy. The ranked list of items for the group would be (d_6 , d_5 , d_8 , d_{10} , d_4 , d_9 , d_2 - d_8 , d_1 , d_3).

- **Borda count strategy** (Borda, 1781). Scores are assigned to the items according to their ratings in a user profile: those with the lowest value get zero scores, the next one up one point, and so on. When an individual has multiple preferences with the same rating, the averaged sum of their hypothetical scores are equally distributed to the involved items. With the obtained scores, an additive strategy is followed, and the larger the sum the more influential the item is for the group.

Figure 4.12 shows an example of the two steps followed by Borda count strategy. In the first step, ratings are normalised according to their relative relevance within the users' preferences. The items with the three lowest ratings for user u_1 are coloured in the tables. For the first one (in increasing rating value), d_3 , a zero score is assigned. The second one, d_2 , receives a score of value 1. The next score to be assigned would be 2. In this case, the next two items with lowest rating value, d_4 and d_7 , have the same rating. In this case, two scores (2 and 3) are considered, and the average of them, i.e., $(2+3)/2=2.5$, is assigned to both of the items.

User	Item									
	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
u_1	10	4	3	6	10	9	6	8	10	8
u_2	1	9	8	9	7	9	6	9	3	8
u_3	10	5	2	7	9	8	5	6	7	6

↓

User	Item									
	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
u_1	8	1	0	2.5	8	6	2.5	4.5	8	4.5
u_2	0	7.5	4.5	7.5	3	7.5	2	7.5	1	4.5
u_3	9	1.5	0	5.5	8	7	1.5	3.5	5.5	3.5
group	17	10	4.5	15.5	19	20.5	6	15.5	14.5	12.5

Figure 4.12 Group formation following the Borda count strategy. The ranked list of items for the group would be (d_6, d_5, d_1, d_4 - $d_8, d_9, d_{10}, d_2, d_7, d_3$).

- **Copeland rule strategy** (Copeland, 1951). Being a form of majority voting, this strategy sorts the items according to their *Copeland index*: the difference between the number of times an item beats (has higher ratings) the rest of the items and the number of times it loses.

Figure 4.13 shows an example of Copeland rule strategy. In the bottom table, a $+/ -$ symbol in the ij -th cell (i for rows, and j for columns) means that item at j -th column was rated higher/lower than item at i -th row by the majority of the users. A zero value in a cell means that the corresponding items were rated with the same number of “beats” and “looses”.

User	Item									
	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
u_1	10	4	3	6	10	9	6	8	10	8
u_2	1	9	8	9	7	9	6	9	3	8
u_3	10	5	2	7	9	8	5	6	7	6

↓

Item	Item									
	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
d_1	0	-	-	-	0	-	-	-	0	-
d_2	+	0	-	+	+	+	0	+	+	+
d_3	+	+	0	+	+	+	+	+	+	+
d_4	+	-	-	0	+	+	-	0	0	-
d_5	0	-	-	-	0	-	-	-	-	-
d_6	+	-	-	-	+	0	-	-	-	-
d_7	+	0	-	+	+	+	0	+	+	+
d_8	+	-	-	0	+	+	-	0	+	-
d_9	0	-	-	0	+	+	-	-	0	-
d_{10}	+	-	-	+	+	+	-	+	+	0
group	+7	-6	-9	+1	+8	+5	-6	0	+3	-3

Figure 4.13 Group formation following the Copeland rule strategy. The ranked list of items for the group would be $(d_5, d_1, d_6, d_9, d_4, d_8, d_{10}, d_2, d_7, d_3)$.

- **Approval voting strategy.** A threshold is considered for the item ratings: only those ratings greater or equal than the threshold value are taking into account for the profile combination. An item receives a vote for each user profile that has its rating surpassing the established threshold. The larger the number of votes the more influential the item is for the group (Figure 4.14).

This strategy intends to promote the election of moderate alternatives: those that are not strongly disliked.

User	Item									
	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
u_1	10	4	3	6	10	9	6	8	10	8
u_2	1	9	8	9	7	9	6	9	3	8
u_3	10	5	2	7	9	8	5	6	7	6

↓ $threshold = 5$

User	Item									
	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
u_1	1			1	1	1	1	1	1	1
u_2		1	1	1	1	1	1	1		1
u_3	1			1	1	1		1	1	1
group	2	1	1	3	3	3	2	3	2	3

Figure 4.14 Group formation following the approval voting strategy. The ranked list of items for the group would be (d_4 - d_5 - d_6 - d_8 - d_{10} , d_1 - d_7 - d_9 , d_2 - d_3).

- **Least misery strategy.** The score of an item in the group profile is the minimum of its ratings in the user profiles. The lower rating the less influential the item is for the group. Thus, a group is as satisfied as its least satisfied member (Figure 4.15). *PolyLens* (O'Connor, Cosley, Konstan, & Riedl, 2001) uses this strategy, assuming a group of people going to watch a movie together tends to be small, and the group is as happy as its least happy member.

Note that a minority of the group could dictate the opinion of the group: although many members like a certain item, if one member really hates it, the preferences associated to it will not appear in the group profile.

User	Item									
	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
u_1	10	4	3	6	10	9	6	8	10	8
u_2	1	9	8	9	7	9	6	9	3	8
u_3	10	5	2	7	9	8	5	6	7	6
group	1	4	2	6	7	8	5	6	3	6

Figure 4.15 Group formation following the least misery strategy. The ranked list of items for the group would be (d_6 , d_5 , d_4 - d_8 - d_{10} , d_7 , d_2 , d_9 , d_3 , d_1).

- **Most pleasure strategy.** It works as the least misery strategy, but instead of considering for an item the smallest ratings of the users, it selects the greatest ones. The higher rating the more influential the item is for the group (Figure 4.16).

	Item									
User	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
u_1	10	4	3	6	10	9	6	8	10	8
u_2	1	9	8	9	7	9	6	9	3	8
u_3	10	5	2	7	9	8	5	6	7	6
group	10	9	8	9	10	9	6	9	10	8

Figure 4.16 Group formation following the least misery strategy. The ranked list of items for the group would be (d_1 - d_5 - d_9 , d_2 - d_4 - d_6 - d_8 , d_3 - d_{10} , d_7).

- **Average without misery strategy.** As the additive utilitarian strategy, this one assigns an item the average of its ratings in the individual profiles. The difference here is that those items which have a rating under a certain threshold will not be considered in the group recommendations. Figure 4.17 shows an example of group formation following this strategy with a threshold value of 3.

	Item									
User	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
u_1	10	4	3	6	10	9	6	8	10	8
u_2	1	9	8	9	7	9	6	9	3	8
u_3	10	5	2	7	9	8	5	6	7	6
group	-	18	-	22	26	26	17	23	-	22

Figure 4.17 Group formation following the average without misery strategy. The ranked list of items for the group would be (d_6 - d_5 , d_8 , d_4 - d_{10} , d_2 , d_7).

- **Fairness strategy.** In this strategy, the items that were rated highest and cause less misery to all the users of the group are combined as follows. A user is randomly selected. His L top rated items are taking into account. From them, the item that less misery causes to the group (that from the worst alternatives that has the highest rating) is chosen for the group profile with a score equal to N , i.e., the number of items. The process continues in the same way considering the remaining $N-1$, $N-2$, etc. items and uniformly diminishing to 1 the further assigned scores. In the final list, the higher score

the more influential the item is for the group. Note that this list would be different if we let other users to choose first.

To better understand the strategy, let us explain its first step on the example shown in Figure 4.18. Suppose we start with user u_1 , whose top ranked items are d_1 , d_5 and d_9 . From these items, we choose item d_5 because it is the one that less misery causes to users u_2 and u_3 , whose lowest ratings for items d_1 , d_5 and d_9 are respectively 1, 7 and 3. We assign item d_5 a score of 10.

	Item									
User	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
u_1	10	4	3	6	10	9	6	8	10	8
u_2	1	9	8	9	7	9	6	9	3	8
u_3	10	5	2	7	9	8	5	6	7	6
group	4	3	1	8	10	9	5	7	2	6

Figure 4.18 Group formation following the fairness strategy. The ranked list of items for the group could be (d_5 , d_6 , d_4 , d_8 , d_{10} , d_7 , d_1 , d_2 , d_9 , d_3), following the user selecting order u_1 , u_2 and u_3 , and setting $L=3$.

- **Plurality voting strategy.** This method follows the same idea of the fairness strategy, but instead of selecting from the L top preferences the one that least misery causes to the group, it chooses the alternative which most votes have obtained.

Figure 4.19 shows an example of the group formation obtained with the plurality voting strategy. The item ratings involved in the first step of the algorithm are coloured.

	Item									
User	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
u_1	10	4	3	6	10	9	6	8	10	8
u_2	1	9	8	9	7	9	6	9	3	8
u_3	10	5	2	7	9	8	5	6	7	6
group	5	3	1	8	10	9	2	7	4	6

Figure 4.19 Group formation following the plurality voting strategy. The ranked list of items for the group could be (d_5 , d_6 , d_4 , d_8 , d_{10} , d_1 , d_9 , d_2 , d_7 , d_3), following the user selecting order u_1 , u_2 and u_3 , and setting $L=3$.

Ontology-based group profiles

In our proposal, because of we explore the combination of ontology-based user profiles, instead of rating lists, we have to slightly modify the original strategies described previously. As explained in Sections 4.1, 4.2 and 4.3, user preferences belong to the range $[-1, +1]$, and the presented personalised and context-aware recommendation models are built based on that premise. For this reason, if we want to apply the same models to group profiles, the latter also have to maintain preference values in $[-1, +1]$. The following are comments about changes and considerations we have taken into consideration to apply social choice strategies for the creation of ontology-based group profiles.

- In the **additive utilitarian strategy**, preference weights are added and averaged by the number of users, so the final group preferences also belong to the range $[-1, +1]$.
- In the **multiplicative utilitarian strategy**, it is advisable not to have null weights in individual profiles because we would discard valued preferences when the group profile is built. So, if this situation happens, we change the null weight values to very small ones (e.g., 10^{-3}).
- In the **Borda count strategy**, the final scores are uniformly normalised to the range $[-1, +1]$.
- In the **Copeland rule strategy**, the final scores are uniformly normalised to the range $[-1, +1]$.
- In the **approval voting strategy**, the final scores are uniformly normalised to the range $[-1, +1]$, and a threshold of 0.5 is considered.
- In the **least misery strategy**, no changes have to be made.
- In the **most pleasure strategy**, no changes have to be made.
- In the **average without misery strategy**, the final scores are uniformly normalised to the range $[-1, +1]$, and a threshold of 0.25 is considered.
- In the **fairness strategy**, at each iteration we decided to select the $L=R/2$ top rated items of the selected user, where R is the number of preferences not assigned to the group profile yet. The final scores are uniformly normalised to the range $[-1, +1]$.
- In the **plurality voting**, at each iteration we decided to select the $L=R/2$ top rated items of the selected user, where R is the number of preferences not assigned to the group profile yet. The final scores are uniformly normalised to the range $[-1, +1]$.

The above modifications have been used in a set of experiments explained in Section 6.1. Here we do not provide empirical results, and we simply describe how we propose to apply the group modelling strategies within our ontology-based content retrieval framework. Basically, we identify two different approaches: 1) the combination of individual preferences of the members of the group, and 2) the combination of the ranked item lists obtained from recommendations obtained from personal profiles.

The first one (Figure 4.20), which we call *profile combination method*, merges individual user profiles to form a common user profile and generate common recommendation according to this new profile. In this method, the computation of the recommendations is done according to only one user profile. However, if the individual user profiles have a large number of preferences, the recommendation process might not be as fast as expected.

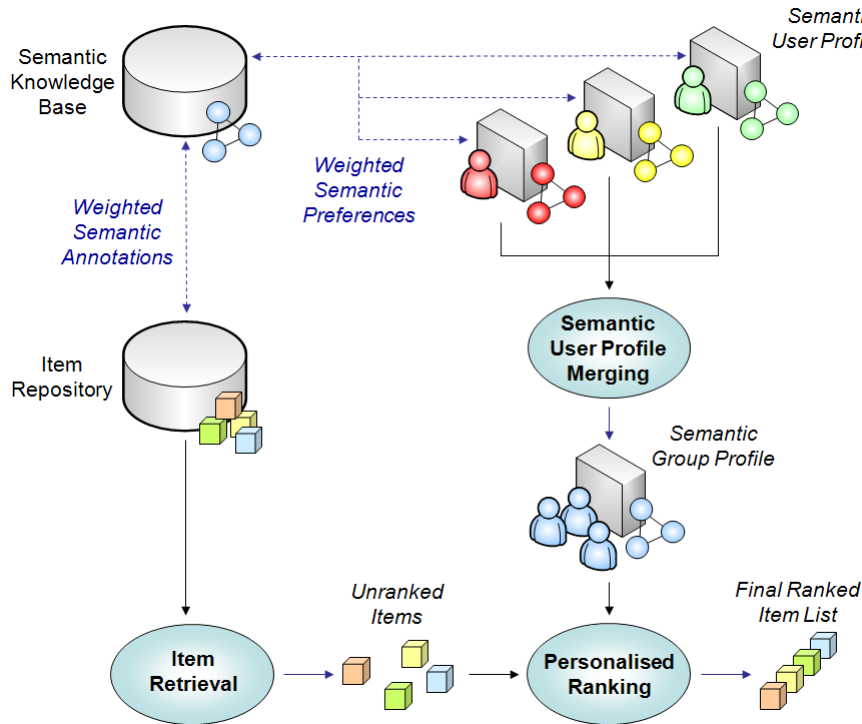


Figure 4.20 Group recommendations by the combination of ontology-based user profiles.

The second approach (Figure 4.21) on the other hand extracts individual user rankings according to individual user profiles, and aggregates them using specific criteria at a later stage. We refer to it as the *ranking combination method*. In this method, the computation of recommendation is done for each user profile. Moreover, if the sizes of the item ranked lists are large, the group modelling strategies would be also run slowly. For these reasons, in most cases, this second method should be much slower than the profile combination one.

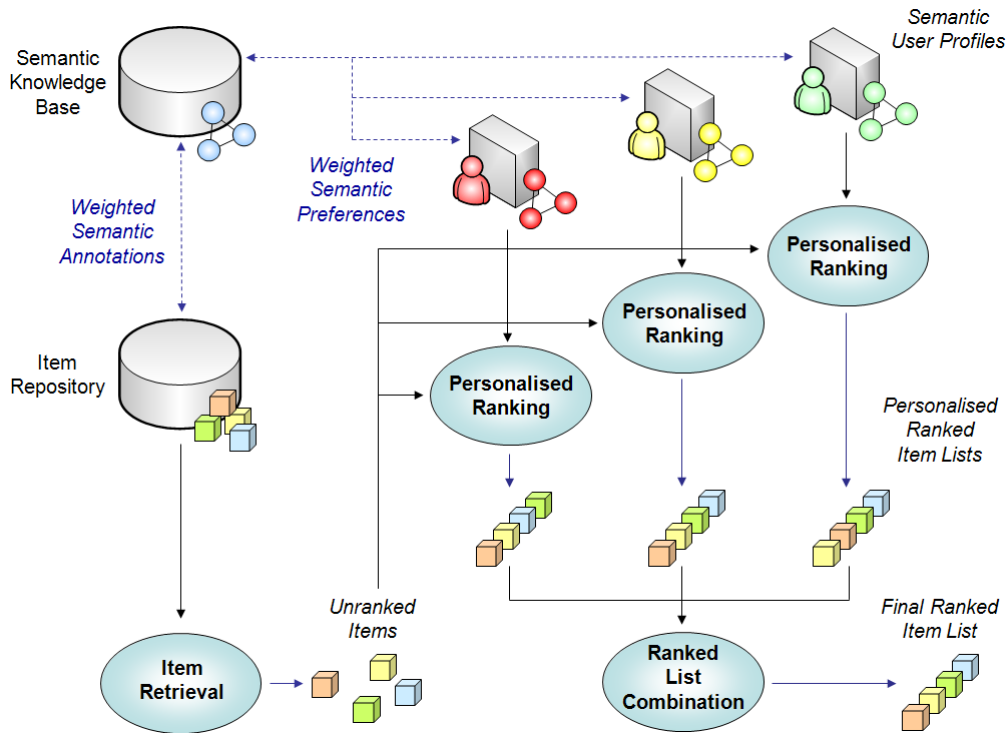


Figure 4.21 Group recommendations by the combination of personalised ranked item lists.

At first sight it is not clear which method is going to better perform group profiling within a recommender system. This and other aspects, such as an optimal group modelling strategy, will be investigated in the experiments described in Chapter 6.

4.5 Summary

The definition and exploitation of the underlying semantic layer between user and item spaces might be very useful to overcome some of the current shortcomings of content-based recommender systems. In this chapter, we have proposed an ontology-based representation of such layer, where user preferences and item features are described in the form of weighted ontology concepts (classes or instances), and are expanded to other concepts applying a spreading activation mechanism through the semantic relations available in the considered domain ontologies.

The proposed ontology-based knowledge representation has not only allowed us to enrich the user and item descriptions, hopefully mitigating the effects of the sparsity problem, but also has permitted the definition of two flexible models that provide semantic context-aware and group-oriented recommendations. The evaluation of these models is postponed to Chapter 6.

Chapter 5

Hybrid recommendation: a semantic multilayer approach

Content-based recommender systems suggest to users items that do have content features expressed in the user profiles. This characteristic is essential to obtain accurate results in applications where personalised content retrieval tasks have to be performed. However, it does not provide the opportunity of suggesting items that may be relevant to the users taking into account social aspects, such as item popularity and interest-based user relations, which are the basis of any collaborative system.

Here we take the step of exploiting the proposed ontology-based knowledge representation in the implementation of hybrid recommendation models, which establish user relations according to semantic content-based similarities between user and item profiles. This idea is achieved by analysing the structure of the domain ontologies, the weighted links between users and concepts (as defined by preferences), the links between concepts and contents (annotations), and the links (explicit ratings) between content and users. Based on this rich interrelation within and across the three spaces (users, concepts, content), we develop strategies of coordinated clustering to produce focused recommendations based on partial but cohesive similarities. Our approach finds groups of interests shared by users, and Communities of Interest (CoI) among users. Users who share interests of a specific concept cluster are connected in the corresponding community, where their preference weights determine the degree of membership to that cluster. This enables focused recommendations layered in the different communities.

This chapter is organised as follows. Section 5.1 summarises past works on CoI identification, and social collaborative filtering that are relevant for our proposal. Section 5.2 describes the proposed clustering technique to build the multi-layer relations between users. Section 5.3 explains the exploitation of the derived CoI to define our semantic content-based collaborative filtering approach. Finally, Section 5.4 presents a simple example where the technique is tested.

5.1 Communities of interest

During the last few years, the rapid development, spread and convergence of information and communication technologies, and their support infrastructures, which are reaching all aspects of businesses and homes in our everyday lives, are giving rise to new and unforeseen ways of inter-personal connection, communication and collaboration. Virtual communities, computer-supported social networks, and collective interaction support technologies are starting to proliferate in increasingly sophisticated ways, opening new research opportunities on social group analysis, modelling and exploitation.

In this scenario, Communities of Practice (CoP) have been defined as groups of people who get involved in a process of collective work in a shared domain of human endeavour (Wenger, 1998): a community of scientists investigating a specific problem, a group of engineers working on similar projects, a clique of students having a discussion about a common subject, etc. These people collaborate over a period of time, sharing ideas and experiences in order to find solutions and build innovations for a particular practice.

However, it is very often the case that the membership to a community is unknown or unconscious. In many social applications, a person describes his interests and knowledge in a personal profile to find people with similar ones, but he is not aware of the existence of other (directly or indirectly) related interests and knowledge that might be useful to find those people. Furthermore, depending on the context of application or situation, a user can be interested in different topics and groups of people. In both cases, a strategy to automatically identify CoP might be very beneficial (Alani, O'Hara, & Shadbolt, 2002).

The issue of finding hidden links between users based on the similarity of their preferences or historic behaviour is not a new idea. In fact, this is the essence of the well-known collaborative filtering systems, where items are recommended to a specific user based on his shared interests with other users, or according to opinions, comparatives, and ratings of items given by similar users. However, in typical approaches, the comparison between users and items is done globally, in such a way that partial, but strong and useful similarities might be missed. For instance, two people may have a highly coincident taste in *cinema*, but a very divergent one in *sports*. The opinions of these people on *movies* could be highly valuable for each other, but risk to be ignored by many recommender systems, because the global similarity between the users might be low.

Communities of Interest (CoI) are a particular case of CoP, and have been defined as a group of people who share a common interest or passion. They exchange ideas and thoughts about the given passion, creating a self-organising commune where they come back frequently and remain for extended periods. In this

chapter, we propose a novel approach towards building emerging multilayered CoI by analysing the individual motivations and preferences of users, described in ontology-based user profiles, and broken into potentially different areas of personal interest. Like in previous approaches (Liu, Maes, & Davenport, 2006), our method builds and compares profiles of user interests for semantic topics and specific concepts in order to find similarities among users. But in contrast to prior work, we divide the user profiles into clusters of cohesive interests, and based on this, several layers of CoI are found. This provides a richer model of interpersonal links, which better represents the way people find common interests in real life.

Our approach is based on the ontological representation of the domain of discourse where user interests are defined, which was presented in Section 4.1. The ontological space takes the shape of a semantic network of interrelated domain concepts, and the user profiles are initially described as weighted lists measuring the user interests for those concepts. Taking advantage of the relations between concepts, and the (weighted) preferences of users for the concepts, our strategy clusters the semantic space based on the correlation of concepts appearing in the preferences of individual users. After this, user profiles are partitioned by projecting the concept clusters into the set of preferences of each user. Then, users can be compared on the basis of the resulting subsets of interests, in such a way that several, rather than just one, (weighted) links can be found between two users.

The identified multilayered CoI are potentially useful for many purposes. For instance, users may share preferences, items, knowledge, and benefit from each other's experience in focused or specialised conceptual areas, even if they have very different profiles as a whole. Such semantic subareas need not be defined manually, as they emerge automatically with our proposed method. Users may be recommended items or direct contacts with other users for different aspects of day-to-day life.

In recommendation environments, there is an underlying need to distinguish different layers within the interests and goals of the users. Depending on the current context, only a specific subset of the segments (layers) of a user profile should be considered in order to establish his similarities with other people when a recommendation has to be performed. Models of CoI partitioned at different common semantic layers can enable more accurate and context-sensitive results in recommender processes. Thus, as an applicative development of our automatic semantic clustering and CoI building methods, in the next sections, we propose and test empirically several content-based collaborative filtering models that retrieve information items according to a number of real user profiles and within different contexts.

5.2 Semantic multilayered communities of interest

In social communities, it has been found that people who are known to share a specific interest are likely to have additional connected interests (Liu, Maes, & Davenport, 2006). For instance, people who share interests in travelling might be also keen on topics related in photography, gastronomy or languages. In fact, this assumption is the basis of many recommender system technologies. We assume this hypothesis here as well, in order to cluster the concept space in groups of preferences shared by several users.

We propose to exploit the links between users and concepts to extract relations among users and derive semantic communities of interest according to common preferences. Analysing the structure of the domain ontology, and taking into account the semantic preference weights of the user profiles we shall cluster the domain concept space generating groups of interests shared by several users. Thus, those users who share interests of a specific concept cluster will be connected in the community, and their preference weights will measure their degree of membership to each cluster.

Specifically, a vector $\mathbf{c}_k = (c_{k,1}, c_{k,2}, \dots, c_{k,M})$ is assigned to each ontology concept c_k present in the preferences of at least one user, where $c_{k,m} = u_{m,k}$ is the weight of concept c_k in the semantic profile of user u_m . Based on these vectors, a classic hierarchical clustering strategy (Duda, Hart, & Stork, 2001) is applied. The obtained clusters (Figure 5.1) represent the groups of preferences (topics of interests) in the concept-user vector space shared by a significant number of users.

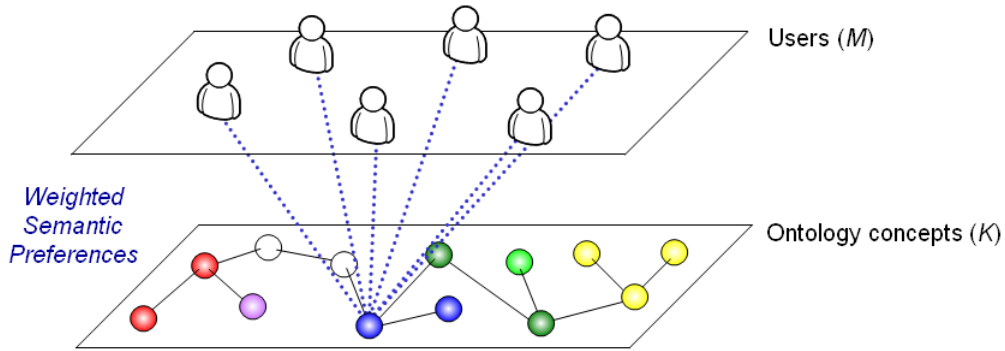


Figure 5.1 Semantic concept clustering based on shared interests of the users.

Once the concept clusters are created, each user can be assigned to a specific cluster. The similarity between a user's preferences $\mathbf{u}_m = (u_{m,1}, u_{m,2}, \dots, u_{m,K})$ and a cluster \mathcal{C}_q is computed by:

$$\text{sim}(u_m, \mathcal{C}_q) = \frac{\sum_{c_k \in \mathcal{C}_q} u_{m,k}}{|\mathcal{C}_q|} \quad (5.1)$$

where c_k represents the concept that corresponds to the $u_{m,k}$ component of the user preference vector, and $|\mathcal{C}_q|$ is the number of concepts included in the cluster. The clusters with highest similarities might be then assigned to the users, thus creating groups of users with shared interests (Figure 5.2).

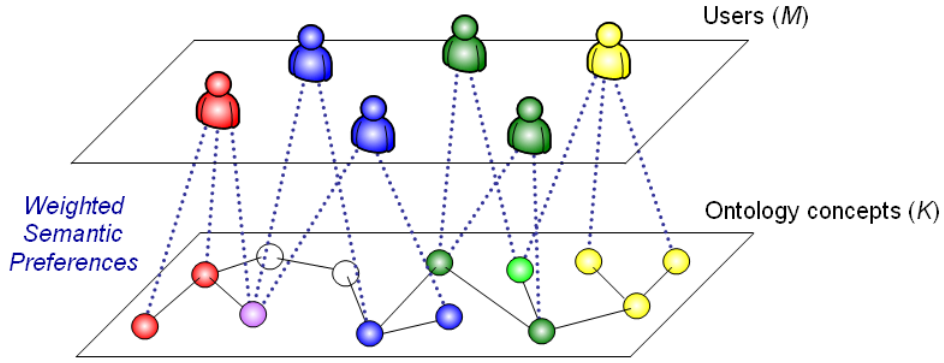


Figure 5.2 Groups of users obtained from shared semantic concept clusters.

Furthermore, the concept and user clusters can be used to find emergent, focused semantic Communities of Interest (CoI). The preference weights of the user profiles, the degrees of membership of the users to each cluster, and the similarity measures between clusters are used to find relations between two distinct types of social items: individuals and groups of individuals.

Taking into account the concept clusters, user profiles are partitioned into semantic segments. Each of these segments corresponds to a concept cluster, and represents a subset of the user interests that is shared by the users who contributed to the clustering process. By thus introducing further structure in user profiles, it is now possible to define relations among users at different levels, obtaining a multilayered network of users. Figure 5.3 illustrates this idea. The image on the left represents a situation where four user clusters are obtained. Based on them (images on the right), user profiles are partitioned in four semantic layers. On each layer, weighted relations among users are derived, building up different semantic communities of interest.

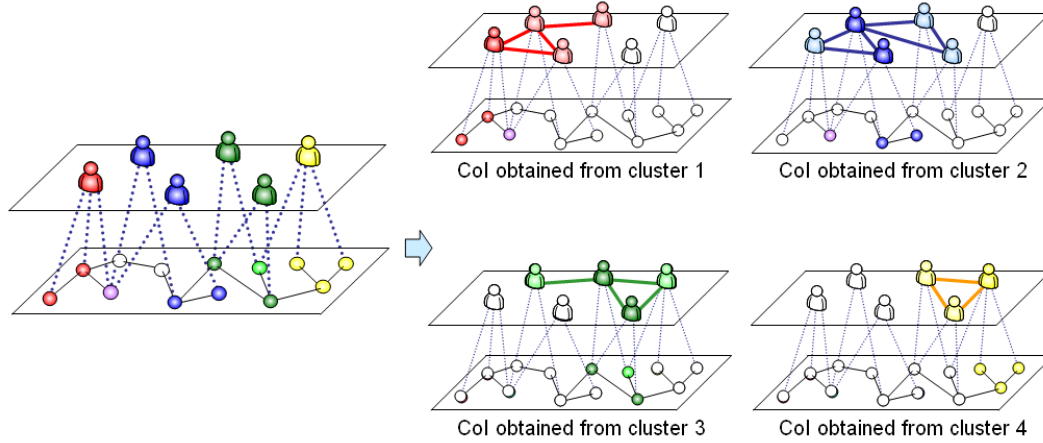


Figure 5.3 Multilayered CoI built from shared semantic concept clusters.

The resulting semantic CoI have many potential applications. For example, they can be exploited to the benefit of content-based collaborative recommendations, not only because they establish similarities between users, but also because they provide powerful means to focus on different semantic contexts for different information needs. The design of content retrieval models in this direction is explored in the next section. Additionally, the identified user clusters could be utilised by group profile modelling strategies as those explained in Section 4.4.

5.3 Semantic hybrid recommendation models

Collaborative filtering applications adapt to groups of people who interact with the system, in a way that single users benefit from the experience of other users with which they have certain traits or interests in common. User groups may be quite heterogeneous, and it might be very difficult to define the mechanisms for which the system adapts itself to the groups of users, in such a way that each individual enjoys or even benefits from the results. Furthermore, once the user association rules are defined, an efficient search for closest neighbours among a large user population of potential neighbours has to be addressed. This is the great bottleneck in conventional user-based collaborative filtering algorithms. Item-based algorithms attempt to avoid these difficulties by exploring the relations among items, rather than the relations among users. However, the item neighbourhood is fairly static and do not allow to easily apply personalised recommendations or inference mechanisms to discover potential hidden user interests. We claim that exploiting the relations of the underlying CoI which emerge from the users' interests, and combining them with semantic item preference information can have an important benefit in collaborative filtering approaches.

Using our semantic multilayered CoI proposal explained in the previous section, we present here two recommendation models that generate ranked lists of items in different scenarios taking into account the obtained links between users. The first model (that we shall label as UP) is based on the semantic profile of the user to whom the ranked list is delivered. This model represents the situation where the interests of a user are compared to other interests in a social network. The second model (labelled NUP) outputs ranked lists disregarding the user profile. This can be applied in situations where a new user does not have a profile yet, or when the general preferences in a user's profile are too generic for a specific context, and do not help to guide the user towards a very particular, context-specific need. Additionally, we consider two versions for each model: a) one that generates a unique ranked list based on the similarities between the items and all the existing semantic clusters, and, b) one that provides a ranking for each semantic cluster. Thus, we shall study four different retrieval strategies, UP (profile-based), UP- q (profile-based, considering a specific cluster \mathcal{C}_q), NUP (no profile), and NUP- q (no profile, considering a specific cluster \mathcal{C}_q). The four strategies are formalised next. In the following, for a user profile \mathbf{u}_m , an information object vector \mathbf{d}_n , and a cluster \mathcal{C}_q , we denote by \mathbf{u}_m^q and \mathbf{d}_n^q the projections of the corresponding concept vectors onto cluster \mathcal{C}_q , i.e., the k -th components of \mathbf{u}_m^q and \mathbf{d}_n^q are $u_{m,k}$ and $d_{n,k}$ respectively if $c_k \in \mathcal{C}_q$, and 0 otherwise.

Model UP

The semantic profile of a user \mathbf{u}_m is used by the system to return a unique ranked list. The preference score of an item \mathbf{d}_n is computed as a weighted sum of the indirect preference values based on similarities with other users in each cluster. The sum is weighted by the similarities with the clusters, as follows:

$$\text{pref}(\mathbf{d}_n, \mathbf{u}_m) = \sum_q \text{nsim}(\mathbf{d}_n, \mathcal{C}_q) \sum_i \text{nsim}_q(\mathbf{u}_m, \mathbf{u}_i) \cdot \text{sim}_q(\mathbf{d}_n, \mathbf{u}_i), \quad (5.1)$$

where

$$\text{sim}(\mathbf{d}_n, \mathcal{C}_q) = \frac{\sum_{c_k \in \mathcal{C}_q} d_{n,k}}{\|\mathbf{d}_n\| \sqrt{|\mathcal{C}_q|}}, \quad \text{nsim}(\mathbf{d}_n, \mathcal{C}_q) = \frac{\text{sim}(\mathbf{d}_n, \mathcal{C}_q)}{\sum_i \text{sim}(\mathbf{d}_n, \mathcal{C}_i)}$$

are the single and normalised similarities between the item \mathbf{d}_n and the cluster \mathcal{C}_q ,

$$\text{sim}_q(u_m, u_i) = \cos(\mathbf{u}_m^q, \mathbf{u}_i^q) = \frac{\mathbf{u}_m^q \cdot \mathbf{u}_i^q}{\|\mathbf{u}_m^q\| \times \|\mathbf{u}_i^q\|}, \quad \text{nsim}_q(u_m, u_i) = \frac{\text{sim}_q(u_m, u_i)}{\sum_j \text{sim}_q(u_m, u_j)}$$

are the single and normalised similarities at layer q between users u_m and u_i , and

$$\text{sim}_q(d_n, u_i) = \cos(\mathbf{d}_n^q, \mathbf{u}_i^q) = \frac{\mathbf{d}_n^q \cdot \mathbf{u}_i^q}{\|\mathbf{d}_n^q\| \times \|\mathbf{u}_i^q\|}$$

is the similarity at layer q between item d_n and user u_i .

The idea behind this first model is to compare the current user interests with those of the others users, and, taking into account the similarities among them, weight all their complacencies about the different items. The comparisons are done for each concept cluster measuring the similarities between the items and the clusters. We thus attempt to recommend an item in a double way. First, according to the item characteristics, and second, according to the connections among user interests, in both cases at different semantic layers.

Model UP- q

The preferences of the user are used by the system to return one ranked list per cluster, obtained from the similarities between users and items at each cluster layer. The ranking that corresponds to the cluster for which the user has the highest membership value is selected. The expression is analogous to equation (5.1), but does not include the term that connects the item with each cluster \mathcal{C}_q :

$$\text{pref}_q(d_n, u_m) = \sum_i \text{nsim}_q(u_m, u_i) \cdot \text{sim}_q(d_n, u_i), \quad (5.2)$$

where q maximises $\text{sim}(u_m, \mathcal{C}_q)$.

Analogously to the previous model, this one makes use of the relations among the user interests, and the user satisfactions with the items. The difference here is that recommendations are done separately for each layer. If the current semantic cluster is well identified for a specific item, we expect to achieve better precision/recall results than those obtained with the overall model.

Model NUP

The semantic profile of the user is ignored. The ranking of an item d_n is determined by its similarity with the clusters, and the similarity of the item with the profiles of the users within each cluster. Since the user does not have connections to other users, the influence of each profile is averaged by the number of users M :

$$\text{pref}(d_n, u_m) = \frac{1}{M-1} \sum_q \text{nsim}(d_n, \mathcal{C}_q) \sum_{i \neq m} \text{sim}_q(d_n, u_i). \quad (5.3)$$

Designed for situations in which the current user profile has not yet been defined, this model uniformly gathers all the user complacencies about the items at different semantic layers. Although it would provide worse precision/recall results than the models UP and UP- q , this one might be fairly suitable as a first approach to recommendations previous to manual or automatic user profile constructions.

Model NUP- q

The preferences of the user are ignored, and one ranked list per cluster is delivered. As in the UP- q model, the ranking that corresponds to the cluster the user is most close to is selected. The expression is analogous to equation (5.3), but it does not include the term that connects the item with each cluster \mathcal{C}_q :

$$\text{pref}_q(d_n, u_m) = \frac{1}{M-1} \sum_{i \neq m} \text{sim}_q(d_n, u_i). \quad (5.4)$$

This last model is the most simple of all the proposals. It only measures the users' complacencies with the items at the layers that best fit them, representing thus a kind of item-based collaborative filtering system.

To better understand the above semantic content-based recommendation models, in the next section, we exemplify its execution with a small number of user profiles, manually defined in such a way that they share semantic preferences in different domains.

5.4 An example

For preliminary testing the proposed strategies and models, a simple experiment has been set up. A set of twenty user profiles are considered. Each profile is manually defined considering six possible topics: *animals*, *beach*, *construction*, *family*, *motor* and *vegetation*. The degree of interest of the users for each topic is shown in Table 5.1, ranging over *high*, *medium*, and *low* interest, corresponding to preference weights close to 1, 0.5, and 0.

As it can be seen from the table, the six first users (1 to 6) have *medium* or *high* degrees of interests in *motor* and *construction*. For them it is expected to obtain a common cluster, named cluster 1 in the table. The next six users (7 to 12) share again two topics in their preferences. They like concepts associated with *family* and *animals*. For them a new cluster is expected, named cluster 2. The same situation happens

with the next six users (13 to 18); their common topics are *beach* and *vegetation*, an expected cluster named cluster 3. Finally, the last two users have noisy profiles, in the sense that they do not have preferences easily assigned to one of the previous clusters. However, it is understandable that $User_{19}$ should be assigned to cluster 1 because of his high interests in *construction*, and $User_{20}$ should be assigned to cluster 2 due to his high interests in *family*.

User	Domain						Expected Cluster
	Motor	Construction	Family	Animals	Beach	Vegetation	
$User_1$	High	High	Low	Low	Low	Low	1
$User_2$	High	High	Low	Medium	Low	Low	
$User_3$	High	Medium	Low	Low	Medium	Low	
$User_4$	High	Medium	Low	Medium	Low	Low	
$User_5$	Medium	High	Medium	Low	Low	Low	
$User_6$	Medium	Medium	Low	Low	Low	Low	
$User_7$	Low	Low	High	High	Low	Medium	2
$User_8$	Low	Medium	High	High	Low	Low	
$User_9$	Low	Low	High	Medium	Medium	Low	
$User_{10}$	Low	Low	High	Medium	Low	Medium	
$User_{11}$	Low	Low	Medium	High	Low	Low	
$User_{12}$	Low	Low	Medium	Medium	Low	Low	
$User_{13}$	Low	Low	Low	Low	High	High	3
$User_{14}$	Medium	Low	Low	Low	High	High	
$User_{15}$	Low	Low	Medium	Low	High	Medium	
$User_{16}$	Low	Medium	Low	Low	High	Medium	
$User_{17}$	Low	Low	Low	Medium	Medium	High	
$User_{18}$	Low	Low	Low	Low	Medium	Medium	
$User_{19}$	Low	High	Low	Low	Medium	Low	1
$User_{20}$	Low	Medium	High	Low	Low	Low	2

Table 5.1 Users' interest degrees for each topic, and expected user clusters to be obtained.

Table 5.2 shows the correspondence of concepts to topics. Note that user profiles do not necessarily include all the concepts of a topic. As mentioned before, in real world applications it is unrealistic to assume profiles are complete, since they typically include only a subset of all the actual user preferences.

Domain	Concepts
<i>Motor</i>	Vehicle, Motorcycle, Bicycle, Helicopter, Boat
<i>Construction</i>	Construction, Fortress, Road, Street
<i>Family</i>	Family, Wife, Husband, Daughter, Son, Mother, Father, Sister, Brother
<i>Animals</i>	Animal, Dog, Cat, Bird, Dove, Eagle, Fish, Horse, Rabbit, Reptile, Snake, Turtle
<i>Beach</i>	Water, Sand, Sky
<i>Vegetation</i>	Vegetation, Tree (instance of Vegetation), Plant (instance of Vegetation), Flower (instance of Vegetation)

Table 5.2 Initial concepts for each of the six considered topics.

We have tested our method with this set of twenty user profiles, as explained next. First, new concepts are added to the profiles by the CSA strategy explained in Section 4.1, enhancing the concept and user clustering that follows. The applied clustering strategy is a hierarchical procedure (Duda, Hart, & Stork, 2001) based on the Euclidean distance to measure the similarities between concepts, and the average linkage method to measure the similarities between clusters. During the execution, K (with K the total number of distinct concepts stored in the user profiles) clustering levels were obtained, and a stop criterion to choose an appropriate number of clusters would be needed. In our case, the number of expected clusters is three so the stop criterion was not necessary. Table 5.3 summarises the assignment of users to clusters, showing their corresponding similarities values. It can be shown that the obtained results completely coincide with the expected values presented in Table 5.1. All the users are assigned to their corresponding clusters. Furthermore, the users' similarities values reflect their degrees of belonging to each cluster.

Cluster	Users						
1	<i>User₁</i>	<i>User₂</i>	<i>User₃</i>	<i>User₄</i>	<i>User₅</i>	<i>User₆</i>	<i>User₁₉</i>
	0.522	0.562	0.402	0.468	0.356	0.218	<i>0.194</i>
2	<i>User₇</i>	<i>User₈</i>	<i>User₉</i>	<i>User₁₀</i>	<i>User₁₁</i>	<i>User₁₂</i>	<i>User₂₀</i>
	0.430	0.389	0.374	0.257	0.367	<i>0.169</i>	<i>0.212</i>
3	<i>User₁₃</i>	<i>User₁₄</i>	<i>User₁₅</i>	<i>User₁₆</i>	<i>User₁₇</i>	<i>User₁₈</i>	
	0.776	0.714	0.463	0.437	0.527	<i>0.217</i>	

Table 5.3 User clusters and associated similarity values between users and clusters. The maximum and minimum similarity values are shown in bold and italics respectively.

Once the concept clusters have been automatically identified, and each user has been assigned to a specific cluster, we apply the recommendation models presented in the previous section. A set of twenty four pictures was considered as the retrieval space. Each picture was annotated with (weighted) semantic metadata describing what the image depicts using the six-domain ontology. Observing the weighted annotations, an expert rated the relevance of the pictures for the twenty users of the example, assigning scores between 1 (totally irrelevant) and 5 (very relevant) to each picture, for each user.

We show in Table 5.4 the final concepts obtained and grouped in the semantic constrained spreading activation and concept clustering phases. Although most of the final concepts do not appear in the initial user profiles, they are very important in further steps because they help in the construction of the clusters. In Chapters 6 and 8, we include studies about the influence of the CSA in more realistic empirical experiments.

Cluster	Users
1	MOTOR: Vehicle, Racing-Car, Tractor, Ambulance, Motorcycle, Bicycle, Helicopter, Boat, Sailing-Boat, Water-Motor, Canoe, Surf, Windsurf, Lift, Chair-Lift, Toboggan, Cable-Car, Sleigh, Snow-Cat CONSTRUCTION: Construction, Fortress, Garage, Road, Speedway, Racing-Circuit, Short-Oval, Street, Wind-Tunnel, Pier, Lighthouse, Beach-Hut, Mountain-Hut, Mountain-Shelter, Mountain-Villa
2	FAMILY: Family, Wife, Husband, Daughter, Son, Mother-In-Law, Father-In-Law, Nephew, Parent, 'Fred' (instance of Parent), Grandmother, Grandfather, Mother, Father, Sister, 'Christina' (instance of Sister), Brother, 'Peter' (instance of Brother), Cousin, Widow ANIMALS: Animal, Vertebrates, Invertebrates, Terrestrial, Mammals, Dog, 'Tobby' (instance of Dog), Cat, Bird, Parrot, Pigeon, Dove, Parrot, Eagle, Butterfly, Fish, Horse, Rabbit, Reptile, Snake, Turtle, Tortoise, Crab
3	BEACH: Water, Sand, Sky VEGETATION: Vegetation, 'Tree' (instance of Vegetation), 'Plant' (instance of Vegetation), 'Flower' (instance of Vegetation)

Table 5.4 Concepts assigned to the obtained user clusters classified by semantic topic.

The four different models are finally evaluated by computing their average precision/recall curves (see Section 2.6) for the users of each of the three existing clusters. Figure 5.4 shows the results.

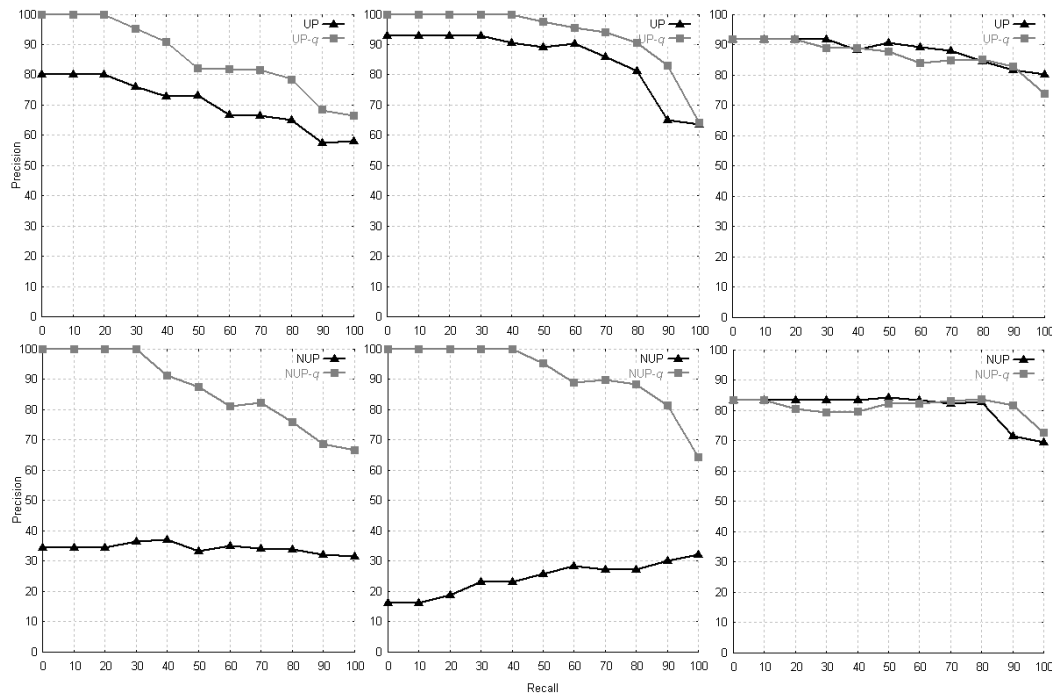


Figure 5.4 Average precision vs. recall curves for users assigned to cluster 1 (left), cluster 2 (centre), and cluster 3 (right). The graphics on top show the performance of the UP and UP- q models. The ones below correspond to the NUP and NUP- q models.

Two conclusions can be inferred from the results: a) the version of the models that return ranked lists according to specific clusters (UP- q and NUP- q) outperforms the one that generates a unique list, and, b) the models that make use of the relations among users in the social networks (UP and UP- q) result in significant improvements with respect to those that do not take into account similarities between user profiles. We shall reinforce this observation in the experiments presented in Chapters 6 and 8.

5.5 Summary

Traditional content-based and collaborative filtering strategies work under the assumption that the entire set of available preferences in the user profiles should be exploited when recommendations have to be performed. The distinction of different layers within the preferences of the users is a desirable property that could help recommender systems to provide more accurate, contextualised item suggestions. Depending on the current context, only a specific subset of the preference layers within a user profile should be considered in order to establish the user's similarities with other people. The identified user similarities based on such context could allow the definition of a community of interest, i.e., a group of people sharing a common interest or passion.

In this chapter, we have presented an approach to the automatic identification of semantic communities of interest according to ontology-based user profiles. Taking into account the semantic preferences of several users, we cluster the ontology concept space, obtaining common topics of interest. With these topics, user profiles are partitioned into different layers. The degree of membership of the obtained sub-profiles to the clusters, and the similarities among them, are used to define links that are exploited by a number of hybrid recommendation models. An illustrative example of the execution of these models has been presented in the chapter, showing initial cues about the benefits of using semantic-based and multilayered techniques to provide content-based collaborative recommendations.

Chapter 6

Evaluation of the recommendation models

In this chapter, we provide empirical results on the evaluation of the collaborative recommenders described in the previous chapters. The experimental work reported here is focused on specific parts of the proposed methods, which are isolated from the rest of the approach in order to a) observe and compare the effect of specific contributions of the thesis, and b) whenever possible, conduct the evaluation on standard collections, adhering to the established evaluation practice in the field.

The experiments on standard datasets support objective observations and comparison, and provide statistic significance, in exchange for some simplifications or adaptations of the proposed techniques, in order to conform to the characteristics and available information in the collections. This is complemented, on the one hand, with smaller prospective tests, in ad-hoc scenarios with a small number of users, of limited scale and objective value, but maximizing their adequacy to the specifics of the proposed methods. On the other hand, an additional, integrative evaluation with real users in a prototype recommender is reported in Chapter 8, which simulates a more natural and realistic scenario where all the proposed models and techniques are integrated in their full form, and where further information from users can be obtained, beyond what is available in standard collections.

In Section 6.1, we describe two different sets of experiments that were conducted to evaluate our semantic group-oriented recommendation model. The first one was designed to find the group modelling strategy that best fits the human way of selecting items when personal tastes of a group have to be considered. The second focused on determining how to measure the satisfaction the strategy offers to the group. Moreover, in Sections 6.2 and 6.3, we present two experiments which assess the feasibility of our semantic multilayer hybrid model when small and large user profile repositories are available. Specifically, the first experiment makes use of manually defined user profiles, and the second exploits synthetic user profiles generated with data from MovieLens (movielens.org) and IMDb (www.imdb.com).

6.1 Evaluation of group-oriented recommendations

Combining several semantic user profiles with the group modelling strategies described in Chapter 4 we seek to establish how humans create an optimal ranked item list for a group, and how they measure the satisfaction of a given list. The theoretical and empirical experiments performed demonstrate the benefits of using semantic user preferences and exhibit which semantic user profile combination strategies could be appropriate to a collaborative environment.

In this section, we study the feasibility of applying the above strategies for combining multiple individual preferences in a personalisation framework from a knowledge-based multimedia retrieval system (Vallet, Castells, Fernández, Mylonas, & Avrithis, 2007). The framework makes use of the ontology-based knowledge representation proposed in this thesis (see Section 4.1), where user preferences are gathered in ontology semantic concept-based user profiles. Using these profiles, and applying the basic semantic content-based recommendation model explained in Section 4.2, the framework retrieves personalised ranked lists of items, and shows them in a graphical interface (Figure 6.1).

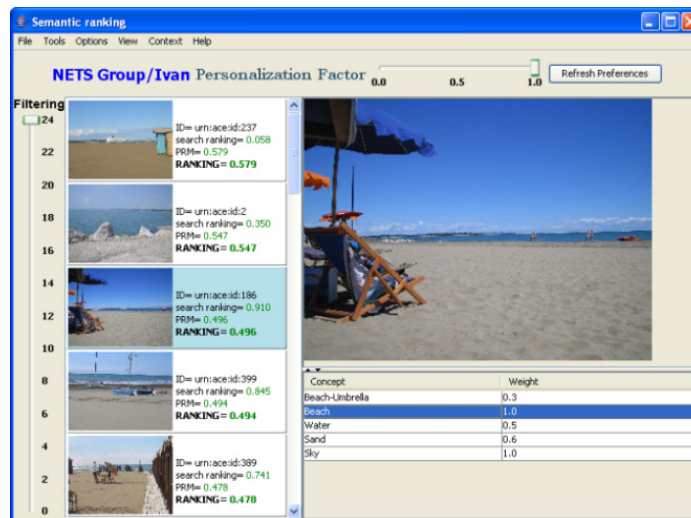


Figure 6.1 Screenshot of the personalisation framework used to evaluate ontology-based group modelling strategies.

In (Masthoff, 2004), Judith Masthoff discusses several strategies for merging individual user models to adapt to groups. Considering a list of TV programs and a group of viewers, she investigates how humans select a sequence of items for the group to watch, how satisfied people believe they would be with the sequence chosen by the different strategies, and how their satisfactions correspond with that predicted by a number of satisfaction functions. These are questions we wanted to investigate through the combination of semantic user profiles.

Two different sets of experiments have been done for those goals. The first one focuses on finding the group modelling strategy that best fits the human way of selecting items when personal tastes of a group have to be considered, i.e., it attempts to establish the strategy that most satisfaction offers to the members of the group. The second tackles the problem in the opposite direction. Given a group modelling strategy, it aims to determine how to measure the satisfaction the strategy offers to the group.

The scenario of the experiments was the following. A set of twenty four pictures was considered. They are shown in Figure 6.2. For each picture, several semantic-annotations were manually taken, describing their topics (at least one of *beach*, *construction*, *family*, *vegetation*, and *motor*) and the degrees (real numbers in $[0,1]$) of appearance the considered topic concepts have on the picture.

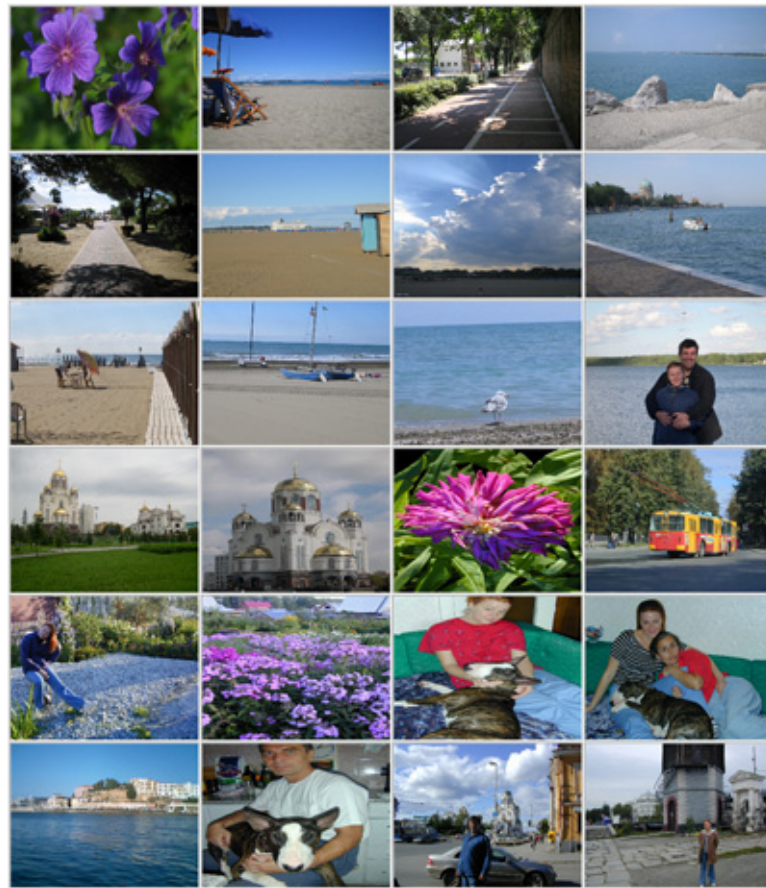


Figure 6.2 Set of pictures used in the evaluation of group-oriented recommendations.

Twenty subjects participated in the experiments. They were Computer Science PhD students. They were asked in all the experiment phases to think about a group of three users with different tastes. In decreasing order of preference (i.e., progressively smaller weights):

- User₁ liked beach, vegetation, motor, construction and family.
- User₂ liked construction, family, motor, vegetation and beach.
- User₃ liked motor, construction, vegetation, family and beach.

Optimal ranking according to human subjects on behalf of a group of users

We define two distances that measure the existing difference between two given ranked items lists. The goal is to determine which group modelling strategies give ranked lists closest to those empirically obtained from several subjects.

Consider \mathcal{I} as the set of items stored and retrieved by the system. Let $\tau_{\text{sub}} \in [0, 1]^{|\mathcal{I}|}$ be the ranked item list for a certain subject, and let $\tau_{\text{str}} \in [0, 1]^{|\mathcal{I}|}$ be the ranked item list for a given combination strategy. We use the notation $\tau(x)$ to refer the position of the item $x \in \mathcal{I}$ in the ranked list τ . The first defined distance between these two ranked lists is:

$$d_1(\tau_{\text{sub}}, \tau_{\text{str}}) = \sum_{x \in \mathcal{I}} |\tau_{\text{sub}}(x) - \tau_{\text{str}}(x)|. \quad (6.1)$$

This expression basically sums the differences between the positions of each item in the subject and strategy ranked lists. Thus, the smaller the distance the more similar the ranked lists. The distance might represent a good measure of the disparity between the user preferences and the ranked list obtained from a group modelling strategy. However, in typical content retrieval systems, where many items are retrieved for a specific query, a user usually takes into account only the first top ranked items. In general, he will not browse the entire list of results, but stop at some top n in the ranking. We propose to more consider those items that appear before the n -th position of the strategy ranking and after the n -th position of the subject ranking, in order to penalise more those of the top n items in the strategy ranked list that are not relevant for the user.

With these ideas in mind, the following could be a valid approximation for our purposes:

$$d(\tau_{\text{sub}}, \tau_{\text{str}}) = \sum_{n=1}^{|\mathcal{I}|} \Pr(n) \frac{1}{n} \sum_{x \in \mathcal{I}} |\tau_{\text{sub}}(x) - \tau_{\text{str}}(x)| \cdot \chi_n(x, \tau_{\text{sub}}, \tau_{\text{str}}),$$

where $\Pr(n)$ is the probability of the user stops browsing the ranked item list at position n , and

$$\chi_n(x, \tau_{\text{sub}}, \tau_{\text{str}}) = \begin{cases} 1 & \text{if } \tau_{\text{str}}(x) \leq n \text{ and } \tau_{\text{sub}}(x) > n \\ 0 & \text{otherwise} \end{cases}.$$

Again, the smaller the distance the more similar the ranked lists.

The problem here is how to define the probability $\Pr(n)$. Although an approximation to the distribution function for $\Pr(n)$ can be taken by interpolation of data from a statistical study, we simplify the model fixing $\Pr(10)=1$ and $\Pr(n)=0$ for $n \neq 10$, assuming that users are only interested in those items shown in the screen at first time after a query.

Our second distance is then defined as follows:

$$d_2(\tau_{\text{sub}}, \tau_{\text{str}}) = \frac{1}{10} \sum_{x \in \mathcal{I}} |\tau_{\text{sub}}(x) - \tau_{\text{str}}(x)| \cdot \chi_{10}(x, \tau_{\text{sub}}, \tau_{\text{str}}). \quad (6.2)$$

Observing the twenty four pictures, and taking into account the preferences of the three users belonging to the group, the twenty subjects were asked to make an ordered list of the pictures. With the obtained lists we measured the distances d_1 and d_2 with respect to the ranked lists given by the group modelling strategies. For each group modelling strategy, two ranked lists were generated by the *profile combination* and *ranking combination* methods proposed in Section 4.4 (see Figures 4.20 and 4.21). The average results are shown in Figure 6.3.

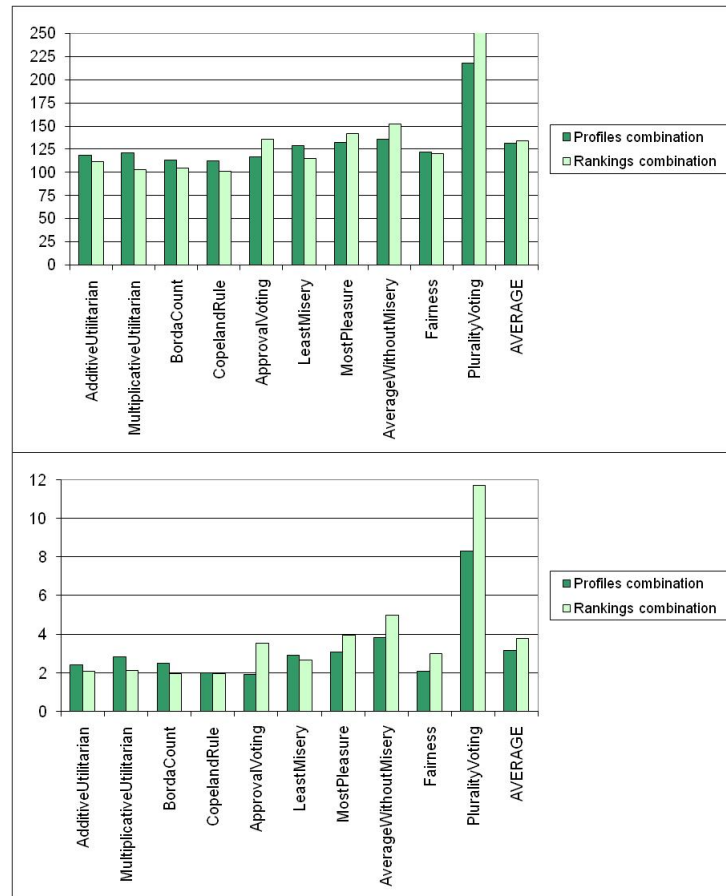


Figure 6.3 Average distances d_1 and d_2 for the subject profile and ranking combination methods.

On one hand, it seems that strategies like *Borda Count* and *Copeland Rule* give lists more similar to those manually created by the subjects, and strategies like *Average Without Misery* and *Plurality Voting* obtained the greatest distances. On the other hand, it can be seen that the profile combination method slightly overcomes the ranking combination method with most of the group modelling strategies.

The above deductions are founded on an empirical point of view. To obtain more theoretical results we also compared the strategies lists against the lists obtained with our personalised content-based recommendation algorithm, applied to the three semantic user profiles. Figure 6.4 exposes the results. Surprisingly, they are very similar to the empirical ones. They agree with the strategies that seem to be more or less adequate for group modelling.

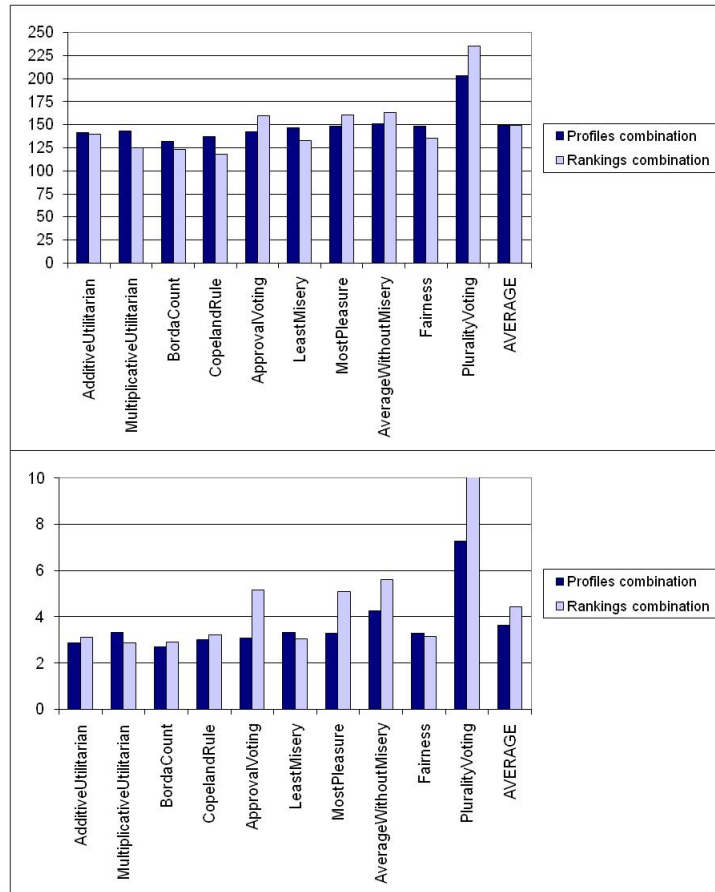


Figure 6.4 Average distances d_1 and d_2 for user profile and ranking combination methods.

Human-measured satisfaction for a content ranking on behalf of a group of users

In the previous experiments we sought to find which group modelling strategies generate ranked list most similar to those established by humans and those created

from our ontology-based user profiles. The idea behind this search is the assumption that the more similar a ranked list is to that generated from a user profile, the most pleasure causes to the user. In the following we establish the same goal, but directly trying to measure the satisfaction each strategy provides. This time, the top ten ranked items from each strategy with all the combination methods were presented to the subjects. Then they were asked to decide the degree of satisfaction each list offers to each of the three users in the group. Four different satisfaction levels were used: *very satisfied*, *satisfied*, *unsatisfied* and *very unsatisfied*, corresponding to four, three, two and one vote respectively. The normalised sums of the obtained votes for each strategy are shown in Figure 6.5.

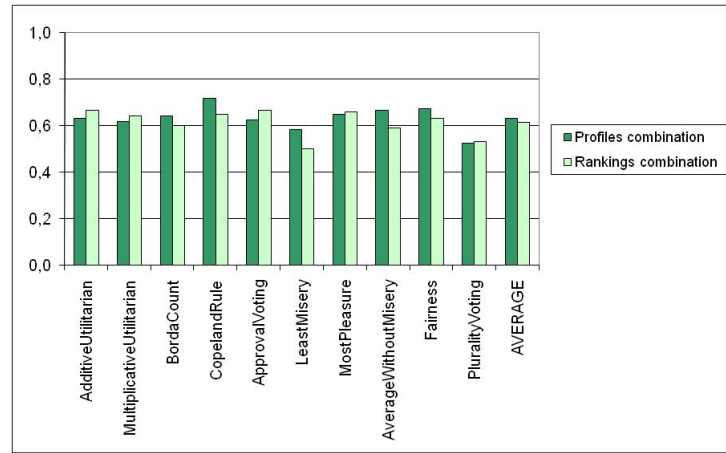


Figure 6.5 Average subject satisfaction.

Once more, a theoretical foundation is needed. In (Masthoff, 2004), three satisfaction functions are presented: a) linear addition satisfaction, b) quadratic addition satisfaction, and, c) quadratic addition minus misery satisfaction. Here, we only study the first one. The quadratic forms are not applicable to our lists because their ratings take values in $[0,1]$, instead of being natural numbers. The way the linear addition satisfaction function measures the pleasure a strategy gives to a specific user is the following. For the n top items of the ranked list τ_{str} , the weights or ratings assigned to these items in the user ranked list are added, and finally normalised:

$$\frac{\sum_{x: \tau_{str}(x) \leq n} w_{user}(x)}{\sum_{x \in I} w_{user}(x)}.$$

In order to be consistent with the empirical experiments, we set $n = 10$. Note that it is necessary for our system to use normalisation. The values of the rankings are skewed within the strategies: some of them are close to 0, and others provide uniform distributed weights in $[0,1]$. Thus, absolute satisfactions values can not be

considered. Figure 6.6 summarises the average satisfaction values for each strategy. The normalised linear addition satisfaction might be a good approximation to real satisfaction values. The satisfaction levels are relatively similar to those obtained from the subjects shown in Figure 6.5, especially in the *Plurality Voting*, where both empirical and theoretical satisfactions are the worst of all the studied strategies. Moreover, it seems there are no significant differences in the satisfaction obtained using profiles and rankings combination methods.

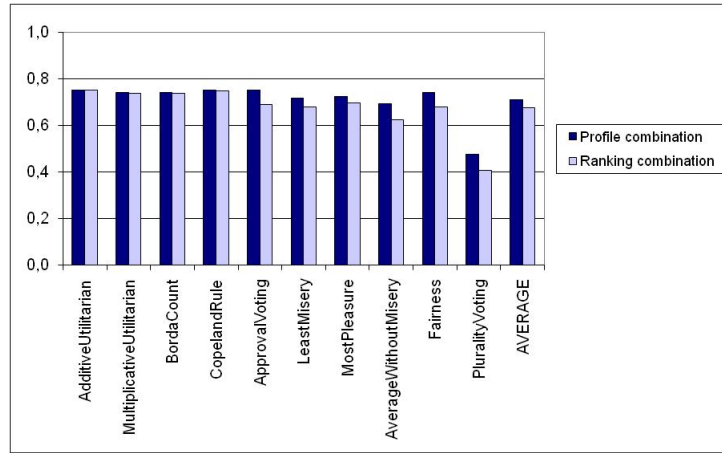


Figure 6.6 Average normalised linear addition user satisfaction.

6.2 Evaluation of hybrid recommendations with a small number of users

We have conducted an experiment with real subjects in order to evaluate the effectiveness of the hybrid recommendation models explained in Chapter 5. Following the ideas explained in the simple example of that chapter, the experiment was set up as follows.

The set of twenty four pictures used in the example was again considered as the retrieval space. Each picture was annotated with semantic metadata describing what the image depicts, using an extended version of the well-known DOLCE upper-level ontology (Gangemi, Guarino, Masolo, Oltramari, & Schneider, 2002), including six certain topics: *animals*, *beach*, *construction*, *family*, *motor* and *vegetation*. A weight in $[0,1]$ was assigned to each annotation, reflecting the relative importance of the concept in the picture.

Twenty graduate students of our department participated in the experiment. They were asked to independently define their weighted preferences about a list of concepts related to the above topics, and existing in the pictures semantic annotations. No restriction was imposed on the number of topics and concepts to be

selected by each of the students. Indeed, the generated user profiles showed very different characteristics, observable not only in their joint interests, but also in their complexity. Some students defined their profiles very thoroughly, while others only annotated a few concepts of interest. This fact was obviously very appropriate to the experiment done. In a real scenario, where an automatic preference learning algorithm should be used, the obtained user profiles would include noisy and incomplete components that will hinder the clustering and recommendation mechanisms.

Once the twenty user profiles were created, we run our method. After the execution of the semantic preference spreading procedure, the domain concept space was clustered according to similar user interests. In this phase, because our strategy is based on a hierarchical clustering method, various clustering levels (which can be represented by the corresponding dendrogram) were found, expressing different compromises between complexity, described in terms of number of concept clusters, and compactness, defined by the number of concepts per cluster or the minimum distance between clusters.

In Figure 6.7, we graph the minimum inter-cluster distance against the number of concept clusters.

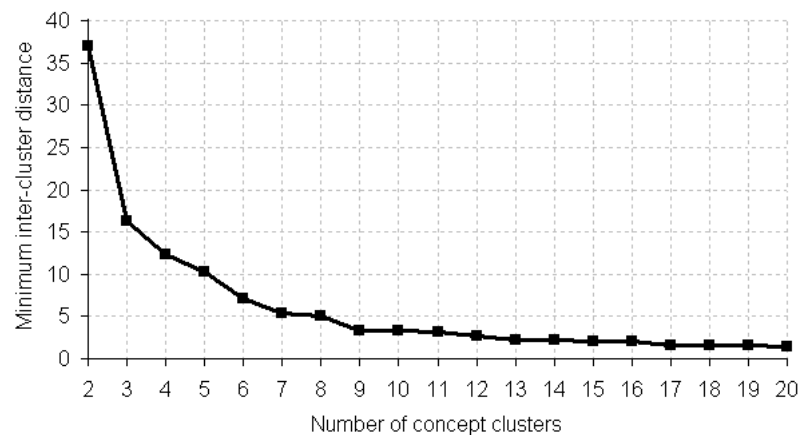


Figure 6.7 Minimum inter-cluster distance at different concept clustering levels.

A stop criterion had then to be applied in order to determine the number of clusters that should be chosen. In this case, we used a rule based on the *elbow criterion*, which says you should choose a number of clusters so that creating another cluster does not add sufficient information. We are interested in a clustering level with a relative small number of clusters, and which does not vary excessively the inter-cluster distance with respect to previous levels. Therefore, attending to the figure, we focused on clustering levels with $Q = 4, 5, 6$ clusters, corresponding to the angle (elbow) in the graph.

Table 6.1 shows the users that most contributed to the definition of the different concept cluster, and their corresponding similarities values.

Q	Cluster	Users									
4	1	User01	User02	User05	User06	User19					
		0.388	0.370	0.457	0.689	0.393					
	2										
	3	User03	User04	User07	User09	User12	User15	User16	User18		
		0.521	0.646	0.618	0.209	0.536	0.697	0.730	0.461		
	4	User08	User10	User11	User13	User14	User17	User20			
		0.900	0.089	0.810	0.591	0.833	0.630	0.777			
5	1	User03	User07								
		0.818	0.635								
	2										
	3	User04	User09	User12	User16	User18					
		0.646	0.209	0.536	0.730	0.461					
	4	User01	User02	User05	User06	User15	User19				
		0.395	0.554	0.554	0.720	0.712	0.399				
	5	User08	User10	User11	User13	User14	User17	User20			
		0.900	0.089	0.810	0.591	0.833	0.630	0.777			
6	1	User6									
		0.818									
	2										
	3	User18									
		0.481									
	4	User02	User05	User06	User19						
		0.554	0.554	0.720	0.399						
	5	User08	User13	User11	User17	User20					
		0.900	0.591	0.810	0.630	0.777					
	6	User01	User04	User07	User09	User10	User12	User14	User15	User16	
		0.786	0.800	0.771	0.600	0.214	0.671	0.857	0.829	0.814	

Table 6.1 User clusters and associated similarity values between users and clusters obtained at concept clustering levels Q=4, 5, 6.

It has to be noted that not all the concept clusters have assigned user profiles. However, there are semantic relations between users within a certain concept cluster, independently from being associated to other clusters or the number of users assigned to the cluster. For instance, at clustering level $Q = 4$, we obtained the weighted semantic relations plotted in Figure 6.8. Representing the semantic CoI of the users, the diagrams of the figure describe the similarity terms $\text{sim}_q(u_i, u_j), i, j \in \{1, 20\}$ (see equations 5.1 and 5.2). The colour of each cell depicts the similarity values between two given users: the dark and light grey cells indicate respectively similarity values greater and lower than 0.5, while the white ones mean no existent relation. Note that a relation between two certain users with a high weight does not necessary implicate a high interest of both for the concepts on the current cluster. What it means is that they interests agree at this layer. They could really like it or they might hate its topics.

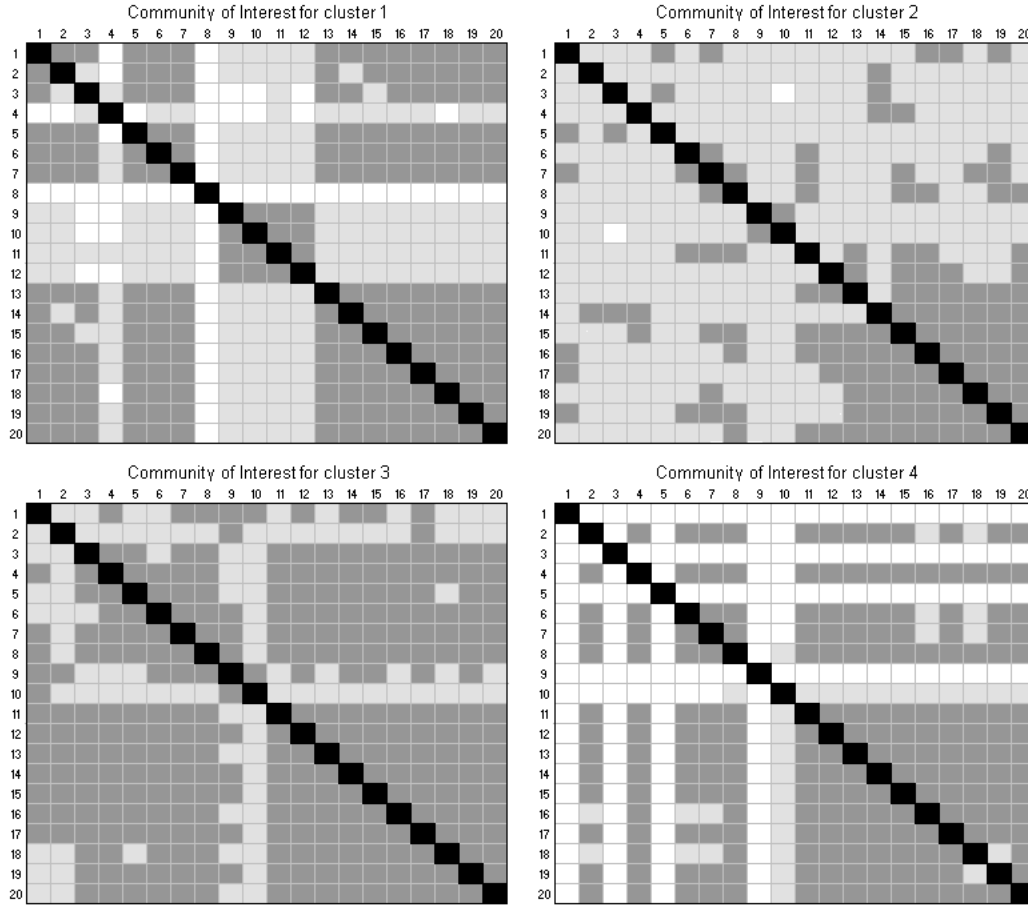


Figure 6.8 Symmetric user similarity matrices at layers 1, 2, 3 and 4 between user profiles u_i and u_j , ($i, j \in \{1, 20\}$) obtained at clustering level $Q=4$. Dark and light grey cells represent respectively similarity values greater and lower than 0.5. White cells mean no relation between users.

Table 6.2 shows the concept clusters obtained at clustering level $Q = 4$. We have underlined those general concepts that initially did not appear in the profiles, and were in the upper levels of the domain ontology. Inferred from our preference spreading strategy, these concepts do not necessary define the specific semantics of the clusters, but help to build the latter during the clustering processes.

Cluster	Concepts
1	ANIMALS: Rabbit CONSTRUCTION: <u>Construction</u> , Speedway, Racing-Circuit, Short-Oval, Garage, Lighthouse, Pier, Beach-Hut, Mountain-Shelter, Mountain-Villa, Mountain-Hut, MOTOR: <u>Vehicle</u> , Ambulance, Racing-Car, Tractor, Canoe, Surf, Windsurf, Water-Motor, Sleigh, Snow-Cat, Lift, Chair-Lift, Toboggan, Cable-Car
2	ANIMALS: <u>Organism</u> , <u>Agentive-Physical-Object</u> , Reptile, Snake, Tortoise, Sheep, Dove, Fish, Mountain-Goat, Reindeer CONSTRUCTION: <u>Non-Agentive-Physical-Object</u> , <u>Geological-Object</u> , <u>Ground</u> , <u>Artefact</u> , Fortress, Road, Street FAMILY: <u>Civil-Status</u> , Wife, Husband MOTOR: <u>Conveyance</u> , Bicycle, Motorcycle, Helicopter, Boat, Sailing-Boat
3	ANIMALS: <u>Animal</u> , <u>Vertebrates</u> , <u>Invertebrates</u> , <u>Terrestrial</u> , <u>Mammals</u> , Dog, ‘Tobby’ (instance of Dog), Cat, Horse, Bird, Eagle, Parrot, Pigeon, Butterfly, Crab BEACH: Water, Sand, Sky VEGETATION: <u>Vegetation</u> , ‘Tree’ (instance of Vegetation), ‘Plant’ (instance of Vegetation), ‘Flower’ (instance of Vegetation)
4	FAMILY: <u>Family</u> , Grandmother, Grandfather, Parent, Mother, Father, Sister, Brother, Daughter, Son, Mother-In-Law, Father-In-Law, Cousin, Nephew, Widow, ‘Fred’ (instance of Parent), ‘Christina’ (instance of Sister), ‘Peter’ (instance of Brother)

Table 6.2 Concept clusters obtained at clustering level $Q=4$.

Several conclusions can be drawn from this experiment. Cluster 1 contains the majority of the most specific concepts related to *construction* and *motor*, showing a significant correlation between these two topics of interest. Checking the profiles of the users associated to the cluster, we observed they overall have medium-high weights on the concepts of these topics. Cluster 2 is the one with more different topics and general concepts. In fact, it is the cluster that does not have assigned users in Table 6.1 and does have the most weakness relations between users in Figure 6.8. It is also notorious that the concepts ‘wife’ and ‘husband’ appear in this cluster. This is due to these concepts were not be annotated in the profiles by the subjects, who were students, not married at the moment. Cluster 3 is the one that gathers all the

concepts about *beach* and *vegetation*. The subjects who liked vegetation items also seemed to be interested in beach items. It also has many of the concepts belonging to the topic of *animals*, but in contrast to cluster 2, the annotations were for more common and domestic animals. Finally, cluster 4 collects the majority of the *family* concepts. It can be observed from the user profiles that a number of subjects only defined their preferences in this topic.

Once the concept clusters were obtained, we evaluated the semantic multilayered hybrid models computing their average precision/recall curves for the users of each of the existing clusters. In this case, we calculated the curves at different clustering levels ($Q = 4, 5, 6$), and we only considered the models UP and UP- q because they make use of the relations among users in the communities of interest, and offer significant improvements with respect to those that do not take into consideration similarities between the active and other users' profiles. Figure 6.9 exposes the results.

Again, the version UP- q , which returns ranked lists according to specific clusters, outperforms the version UP, which generates a unique list assembling the contributions of the users in all the clusters. Obviously, the more clusters we have, the better performance is achieved. The clusters tend to have assigned fewer users, and seem more similar to the individual profiles. However, it can be seen that very good results are obtained with only three clusters. Additionally, for both models, we have plotted with dotted lines the curves achieved without spreading the semantic user preferences. Although more statistically significant experiments have to be done in order to make founded conclusions, it can be pointed out that our clustering strategy performs better when it is combined with the CSA algorithm, especially in the UP- q model. This fact let give us preliminary evidences of the importance of spreading the user profiles before the clustering processes.

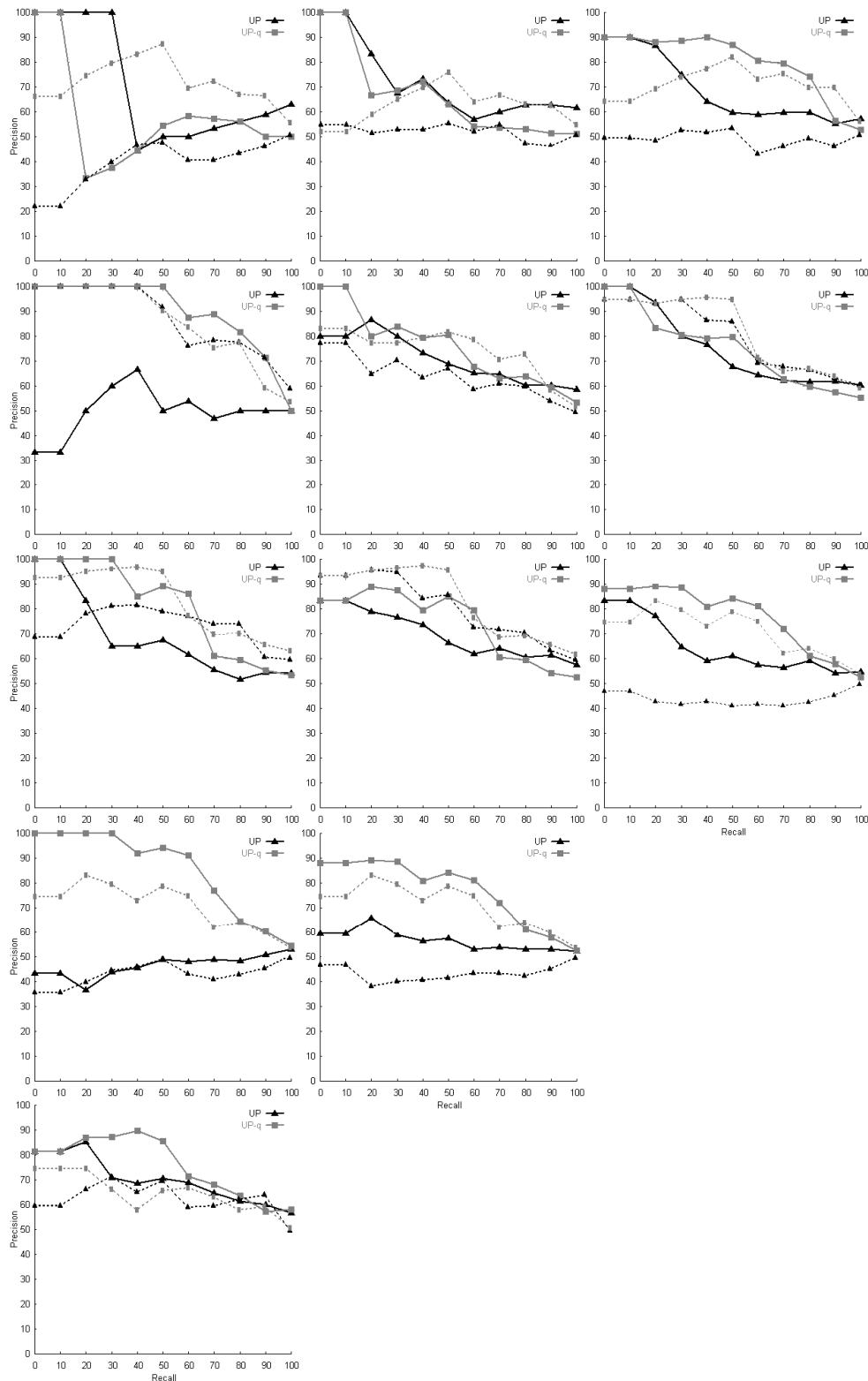


Figure 6.9 Avg. precision vs. recall curves for users assigned to the clusters obtained with the UP (black lines) and UP- q (grey lines) models at levels $Q=6$ (graphics on the left), $Q=5$ (graphics in the middle), and $Q=4$ (graphics on the right) clusters. Dotted lines represent the results achieved without preference spreading.

6.3 Evaluation of hybrid recommendations with a large number of users

The acquisition of a remarkable dataset of user preference and rating information requires a long period of time running a recommender system that really motivates the users to evaluate and rate the existing items. As opposed to the Machine Learning field, in which the UCI repository⁹ gathers tens of datasets that are commonly used by researchers to empirically evaluate and compare the appearing learning algorithms, the Recommender System community lacks the existence of equivalent collaborative rating repositories. The GroupLens research lab¹⁰ at the University of Minnesota (USA) is one the few organisations that has made public a dataset of ratings obtained from an active system. Its recommender system, which is called MovieLens (Figure 6.10), recommends the user movies according to a collaborative filtering approach (Herlocker, Konstan, Borchers, & Riedl, 1999). In this section, we present experiments that exploit the rating information available in MovieLens dataset in order to evaluate our hybrid recommendation models with a large number of users.



Figure 6.10 Screenshot of a MovieLens page, where most recent and rated movies are shown.

Merging MovieLens and IMDb repositories

The MovieLens database is one of the most referenced and evaluated repositories by the Recommender Systems research community. In its large public version, it consists of approximately 1 million ratings for 3,900 movies by 6,040 users on a 1-5

⁹ University of California Irvine (UCI) machine learning repository, <http://archive.ics.uci.edu/ml/>

¹⁰ GroupLens research lab, <http://www.grouplens.org/>

rating scale. This repository is in turn based on the Internet Movie Database¹¹ (IMDb), which probably constitutes the largest collection of movie-related information on the Internet. IMDb pages contain a catalogue of every pertinent detail about a movie, such as the cast, director, genres, shooting locations, languages, soundtracks, etc., as shown in Figure 6.11.



Figure 6.11 Screenshot of an IMDb page, where information about a movie is shown: title, plot, date, genres, director, writer, cast, etc.

In our experiments, we have explored the combination of both sources of data. Specifically, we exploit some of the IMDb information to produce ontology-driven, content-based user profiles from the MovieLens ratings.

For such purpose, we have defined a domain ontology describing the fundamental concepts involved in IMDb, including classes such as movies, actors, directors, genres, languages, countries, keywords, etc., and relations among them. We have parsed the IMDb content (as publicly available in text form), and converted it to an OWL KB, based on the aforementioned movie ontology. Semantic user preferences are then built from the MovieLens ratings by means of a number of transformations that exploit the IMDb KB, and are explained below. The class hierarchy and the semantic relations (object and datatype properties) defined in the domain ontology are shown in Figure 6.12.

¹¹ The Internet Movie Database (IMDb), <http://www.imdb.com/>

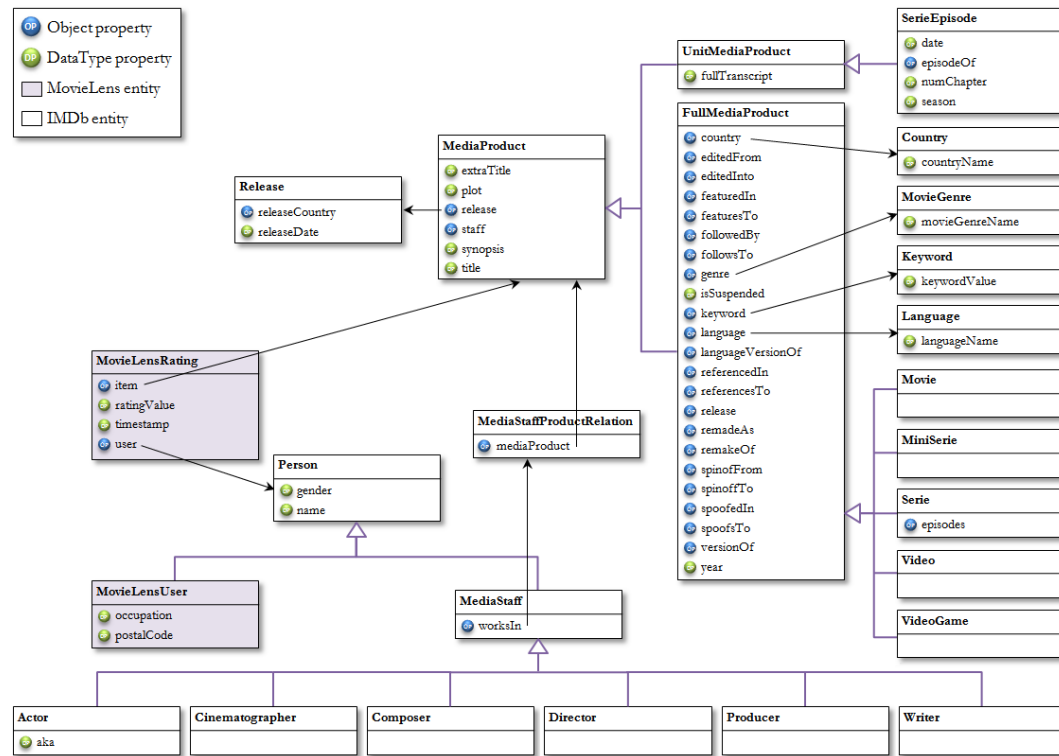


Figure 6.12 MovieLens-IMDb ontology. White boxes correspond to IMDb entities, while coloured boxes are associated to classes that store the information obtained from MovieLens rating repository.

Table 6.3 gathers information about the size of the data and knowledge bases generated from MovieLens and IMDb repositories. Because of the inexact matching between MovieLens and IMDb titles, a set of approximately 250 movies and 30,000 ratings had to be discarded from the original MovieLens database.

MovieLens database	<i>Movies</i>	3,655
	<i>Users</i>	6,040
	<i>Ratings</i>	968,418
IMDb database	<i>Movies</i>	1,095,404
	<i>Genres</i>	28
	<i>Languages</i>	295
	<i>Keywords</i>	32,244
	<i>Actors</i>	1,451,667
	<i>Directors</i>	138,686
IMDb knowledge base	<i>Statements</i>	79,689,194
	<i>Classes</i>	25
	<i>Disk space</i>	~40 GB

Table 6.3 Information about the size of the IMDb and MovieLens data and knowledge bases used in our experiments.

The merging of MovieLens and IMDb information has been followed by other authors. A modified version of the item similarity formula used by item-based CF (expressions 2.18, 2.19 and 2.20) which incorporates semantic-based movie information is presented in (Mobasher, Jin, & Zhou, 2004). More recently, (Symeonidis, Nanopoulos, & Manolopoulos, 2007) proposes the construction of movie feature-weighted user profiles to disclose the duality between users and features in CF. Finally, the gathering of such sources of information in ontological structures for tag-driven recommendation is described in (Szomszor, et al., 2007).

Generating user profiles from MovieLens ratings and IMDb data

The main idea of our approach to build movie content-based user profiles from MovieLens ratings is the following. For each user, we gather all the features (genres, directors, actors, etc.) of those movies he rated. The features are assigned a weight according to the ratings provided by the user. Finally, taking into account the feature distributions, only the less informative features are discarded.

More specifically, let $i_{m,1}, i_{m,2}, \dots, i_{m,N_m}$ be the N_m items (movies) rated by user u_m and let $r_{m,1}, r_{m,2}, \dots, r_{m,N_m} \in [1, 5]$ be the corresponding ratings. We define the weight of movie i_n for user u_m as:

$$w_{m,n} = \frac{r_{m,n}}{5} \in (0, 1].$$

For each user u_m , we measure the relevance of the different movie features by summing the weights of the movies in which these features appear:

$$w_{m,f} = \frac{1}{N_m} \sum_{n: f \in \text{features}(i_n)} w_{m,n}.$$

Hence, for example, we could define the weights for a given movie genre and a specific user as follows:

$$w_{m,g} = \frac{1}{N_m} \sum_{n: g \in \text{genres}(i_n)} w_{m,n}.$$

Taking into account all the movies rated by a user, the feature weights obtained with the previous formulas could be taken as initial semantic user preferences. However, we noticed that we had to filter and select an appropriate proportion of the features to be included in the final profiles as follows. After we expanded the features, we found out that some of them appeared in the user profiles with too many instances, while others with very few. For instance, we observed that in general the initial user profiles contained lots of keywords and very few directors (Figure 6.13). Furthermore, we obtained a lot of weights with values very close to 0, too low

to be considered significant or reliable.

According to the cumulative distributions, for each feature, we selected the number of instances that covers approximately 90% of the feature values distribution. By applying this criterion, the resulting semantic user preferences included the 8 top-weighted genres, 3 countries, 15 actors, and 3 directors per movie. On the other hand, we rejected as user preferences the movie keywords (hundreds per movie) and the spoken languages (the majority of the movies were in English).

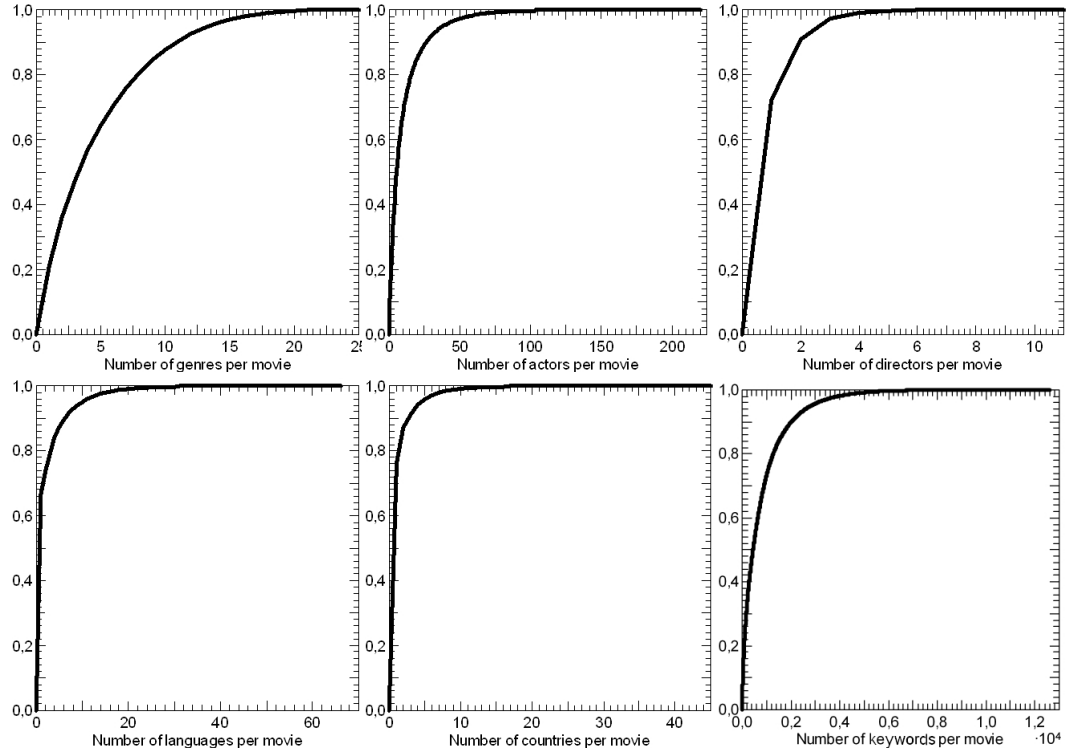


Figure 6.13 Cumulative distributions of IMDb features (genres, actors, directors, languages, countries, keywords) per movie.

Evaluating the hybrid recommendation models

Once the domain ontology and user profiles were built, we evaluated our hybrid recommendation models, comparing them against our pure content-based recommendation algorithm and a classic collaborative filtering strategy.

Conventional recommender algorithms are modelled as ratings estimators. They receive a set of existent user ratings as input and predict new ratings for unseen items. In this context, it is easy to measure the effectiveness of the models if we use evaluations based on the Mean Absolute Error (MAE), i.e., the mean of the absolute differences between the ratings $r_{m,n}$ and their predicted values $p_{m,n}$:

$$\text{MAE} = \frac{1}{M} \sum_{m=1}^M \frac{1}{N_m} \sum_{n=1}^{N_m} |r_{m,n} - p_{m,n}|. \quad (6.1)$$

However, since our recommenders have been defined under a personalised content retrieval framework that generates rankings with values in $[0,1]$, and aiming to make comparisons with MovieLens ratings, we saw the need to convert our recommendations into 1-5 scale ratings. To tackle this issue, we used again the cumulative distributions. In Figure 6.14, we show the cumulative distributions F and G of the real MovieLens ratings and the values obtained with our recommenders.

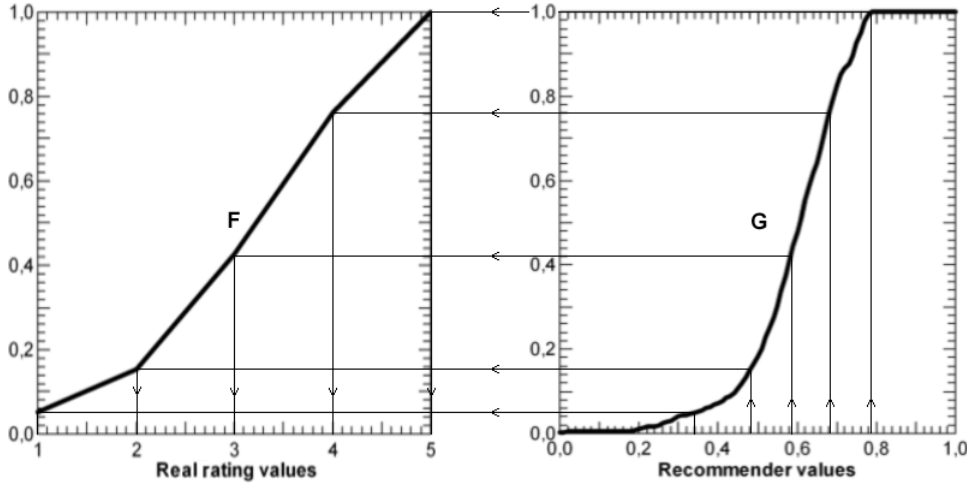


Figure 6.14 Cumulative distribution mappings of our recommender values into MovieLens ratings.

To normalise each predicted value $p_{m,n}$ we first map its cumulative probability $G(p_{m,n})$ into the equivalent cumulative probability $F(r_{m,n})$ in the rating value distribution. Then, we calculate its inverse value $F^{-1}(G(p_{m,n}))$ to extract the corresponding rating $r_{m,n}$:

$$r_{m,n} = F^{-1}(G(p_{m,n})).$$

Once the rating transformations are defined, we are able to evaluate our recommenders by measuring their MAE. To this end, we built (“trained”) the models with 100 and 1,000 users, and considering 10% to 90% of their MovieLens ratings. The rest of their ratings were used for testing. Figure 6.15 shows a comparison between the MAE values obtained with the pure content-based and the hybrid recommendation models (UP and UP- q).

For both models, the obtained MAE values are not as good as they could be. It is very important to note that the way in which the ontology-based user profiles are generated from MovieLens ratings and IMDb movie features, and the mechanism performed to convert $[0,1]$ personalisation values into 1-5 ratings, are, without any doubt, processes which can be improved. However, this was not the purpose of our

experiment. The important conclusion here is that the cluster-oriented UP- q model appears again to be an appropriate hybrid recommender strategy, significantly outperforming the base line established by our content-based recommender.

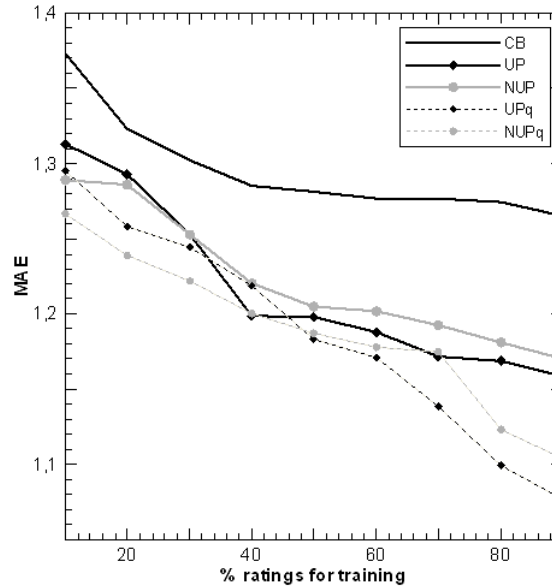
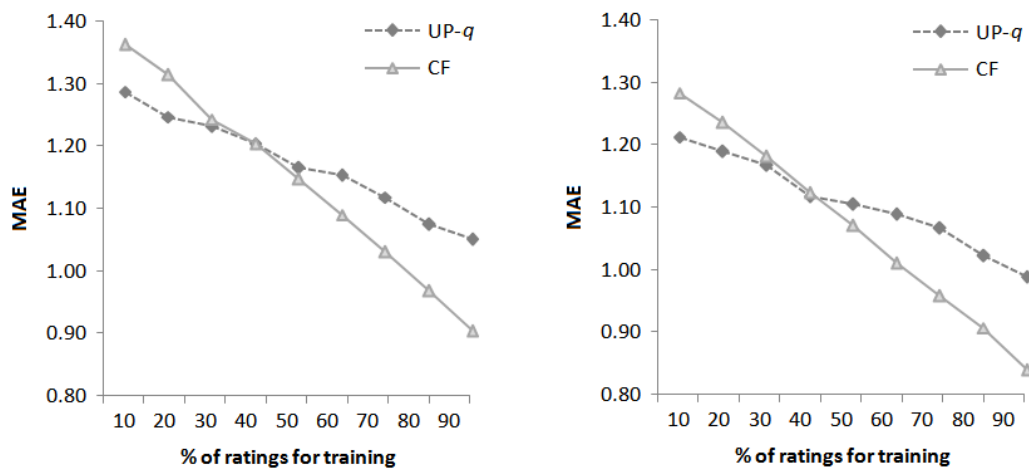


Figure 6.15 MAE for our content-based (CB), and UP, UP- q , NUP and NUP- q hybrid recommenders.

Apart from the comparison between our content-based and hybrid recommendation models, we also wanted to investigate the behaviour of a classic collaborative filtering algorithm when few ratings are available (*cold-start* and *sparsity* problems). Using a public implementation¹² of the item-based CF algorithm, we measured its MAE on the previously used rating datasets. Figure 6.16 shows the results of the CF and the UP- q approaches for 100 and 1,000 users.



¹² Taste Java-based collaborative filtering library, <http://taste.sourceforge.net/>

Figure 6.16 MAE for UP- q and CF recommenders built with 100 (left) and 1,000 (right) users.

When less than the half of the available ratings were used for building the models, our recommender outperformed the collaborative filtering approach, demonstrating thus that the former might be useful when no many ratings are available, and might successfully confront the well-known cold-start and sparsity problems.

6.4 Conclusions

We have presented a set of experiments conducted to assess the feasibility of our collaborative recommendation techniques, i.e., the semantic group-oriented and multilayer hybrid models explained in Chapters 4 and 5.

For the group modelling strategies, through early empirical and theoretical evaluations, we have observed that strategies like *Borda Count* and *Copeland Rule* might be good candidates for the generation of semantic group profiles. We also have shown that the combination of semantic user profiles before the execution of a content retrieval algorithm outperforms the approach of combining ranked item lists, obtained from personalised recommendations with single user profiles.

With respect to our semantic multilayer hybrid recommendation proposal, two sets of experiments were done. The first one was set with a small number of 20 manually defined user profiles, while the second was designed for 100 and 1,000 anonym users whose semantic profiles were built merging information from the MovieLens rating repository and the IMDb movie information database. In both cases, we concluded that the recommendation model focused on specific clusters of shared semantic interests outperforms the global model that computes user and item similarities based on the whole profiles. Moreover, we observed that the semantic preference extension is beneficial not only for our clustering and CoI discovery strategies, but it is essential to obtain accurate recommendation results when little preference and rating information is available, fact that raises the well-known cold-start and sparsity limitations in current recommender systems.

Our implementation of the applied clustering strategy was a hierarchical procedure based on the Euclidean distance to measure the similarities between concepts, and the average linkage method to measure the similarities between clusters. Of course, several aspects of the clustering algorithm have to be investigated in future work using noisy user profiles, such as the type of clustering, the distance measure between two concepts, the distance measure between two clusters, the stop criterion that determines what number of clusters should be chosen, and the similarity measure between given clusters and user profiles; we have used a measure considering the relative size of the clusters, but we have not taken into account what

proportion of the user preferences is being satisfied by the different concept clusters. Moreover, we have to study efficient clustering strategies based on Latent Semantic Analysis (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990; Landauer, Foltz, & Laham, 1998), and/or co-clustering (George & Merugu, 2005).

We are also aware of the need to test our approach in combination with automatic user preference learning techniques in order to investigate its robustness to imprecise user interests, and the impact of the accuracy of the ontology-based profiles on the correct performance of the clustering processes. An adequate acquisition of the concepts of interest and their further classification and annotation in the ontology-based profiles will be crucial to the correct performance of the clustering processes.

In the next part of the thesis, we present a web-based recommender system which integrates all our recommendation models. This system allows users to easily define their profiles, see their semantic relations with other people, and evaluate/rate the existing items. Enlarging the repositories of user and item profiles, we introduce additional experiments that enhance our empirical studies, and reinforce the conclusions obtained in this chapter.

The experiments described in the previous sections focus on conforming to the established scientific experimental practice in the field, using standard datasets, and comparing the proposed recommendation models with classic approaches reported in the literature. These experiments were centred on the evaluation of the multilayer recommendation approach. Other techniques, such as the context-awareness model, require further data (mainly user input) that is neither available in standard collections, nor easy to add as extensions of the latter. On the other hand, the evaluation of the multilayer approach with the standard dataset is achieved, in a way, in an artificial setting. In order to complement these experiments, and reach where they fall short, we have conducted additional evaluations in the above prototype system which completes the experimental work, in a less restricted, more natural way, with a more realistic setting.

Part III

Further evaluations: an integrative experiment

Chapter 7

Evaluation platform

The chapters of the second part of this thesis have presented several ontology-based recommendation models which make item suggestions to single and multiple users, allowing the incorporation of semantic preference spreading and contextualisation mechanisms into the content retrieval processes. As reported in Chapter 6, these models were evaluated in isolation in different experimental setups. An experiment was conducted with manually defined user profiles in controlled small-scale domain scenarios. Another experiment was conducted using synthetic user profiles, generated by merging MovieLens (a well-known movie rating repository) and IMDb (a large movie information database). Positive results, showing the benefits of the proposed approaches, were obtained in both cases. However, we noticed the need of testing the above recommendation models in a more natural scenario, with less constrained usage conditions, and evaluating other techniques, such as the context-aware recommendation approach, which require further, more precise profiling information from explicit user feedback. For these reasons, we decided to develop a prototype system in which all the proposed models were integrated and jointly tested.

In this last part of the thesis, we present *News@hand*, a news recommender system which integrates the personalised, context-aware, collaborative filtering, and hybrid recommendation techniques exposed in previous chapters. The system automatically retrieves news items from on-line media sources, annotates their contents with concepts available in domain ontologies, and allows users to define their semantic profiles in the same concept space to receive personalised ranked lists of news articles. Chapter 7 is dedicated to the description of the system architecture and graphical user interface functionalities. Chapter 8 presents a set of experiments where combinations of the proposed recommendation algorithms are investigated.

The rest of the chapter is organised as follows. Section 7.1 motivates the goal of providing personal recommendations of news items, and introduces *News@hand* system. Section 7.2 summarises the state-of-the-art in news recommender systems. Finally, Sections 7.3 and 7.4 explain respectively the architecture and the graphical user interface of *News@hand*.

7.1 News@hand: a semantic-based approach to recommending news

With the advent of the WWW, people nowadays not only have access to more worldwide news information than ever before, but can also obtain it in a more timely manner. Online newspapers present breaking news on their websites in real time, and users can receive automatic notifications about them via RSS¹³ feeds. RSS is a convenient way to promote a site without the need to advertise or create complicated content sharing partnerships, and an easy mechanism for the users to be informed of the latest news or web contents. Even with such facilities, further issues remain nonetheless to be addressed. For one, the increasing volume, growth rate, ubiquity of access, and the unstructured nature of content challenge the limits of human processing capabilities. It is in such scenario where recommender systems can do their most, by scanning the space of choices, and predicting the potential usefulness of news for each particular user, without explicitly specifying needs or querying for items whose existence is unknown beforehand.

However, general common problems have not been fully solved yet, and further investigation is needed. For example, typical approaches are domain dependent. Their models are generated from information gathered within a specific domain, and cannot be easily extended and/or incorporated to other systems. Moreover, the need for further flexibility in the form of query-driven or group-oriented recommendations, and the consideration of contextual features during the recommendation processes are also unfulfilled requirements in most systems.

In this chapter, we present *News@hand*, a system that makes use of semantic-based technologies to recommend news. The system supports different recommendation models for single and multiple users which address several limitations of recommender systems. The exploitation of meta-information in the form of ontologies that describe user preferences and news contents in a general, portable way, along with the capability of inferring knowledge from the semantic relations defined in the ontologies, represent novel aspects of the system.

7.2 Related work

We briefly describe adaptive news recommender systems that have been proposed in the literature, and highlight the ways in which they suggest news: based on personal content-based preferences, or using collaborative ratings. In subsequent sections, we shall compare the systems' characteristics/functionalities with those of *News@hand*.

¹³ Really Simple Syndication (visit RSS Advisory Board website, <http://www.rssboard.org/>)

7.2.1 Content-based news recommender systems

In content-based approaches, articles are suggested according to a comparison between their contents and the user profiles, the latter containing information about the users' content-based tastes and interests. Data structures for both of these components are created using features extracted from the texts, and a weighting scheme is often used to assign high weights to the most discriminating features/preferences, and low weights to the less informative ones.

News Dude (Billsus & Pazzani, 1999) is a personal news agent that uses a separate model for short-term and long-term interests. To determine the short-term recommendations, news stories are described in terms of TF-IDF vectors, and are provided to a learning module based on the Nearest Neighbours algorithm. To establish the long-term recommendations, news stories are represented as Boolean feature vectors, where each feature indicates the presence or absence of a word, and are presented to a Bayesian learning module.

News4U (Jones, Quested, & Thomson, 2000) is a system where articles from a variety of online news sources are used to create a personalised news paper. The user can decide which news sources to include in the newspaper, and can choose from a list of topics those he is interested in. For a single user, the system applies content-based filtering on previous classifications to rank news.

YourNews (Ahn, Brusilovsky, Grady, He, & Syn, 2007) is a personalised news system which allows users to view and edit their interest profiles. The system's crawlers periodically gather new articles from RSS feeds, passing them to an indexing module to build an index based on news title, description and content. The indexing module creates and stores TF-IDF term vectors of the articles. The user profile for each of the existing news topics is also presented as a weighted term vector extracted from the user's news view history. Users are provided a number of different news rankings according to the specific selection of a topic, a time period (short and long-term preferences), and a type of view (recent and recommended news).

ePaper (Shoval, Maidel, & Shapira, 2008) is a personalised electronic newspaper which incorporates a common ontology for representing both the users' and the items' profiles with concepts taken from the same vocabulary. Based on this knowledge representation, and utilising the ontology hierarchy, the system makes use of a content-based method for filtering items to a given user. The active user's profile is compared with the items' profiles using a similarity measure that takes into account the existence of mutual concepts in both profiles, as well as "related" concepts according to their position in the ontology hierarchy. Based on the computed similarities, items are ranked to the user. At the time of writing, *ePaper* system is utilising an ontology with the high levels of the IPTC¹⁴ news categorisation.

¹⁴ International Press Telecommunications Council, <http://www.iptc.org/>

The content-based features can be combined with additional information, such as implicit behaviour of the user or explicit relevance feedback.

NewT, News Tailor, (Maes, 1994) is a system which filters incoming news articles. Based on full text analysis to retrieve keywords from each article, several filtering agents are trained for different types of information: one for political news, one for sports, etc. The user can provide positive or negative feedback on articles, parts of an article, authors or sources, and this feedback is used to update the corresponding agent.

Daily Learner (Billsus & Pazzani, 2000) is an adaptive news service in which a user first chooses categories he wants to receive news about. Based on the user profile, the system delivers those stories that best match the user's interests. Then, the user explicitly provides feedback using four rating values: interesting, not interesting, more information, already known. Short-term interests are determined by analysing the N most recently rated stories. Long-term interests are not user specific, but category specific.

PENS (Nadjarbashi-Noghani, Zhang, Sadat, & Ghorbani, 2005) is a personalised news system designed as a framework for providing adaptation to user location, user navigation history, and different user devices. A module that implements an unsupervised learning algorithm on user navigation history provides association rules helping to recommend a list of RSS news for a user.

7.2.2 Collaborative news recommender systems

In CF systems, news items are suggested to a particular user according to the articles previously evaluated by other users. In general, users evaluate the texts submitting ratings. These ratings are matched against ratings submitted by all other users, obtaining the user's set of "nearest neighbours". The items that were rated highly by the user's nearest neighbours, and were not rated by the user are finally recommended.

GroupLens project (Konstan, Miller, Maltz, Herlocker, Gordon, & Riedl, 1997) is one of the most referenced CF approaches. Its Netnews recommender is based on a client/server architecture, where users and Netnews are clustered according to the existing news groups, and implicit ratings are built measuring the time the users spent reading the articles.

Personalised Google News (Das, Datar, Garg, & Rajaram, 2007) generates recommendations with three techniques: collaborative filtering using MinHash clustering, Probabilistic Latent Semantic Indexing, and co-visitation counts. These techniques are combined using a linear model providing a scalable recommendation framework. The news ratings get Boolean values taking into account whether the news were clicked or not by the users. Thus, the system presents suggestions to users based on their click history, and the click history of the community.

7.2.3 Hybrid news recommender systems

Hybrid recommendation techniques combine content-based and collaborative filtering strategies under a single framework, mitigating inherent limitations of either paradigm. Numerous ways for combining both types of approaches are conceivable. Among them, the most widely adopted is the so-called “collaborative via content” paradigm, where content-based profiles are built to detect similarities among users.

NewsWeeder (Lang, 1995) is a Netnews filtering system that uses both content-based and collaborative filtering. The user can have access to news through a list of topics (newsgroups), or a virtual personal newsgroup for which a list of articles were selected and ranked. The user must rate each article to have access to the following one with a numeric rating from 1 to 5. The system uses the collected rating information to learn a new model of the user’s interests (off-line learning).

Tango (Claypool, Gokhale, Miranda, Murnikov, Netes, & Sartin, 1999) presents an on-line newspaper recommender which bases a prediction on a weighted average of content-based and collaborative predictions. The content-based and collaborative weights are computed for each user and item according to the number of related ratings. Articles are described as a set of keywords and the newspaper sections they belong to. User profiles are divided into segments corresponding to the newspaper sections. Each segment contains a set of explicit ratings and keywords given by the user, and a list of implicit keywords populated with the keywords of the highly rated articles.

7.3 System architecture

News@hand combines textual features and collaborative information to make news suggestions. However, contrary to previous systems, but similarly to (Shoval, Maidel, & Shapira, 2008), it uses a controlled and structured vocabulary to describe the user preferences and news contents. For this purpose, it makes use of semantic-based technologies. Following the ontology-based knowledge model explained in Section 4.1, user profiles and news items are represented in terms of concepts appearing in domain ontologies, and semantic relations among those concepts are exploited to enrich the above representations, and enhance recommendations.

Figure 7.1 depicts how ontology-based user profiles and item descriptions are created in *News@hand*. Like in other systems (Jones, Quested, & Thomson, 2000; Nadjarbashi-Noghani, Zhang, Sadat, & Ghorbani, 2005; Ahn, Brusilovsky, Grady, He, & Syn, 2007), news are automatically and periodically retrieved from several on-line news services via RSS feeds. Using Natural Language Processing (NLP) and indexing tools, the title and summary of the retrieved news are then annotated with concepts (classes and instances) of the domain ontologies available to the system.

Thus, for example, all the news about actors, actresses and similar terms might be annotated with the concept “actor”. As we shall explain in Chapter 8, *News@band* ontologies contain concepts of multiple domains such as education, culture, politics, religion, science, technology, business, health, entertainment, sports, weather, etc. Similarly to other approaches (Billsus & Pazzani, 1999; Ahn, Brusilovsky, Grady, He, & Syn, 2007), a TF-IDF technique is applied to assign weights to the annotated concepts, measuring their importance (informativeness) to the news contents in the document repository.

News@band has a client/server architecture, where users interact with the system through a web interface in which they receive on-line news recommendations, and update their semantic profiles. Thanks to the AJAX (Asynchronous JavaScript And XML) technology, a dynamic graphical interface allows the system to automatically store all the users’ inputs, analyse their behaviour with the system, update their semantic preferences, and adjust the news recommendations in real time. As done in (Claypool, Gokhale, Miranda, Murnikov, Netes, & Sartin, 1999), explicit and implicit user preferences are taken into account, via manual preferences, tags and ratings, and via automatic learning from the users’ actions (see Chapter 8).

Leveraging the semantically annotated news items, the defined ontology-based user profiles, and the knowledge represented by the domain ontologies, a set of recommendation algorithms is executed. Specifically, *News@band* integrates all the recommendation models explained in Chapters 4 and 5, i.e., personalised, context-aware, group-oriented, and multilayer recommendations.

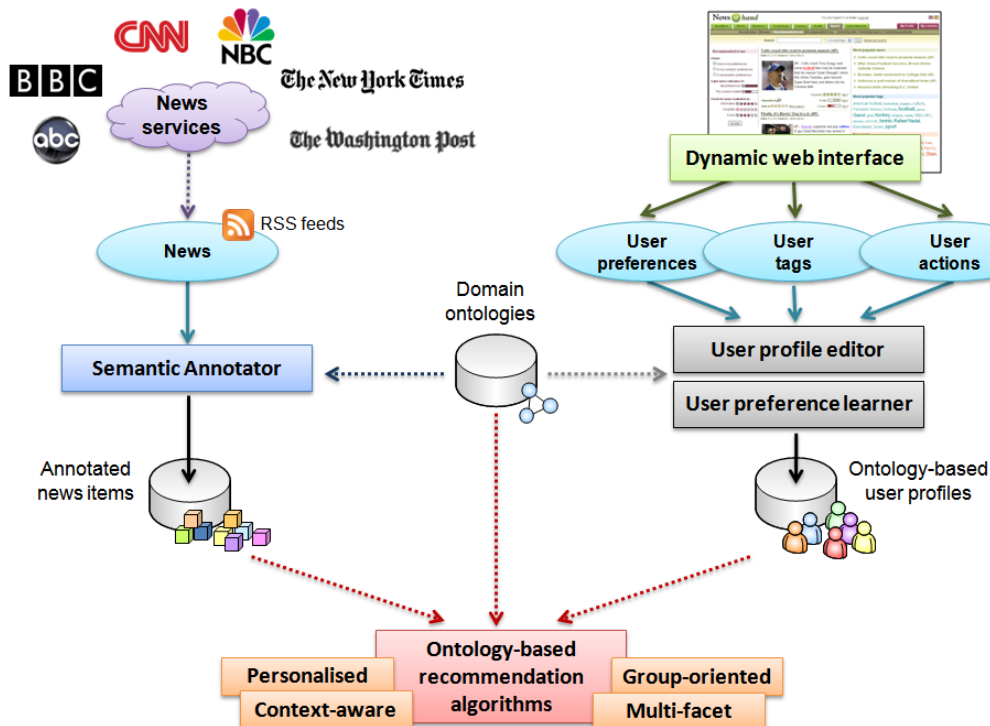


Figure 7.1 Architecture of *News@band*.

Figure 7.2 shows a more detailed schema of the system modules which are directly involved in the domain-independent semantic-based recommendation and user profiling processes. Issues such as the automatic ontology population, the semantic annotation of items, or the capture of user preferences, are explained in Chapter 8 because they are not general issues of the system architecture, and depend on the nature of the items to recommend (textual contents in the case of *News@hand*). In the figure, the arrows indicate dependency relationships from a source to a target component. Three main layers of related modules can be distinguished:

- The **server-side access layer** (top part of the figure) is composed by those modules that receive requests from a client interface, and return the corresponding results: short- and long-term preference reads/updates, and recommendation responses.
- The **recommendation layer** (right part of the figure) contains and combines the proposed semantic-based personalised and collaborative recommenders.
- The **data access layer** (bottom part of the figure) provides functionalities to manage the domain, user preference, user rating, log, and item annotation information exploited by the system using ontologies, databases and indices.

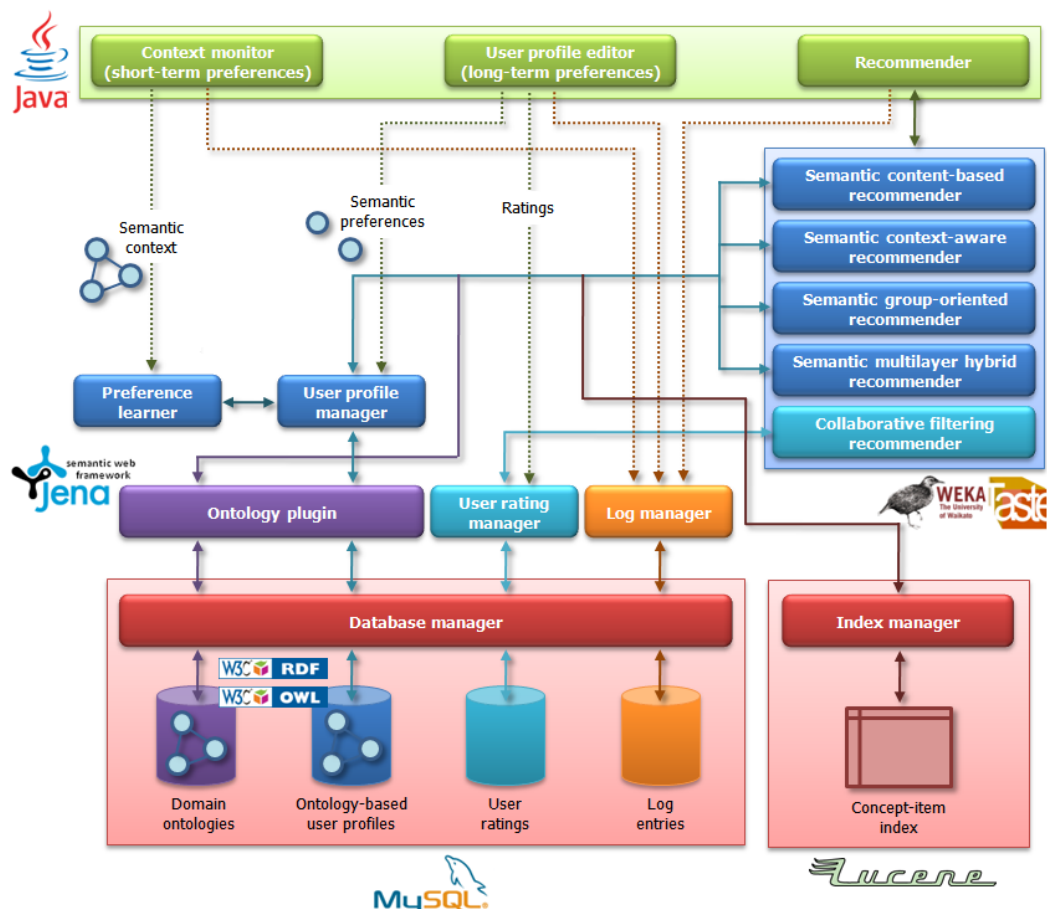


Figure 7.2 Recommendation and user profiling modules of *News@hand*.

All the server-side modules have been implemented in Java¹⁵, and communicate with the web-based client-side software layer through the popular AJAX¹⁶ technology. This allows us to have a web application that asynchronously sends and receives function calls that, among other things, let to make item recommendations in real-time.

The client graphical interface has been developed with Google Web Toolkit¹⁷ (GWT), which provides easily-adaptable rich interface components. These components are compatible with the current most popular web browsers, and have allowed us to include complex functionalities in the recommender system: an on-line ontology viewer, star- and bar-based rating indicators, dynamic news evaluation pop-ups, etc. (see Section 7.4 for more details).

Finally, the data management layer of the system has been built upon relational databases. The database manager chosen for the system was MySQL¹⁸ because the ontology access framework we use, Jena¹⁹, utilises a MySQL connector to retrieve and store ontological information from/to relational databases. In addition, the indexed ontology and content information is accessible via Lucene²⁰ search engine.

The software components are briefly described in the next subsections. In the following, we organise them in four main groups:

- **General-purpose components.** Many of *News@hand* modules access and manage information stored in relational databases and ontology models. For this reason, the system implementation includes:
 - A general Java component for managing relational *databases*, and an implementation of specific Java classes for managing MySQL databases.
 - A general Java component for managing *ontologies*, composed of a set of Java classes that read and write RDF and OWL models stored in text files or relational databases. The component contains the implementation of specific Java classes to manage ontologies using the Jena framework. The access to databases is delegated to the developed database component.
 - A set of general-purpose utility Java classes to make *mathematical computations, vector operations, string manipulation, etc.*

¹⁵ Sun Microsystems Developer Network, <http://java.sun.com/>

¹⁶ AJAX resources, <http://www.dmoz.org/Computers/Programming/Languages/JavaScript/AJAX/>

¹⁷ Google Web Toolkit homepage on Google Code, <http://code.google.com/webtoolkit/>

¹⁸ MySQL database, <http://www.mysql.com/>

¹⁹ Jena Semantic Web framework, <http://jena.sourceforge.net/>

²⁰ Lucene search engine, <http://lucene.apache.org/>

- **User profile management components.** The functionalities associated to the management of user profiles have been distributed in different layers:
 - A component for handling *ontology-based user profiles* stored in OWL models, which accesses to ontology information using the general-purpose components.
 - An upper-level component that stores the content of *user profiles* in the form of Java classes. The information is retrieved and saved through the ontology-based user profile handling component.
 - A component that offers *long-term preference adaptation*. This is a process which is triggered periodically, and updates the semantic interests of the user based on the consumed content.
- **Personalised content retrieval components.** The personalisation content retrieval functionalities have been developed in the following components:
 - A component that performs the *expansion of user preferences* through the relations existing in the domain ontologies (see Section 4.1), providing a semantically enriched description of user interests.
 - A component that computes *personalised semantic content-based recommendations* (Section 4.2), i.e., that generates ranked news lists according to the semantic annotations of the news contents, and to the semantic preferences belonging to the current user's profile.
 - A component that adds into the personalisation content retrieval process those semantic concepts involved in the current *semantic context*, following the formulas given in Section 4.3.
 - A component that implements the *group-oriented recommendation* strategies explained in Section 4.4.
- **Collaborative recommendation components.** The collaborative recommendation of news taking into account the opinions and preferences of other users has been implemented in the following components:
 - A Java component that encapsulates a number of well-known *collaborative filtering* strategies (explained in Section 2.3), adapted from the original Taste²¹ recommendation framework.
 - A set of Java classes implementing the *semantic multilayered hybrid recommendation* strategies explained in Chapter 5 that take into consideration the semantic preferences of users belonging to particular communities of interest.

²¹ Taste Java-based collaborative filtering library, <http://taste.sourceforge.net/>

Appendix B explains in detail the Java packages and classes that contain the software implementation of all the above components. We do not include such explanations in this section because they describe technical issues which may not be of interest for a non computer scientist reader.

7.4 Graphical user interface

The components introduced in Section 7.3 (and detailed in Appendix B) comprise a server-side middleware which abstracts the complex data access and recommendation processes carried out by *News@hand*, providing an easy-to-use API for client programs. The combination of the previous API with the asynchronous remote communication protocol provided by the AJAX technology has facilitated the implementation of a web browser-based graphical user interface, which contains novel functionalities not seen in previous recommender systems, and are worth describing here.

Figure 7.6 shows a screenshot of a typical news recommendation page in *News@hand*. The news items are classified into eight different sections: headlines, world, business, technology, science, health, sports and entertainment. When the user is not logged in the system, he can browse any of the previous sections, but the items are listed without any personalised criterion. On the other hand, when the user is logged in the system, recommendation and user profile edition functionalities are enabled, and the user can browse the news according to his and others' preferences in different ways.

In the middle of the screen, for each news item, apart from its title, source, date, summary, image and link to the full article, additional information is shown. Those terms appearing in the item that are associated to semantic annotations of the contents, the user profile, and the current context are highlighted with different colours. Its global collaborative rating (a linear combination of the results obtained with a pure item-based collaborative filtering strategy, and the semantic multilayer hybrid recommendation technique) is shown in a five-star scale, and two coloured bars indicate the relevance of the news item for the semantic user profile and context separately.

On the left side of the screen, the user can set the input parameters he wants for single or group-oriented recommendations: the consideration of preferences of the user, the user's contacts, or all the users; the degree (weight) of relevance that the concepts of the semantic user profile and context should have in the recommendation algorithms; and multi-criteria conditions to be fulfilled by the user evaluations of the news articles to retrieve.

Finally, on the right side of the screen, general social information such as the most popular news articles (i.e., the best rated by the community), the most used tags, and the top users is shown.

The screenshot shows the News@hand web interface. At the top, there's a navigation bar with categories: Headlines, World, Business, Technology, Science, Health, Sports, and Entertainment. Below this is a search bar and user login information: "You are logged in as Ivan | Log out". The main content area is divided into three columns. The left column contains a "Recommend to me" section with options for "I trust" (Only in my preferences, In my contacts' preferences, In all people's preferences) and "I give more priority to" (My preferences, The current context). The middle column displays three news items: "Commissioners to Congress: No federal law needed", "Hank: Baseball unfairly singled out for steroids", and "Clemens at Congress: Winners, losers, more". Each item includes a date, source (MSNBC), a small image, a summary, and a "Tag it" button. The right column shows "Most popular news" and "Most popular tags".

Figure 7.3 A typical news recommendation page in *News@hand*.

In the next subsections, we explain in more detail remarkable aspects and functionalities provided by the web-based graphical user interface of *News@hand*. Specifically, we explain how a user can set the parameters of the recommendation algorithms, evaluate (rate, tag, comment) suggested items, and edit his profile.

7.4.1 News recommendations

For each news item, in addition to its title, summary, date and source of publication, meta-information is given to the user. Figure 7.4 shows two screenshots where the presentation of news articles includes the following additional data:

- **Coloured terms** for those concepts appearing in the news article title and summary that have been matched (annotated) with a class or instance of the domain ontologies. In the system, the colours have different meanings:
 - The *blue* colour is assigned to concepts appearing in the user profile. When a concept belongs to the extended version of the user profile, the word is also underlined.

- The *purple* colour is assigned to concepts appearing in the current semantic context. If a concept belongs to the extended version of the semantic context, the word is also underlined.
 - The *red* colour is assigned to concepts appearing in both the user profile and the semantic context. Again, if a concept belongs to either the user profile or context extended versions, the word is also underlined.
- A **five (green) starts-scale rating** indicating an average value that takes into account collaborative filtering and ontology-based multilayer hybrid recommendations. If the user wanted more information about how the rating was computed, he could click the link “Why?”
- Two **green-red slide bars** that represent the numeric values obtained with the personalisation mechanism using only the user profile and the current semantic context. If the user wanted more information about how the personalisation values were computed, he could click the link “Why?”
- **Tags** and **comments** given by other users to describe and/or criticise the news item, according to several criteria.



Figure 7.4 Example of meta-information provided by *News@hand* to news items.

Before receiving news item suggestions, the user can set the values of some input parameters of the personalised and group-oriented recommenders: 1) the activation/deactivation of individual preferences, and those of personal contacts or all the users; 2) the weight that the dynamic context should have over the profile, and 3) lower threshold values of various rating criteria to be satisfied by evaluations of the retrieved items. Figure 7.5 shows a screenshot of the panel where the user indicates the values of the above parameters. In this example, the user wants to receive news item suggestion according to semantic preferences of two of his contacts, without taking into consideration the current semantic context and any evaluation restriction, and making use of the semantic expansion mechanism.

Figure 7.5 *News@hand* panel to establish constraints for group recommendations.

7.4.2 User feedback

The user has the possibility to view and add comments, tags and ratings to the articles, following the ideas presented in (Maes, 1994; Lang, 1995; Konstan, Miller, Maltz, Herlocker, Gordon, & Riedl, 1997).

Figure 7.6 shows the pop-up window that appears in the screen after clicking the “tag it” icon of a given news item. When the user is introducing a tag in the text box component, the system suggests those tags existing in its database that start with the already introduced letters. Thus, the user does not have to write the whole words, expending less time during the tagging process, and helping to achieve a reduced set of tags shared by all the users (i.e., a *folksonomy*).

Figure 7.6 Pop-up window to tag a news item in *News@hand*.

In *News@hand*, click history is used to detect the short term user interests that represent the dynamic semantic context exploited by our personalised content retrieval mechanism. When a user clicks the title of a news item, a pop-up window appears showing the source web page with the full article text (Figure 7.7). The lower part of this window contains three buttons that allow the user to evaluate the article. They are labelled as “I like”, “I dislike” and “I don’t mind”. If the user presses the first button, all the semantic concepts annotated in the news item are added into the current context with positive weights (the same of the annotations). In contrast, if the user presses the second button, the concepts are included in the context, but having negative weights (minus the absolute value of the annotation weights). Otherwise, no concept is considered for contextualisation. The context is also updated with the concepts of those news items rated by the user.

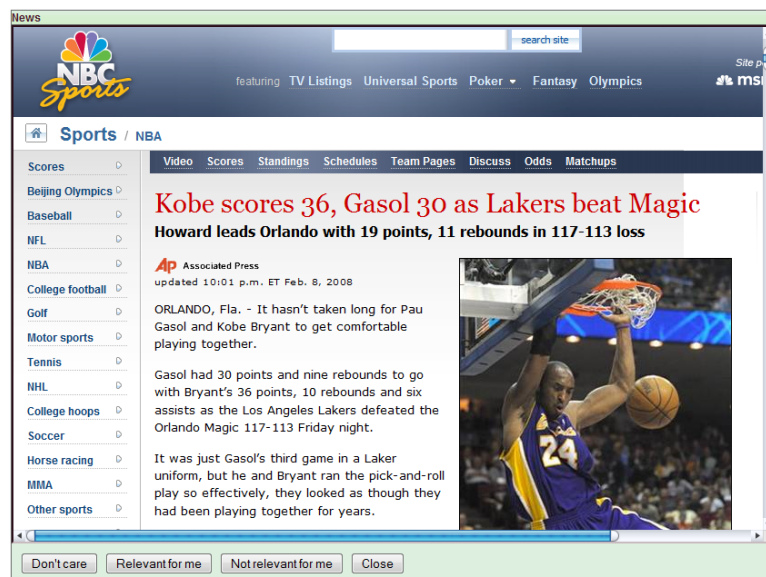


Figure 7.7 Pop-up window to evaluate a news item in *News@hand*.

7.4.3 User profile editor

Apart from the activation/deactivation of multiple recommendation approaches, and the visualisation of annotations and ranking results in the web interface, another important functionality in the graphical interface of *News@hand*, which is shared with other systems (Ahn, Brusilovsky, Grady, He, & Syn, 2007), is the fact that the user can explore and manually edit his profile. Figure 7.8 shows a screenshot of *News@hand* semantic preference editor.

On the upper side of the screen, an editable table contains the user’s semantic preferences, their weights (represented with coloured bars) and their access privacy degrees (public, public for contacts, and private). In this table, if the user clicks on one of the preferences and then presses the button labelled “delete”, the clicked preference is removed from the local user profile. On the other hand, if the user

presses the “save” button, the current preferences shown in the table are automatically sent to and updated in the server.

On the lower part of the screen, an ontology browser allows to view the domain ontology hierarchies, expand/compress their branches, and easily search for specific concepts (auto-complete functionalities are enabled in the search box components).

The screenshot shows the News@hand web application interface. At the top, there's a navigation bar with categories like Headlines, World, Business, Technology, Science, Health, Sports, and Entertainment. Below this is a user profile section with tabs for Personal data, My preferences, My ratings, and My tags. The 'My preferences' tab is active, showing a table of concepts and their weights. The table has columns for Concept, Weight, and Public for (All, My, Only people contacts me). The concepts listed are football, soccer, human interest, political economy, cristiano ronaldo, and banking. Below the table, there's an 'Add' button and a search box. The search box contains the text 'banking'. Below the search box, there's a section titled 'Insert a defined concept or select it from the categories'. This section has a dropdown menu for 'Vocabulary about economy, business and finance' and a list of categories. The 'economy, business and finance' category is selected, and its sub-categories are listed on the left. The right panel shows a list of instances for the selected category, including 'apple bank', 'automated teller machine', 'bank', 'bank charges', 'bank machine', 'bank of america', 'bank of central african states', 'banker', 'banking', and 'bankomat'. The 'banking' instance is highlighted.

Concept	Weight	Public for		
		All	My	Only people contacts me
football	<input type="text" value="100"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
soccer	<input type="text" value="100"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
human interest	<input type="text" value="100"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
political economy	<input type="text" value="100"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
cristiano ronaldo	<input type="text" value="100"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
banking	<input type="text" value="100"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Figure 7.8 Semantic preference editor and ontology browser of *News@hand*.

Analogously to other approaches (Lang, 1995; Claypool, Gokhale, Miranda, Murnikov, Netes, & Sartin, 1999; Billsus & Pazzani, 2000; Jones, Quested, & Thomson, 2000), the news topics/categories are exploited. In this case, they are used to visualise the different ontologies separately in a more legible way. The user selects a category from the combo box component. The class hierarchy of the corresponding ontology (“economy, business and finance” in the example) is then loaded and shown in the left panel of the ontology browser. When a class is clicked in this panel, all its instances are immediately listed in the right panel. Similarly to the deletion of semantic preferences, after a class or an instance is clicked, if the user presses the button labelled “add”, the selected class/instance is incorporated into the upper table (i.e., into the local user profile), where its weight has to be established.

The user can also check and modify personal data (name, age, gender, etc.), ratings, tags and contacts. Here, we do not explain these functionalities because they do not represent relevant research issues. Nonetheless, in Figures 7.9 and 7.10, we show screenshots of the demographic and collaborative profile managers.

News@hand

You are logged in as **ivan** | [Log out](#)

Headlines World Business Technology Science Health Sports Entertainment

Personal data My preferences My ratings My tags

Choose an interest situation: At home [New interest situation](#)

[Save](#)

User password:

Name:

Middle name:

Last name:

Gender: male

Birth:

Title:

Marital status: single

Nationalities:

Languages:

This system has been developed by **Iván Cantador** and **Alejandro Bellogín** (Networked Semantics Team, <http://nets.ii.uam.es>)

Figure 7.9 Personal data editor of *News@hand*.

News@hand

You are logged in as **ivan** | [Log out](#)

Headlines World Business Technology Science Health Sports Entertainment

Personal data My preferences My ratings My tags

Choose an interest situation: At home [New interest situation](#)

My ratings [Save](#)

News item	Rating	Comments	Interest situation	Timestamp	Delete?
Google troubled by Microsoft move	☆☆☆☆☆	View comments	At home	Mon 25 Aug 2008 21:02:20	Ok
Kobe lets Gasol carry Lakers' load vs. Nets	☆☆☆☆☆	View comments	At home	Mon 25 Aug 2008 21:05:10	Ok
Kobe, Gasol in groove as Lakers beat Magic	☆☆☆☆☆	View comments	At home	Mon 25 Aug 2008 21:08:22	Ok

This system has been developed by **Iván Cantador** and **Alejandro Bellogín** (Networked Semantics Team, <http://nets.ii.uam.es>)

Figure 7.10 Personal rating manager of *News@hand*.

7.5 Summary

News@hand is an on-line news recommender system which integrates the ontology-based recommendation models proposed in this thesis. As can be ascertained from the literature reviews of Sections 3.5 and 7.2, the system represents one of the first approaches that make use of semantic-based technologies to describe user preferences and item content features, and exploit the semantic relations between both knowledge representations for making enhanced recommendations.

The architecture of *News@hand* comprises a set of software component layers of special interest for computer scientists and engineers. By following a modular design, we have implemented independent and reusable Java libraries which could be incorporated in other applications. In particular, we have developed general

components for database access, multi-ontology management, and semantic-based recommendation, and more specific modules for index-based search, collaborative filtering, and data clustering that wrap well-known public implementations such as Lucene, Taste and Weka software toolkits.

The graphical user interface of *News@hand* might also seem interesting for recommender system developers. The use of AJAX technology for asynchronous remote communications has allowed us to build a web browser-based interface which incorporates complex graphical components not seen before in previous recommender systems. Remarkable is, for example, the user profile editor. It provides an on-line ontology browser that lets to easily explore ontology taxonomies, search for ontology classes and instances with auto-complete functionalities, and add selected ontology concepts into a semantic user profile.

Chapter 8

User-centred evaluations in the prototype system

News@hand prototype system was implemented in order to allow us to make complementary evaluations with real users, in a less restrictive environment than those in which the isolated experiments of Chapter 6 were conducted.

The architecture and graphical user interface of the system have been described in Chapter 7, but further issues, which are essential to the design of the experimental setups, need to be taken care of in order for the evaluation setting to be fully operational:

- Firstly, the domain ontologies included in the system are adaptations of the IPTC ontology, which is merely a subject taxonomy. This hierarchy must be populated with instances. But, how can real instances be found, and once they are obtained, how are they incorporated into the ontology classes?
- Secondly, the news contents are retrieved automatically from RSS feeds. Then, how are they related (annotated) with ontology classes and instances?
- Finally, a profile editor allows users to define their semantic profiles. However, it is well known that users tend to not declare their interests explicitly. How does the system help the users build their semantic profiles? How can semantic preferences be learned from the users' actions?

In this chapter, we address all the previous issues. We present several methods to automatically populate the ontological knowledge base (Section 8.1), annotate news items (Section 8.2), and obtain semantic user preferences from social tags (Section 8.3). We also report on an additional set of experiments in which our recommendation models are evaluated in an integrative way (Section 8.4).

8.1 Knowledge base

In this section, we describe the Knowledge Base (KB) exploited by *News@hand*. We depict the taxonomy of the domain ontologies, and explain how their classes have been populated with instances extracted from Wikipedia²².

A total of 17 ontologies have been used for the current version of the system. They are adaptations of the IPTC ontology²³, which contains concepts of multiple domains such as education, culture, politics, religion, science, technology, business, health, entertainment, sports, weather, etc. They have been populated with semantic information extracted from news contents and social tags, applying an automatic population mechanism that is explained below. A total of 137,254 Wikipedia entries were used to populate 744 classes with 121,135 instances. Table 8.1 gathers the characteristics of the generated knowledge base. A preliminary evaluation of the ontology population process is given in Section 8.4.1.

Ontology	Attributes			
	#classes	#instances	Avg. #instances/class	memory (KB)
<i>Arts, culture, entertainment</i>	87	33,278	383	5,347
<i>Crime, law, justice</i>	22	971	44	444
<i>Disasters, accidents</i>	16	287	18	358
<i>Economy, business, finance</i>	161	25,345	157	8,468
<i>Education</i>	20	3,542	177	649
<i>Environmental issues</i>	41	20,581	502	692
<i>Health</i>	26	1,078	41	967
<i>Human interests</i>	6	576	96	288
<i>Labour</i>	6	133	22	688
<i>Lifestyle, leisure</i>	29	4,895	169	820
<i>Politics</i>	54	3,206	59	2,989
<i>Religion, belief</i>	31	3,248	105	711
<i>Science, technology</i>	50	7,869	157	1,591
<i>Social issues</i>	39	8,673	222	2,649
<i>Sports</i>	124	5,567	45	6,454
<i>Unrests, conflicts, wars</i>	23	1,820	79	355
<i>Weather</i>	9	66	7	92
	744	121,135	163 (avg.)	33,562

Table 8.1 Number of classes and instances available in *News@hand* knowledge base.

²² Wikipedia, the free encyclopaedia, <http://www.wikipedia.org/>

²³ IPTC ontology, http://nets.ii.uam.es/news-at-hand/news-at-hand_iptc-kb.zip

8.1.1 Domain ontologies

Table 8.2 shows representative classes of the 17 domain ontologies available in the KB of *News@hand*. Some subclasses are given between parentheses.

Ontology	Example classes
<i>Arts, culture, entertainment</i>	art (painting, sculpture, architecture, literature), culture (custom and tradition), entertainment (cinema, theatre), mass media (television, radio, newspaper), music, dance, photography
<i>Crime, law, justice</i>	crime (murder, theft, fraud, drug trafficking, hacking, spamming), law, justice (right, police, trial, punishment, prosecution, prison)
<i>Disasters, accidents</i>	accident, natural disaster (earthquake, hurricane, flood, drought, fire, volcanic eruption), famine, relief and aid organisation, emergency
<i>Economy, business, finance</i>	economy, company information (sale, earning, loss, productivity, bankruptcy, vendor, consumer, contract, marketing, stock option), agriculture, consumer good (food, beverage, clothing, luxury good), metal and mineral, industry, business, finance (banking, market), tourism
<i>Education</i>	educational institution (preschool, school, high school, university), teaching and learning (teacher, student, adult education)
<i>Environmental issues</i>	environmental pollution, environmental politic (waste, energy saving, renewable energy, global warming), natural resource (nature, wildlife, forest, land resource, energy resource)
<i>Health</i>	health care, health problem (disease, injury, epidemic and plague), health treatment (medicine, prescription drug), health organisation (hospital, clinic), medical staff (doctor, nurse)
<i>Human interests</i>	society, imperial and royal matter, award and prize, mystery, curiosity
<i>Labour</i>	labour legislation (health and safety at work), employment and unemployment (occupation, labour market), contract, strike, wage and pension (social security), retirement, workers union
<i>Lifestyle, leisure</i>	lifestyle, leisure (hobby, fishing, hunting), game, lottery, travel and commuting, holiday or vacation, gastronomy
<i>Politics</i>	politics (democracy, socialism, communism, republic), election (political candidate, political campaign, voting), government (head of state, minister, nationalisation, privatisation, civil service, safety of citizens), constitution, parliament, referendum, censorship, human right, foreign aid
<i>Religion, belief</i>	belief [faith], religion (christianity, catholicism, judaism, islam, buddhism), place of worship (church, synagogue, mosque, pagoda), cult and sect
<i>Science, technology</i>	technology (engineering, computer science, micro science, nanotechnology, electronics, biotechnology), human science (history, philosophy, psychology), applied science (mathematics, physics, chemistry, biology, botany, zoology, geology), scientific institution, research, standard
<i>Social issues</i>	social issue (abortion, poverty, charity, homelessness, discrimination, slavery, prostitution, pornography, juvenile delinquency), family (marriage [wedding], divorce, adoption), demographics (immigration, population and census, racism), drug addiction, death and dying, euthanasia
<i>Sports</i>	soccer, football, basketball, tennis, baseball, swimming, motor racing, cycling, athletics, sports event (championship, competition, tournament, grand prix, world cup, olympics)
<i>Unrests, conflicts, wars</i>	armed conflict, war (military intervention, prisoner and detainee), terrorism (guerrilla activity, bioterrorism), riot, civil unrest (rebellion, revolution, religious conflict), massacre, weaponry
<i>Weather</i>	forecast (sunny, cloudy, rainy, snowy, foggy)

Table 8.2 Some classes belonging to the domain ontologies of *News@hand*.

8.1.2 Ontology population

In *News@hand*, ontologies are populated with semantic concepts associated to noun terms extracted from the news contents to be annotated and recommended (Section 8.2), and tags manually introduced by users (Section 8.3.2). These terms are categorised as common nouns (e.g., *actor*) and proper nouns (e.g., *Brad Pitt*).

The terms belonging to the first category are easily processable because their corresponding semantic concepts are the terms themselves. In this case, with simple morphological transformations, the concepts can be found in English dictionaries like WordNet.

The terms of the second category may result in a complex processing. In order to infer their semantic concepts, general multi-domain semantic knowledge is needed. For *News@hand*, we propose to extract that information from Wikipedia.

Wikipedia is a multilingual, open-access, free content encyclopaedia on the Internet. The English Wikipedia edition passed the 2,000,000 article mark on September 2007, and as of October 2008 it had over 2,500,000 articles consisting of over 1 billion words. The Wikipedia articles describe a number of different types of entities: people, places, companies, etc., providing descriptions, references, and even images about the described entities.

Apart from the above elements that describe an entity, every Wikipedia article contains a set of categories that give an idea of the meaning of the associated concept. We have implemented an automatic mechanism that creates ontology instances using, among other things, the Wikipedia categories of the terms. The basic idea of the proposal is to somehow match the categories of an entity with classes of the ontologies, and then link the entity with the matched ontology class that is most “similar” to the entity categories. We explain in detail the whole population process in the following. Firstly, we describe how we extract semantic information from the Wikipedia, and secondly, we explain how we match the extracted information with the ontology classes.

Obtaining semantic information about a term

Many of the entities are ambiguous, having several meanings for different contexts. For instance, the same tag “java” could be assigned to a Flickr²⁴ picture of the Pacific island, or a del.icio.us²⁵ page about the programming language. One approach to address tag disambiguation is by using the information available in Wikipedia. A Wikipedia article is fairly structured: the title of the page is the entity name itself (as found in Wikipedia), the content is divided into well delimited sections, and a first paragraph is dedicated to possible disambiguations for the corresponding term. For

²⁴ Flickr, photo sharing, <http://www.flickr.com/>

²⁵ del.icio.us, social bookmarking, <http://del.icio.us/>

example, the page of the entry “apple” (shown in Figure 8.1) starts as follows:

- “This article is about the *fruit*...”
- “For the *Beatles multimedia corporation*, see...”
- “For *the technology company*, see...”



Figure 8.1 Disambiguation information of the term “apple” in Wikipedia.

Apart from these elements, every article contains a set of collaboratively generated categories. Hence, for example, the categories created for the concept “Teide” are: world heritage sites in Spain, Tenerife, mountains of Spain, volcanoes of Spain, national parks of Spain, stratovolcanoes, hotspot volcanoes, and decade volcanoes (see Figure 8.2). Processing the previous information, we could infer that “Teide” is a volcano located in Spain.



Figure 8.2 Wikipedia categories for the term “Teide”.

Disambiguation and categorisation information have been therefore extracted from Wikipedia for every concept appearing in our news item and social tag datasets. Once the most suitable category for a term is determined, we match its relevant categories to classes defined in the domain ontologies, as explained below.

We have implemented a Java class, called *WikipediaConnector*, which accesses to Wikipedia and extracts the semantic information mentioned in this section for a given term. The structure in which we store all this information is a *WikipediaEntry*. A Wikipedia entry contains the following information:

- **term**, the term of the entry.
- **wikipediaTerm**, the corresponding “filtered” (transformed) term of the entry as it is found in Wikipedia.
- **wikipediaFilter**, the heuristic used to transform the original term (as it is obtained from news contents or tags) into one which is available in Wikipedia.
- **isAbout**, a brief description of the meaning of the term.
- **equivalentTo**, an equivalent term in Wikipedia.
- **ambiguousWith**, ambiguous terms in Wikipedia.
- **hasOtherMeanings**, additional meanings of the term in Wikipedia.
- **categories**, the Wikipedia categories of the term.

The following are the entries automatically generated in the database by the search of the term “java” using the *WikipediaConnector*.

	Entry
Field	java
<i>wikipediaTerm</i>	java
<i>wikipediaFilter</i>	none
<i>isAbout</i>	java island
<i>equivalentTo</i>	
<i>ambiguousWith</i>	java (programming language)
<i>hasOtherMeanings</i>	programming language cigarettes band dance board game
<i>categories</i>	java

	Entry
Field	java (programming language)
<i>wikipediaTerm</i>	java (programming language)
<i>wikipediaFilter</i>	ambiguous
<i>isAbout</i>	programming language
<i>equivalentTo</i>	
<i>ambiguousWith</i>	japanese language
<i>hasOtherMeanings</i>	
<i>categories</i>	java programming language java platform java specification requests c programming language family sun microsystems concurrent programming languages class-based programming languages object-oriented programming languages jvm programming languages

	Entry
Field	japanese language
<i>wikipediaTerm</i>	japanese language
<i>wikipediaFilter</i>	ambiguous
<i>isAbout</i>	indonesian spoken language
<i>equivalentTo</i>	
<i>ambiguousWith</i>	
<i>hasOtherMeanings</i>	
<i>Categories</i>	

Table 8.3 Database entries created after searching for the term “java”.

WikipediaConnector allows us to simplify the datasets identifying concepts that are usually written in different ways (e.g., acronyms). For example, “new york” and “ny” correspond to New York state, and might be related to “new york city” or “nyc”, which correspond to New York city. The interconnected entries generated for these terms are:

	Entry
Field	ny
<i>wikipediaTerm</i>	ny
<i>wikipediaFilter</i>	none
<i>isAbout</i>	state
<i>equivalentTo</i>	new york
<i>ambiguousWith</i>	new york city
<i>hasOtherMeanings</i>	magazine album typeface
<i>categories</i>	new york new york states of the united states former british colonies

	Entry
Field	new york
<i>wikipediaTerm</i>	new york
<i>wikipediaFilter</i>	none
<i>isAbout</i>	state
<i>equivalentTo</i>	
<i>ambiguousWith</i>	new york city
<i>hasOtherMeanings</i>	magazine album typeface
<i>categories</i>	new york new york states of the united states former british colonies

	Entry
Field	new york city
<i>wikipediaTerm</i>	new york city
<i>wikipediaFilter</i>	none
<i>isAbout</i>	
<i>equivalentTo</i>	nyc
<i>ambiguousWith</i>	
<i>hasOtherMeanings</i>	
<i>categories</i>	neighbourhoods in new york city new york coastal cities in the united states former u.s. capitals former u.s. state capitals metropolitan areas of the united states

	Entry
Field	nyc
<i>wikipediaTerm</i>	nyc
<i>wikipediaFilter</i>	none
<i>isAbout</i>	
<i>equivalentTo</i>	
<i>ambiguousWith</i>	
<i>hasOtherMeanings</i>	
<i>Categories</i>	neighbourhoods in new york city new york coastal cities in the united states former u.s. capitals former u.s. state capitals metropolitan areas of the united states

Table 8.4 Database entries automatically created for the term “ny”.

Categorisation of terms into ontology classes

The assignment of an ontology class to a Wikipedia entry is based on a morphological matching measure between the name and the categories of the entry, and the “names” of the ontology classes. The ontology classes with most similar names to the name and categories of the entry are chosen as the classes whereof the corresponding individual (instance) is to be created. The created instances are assigned a URI (see Appendix A) containing the entry name, and RDFS labels with the Wikipedia category names.

To better explain the proposed matching method, let us consider the following example. Let “Brad Pitt” be the concept we want to instantiate. If we look for this concept in Wikipedia, a page with information about the actor is returned. At the end of the page, several categories are shown: “action film actors”, “American film actors”, “American television actors”, “best supporting actor Golden Globe (film)”,

“living people”, “Missouri actors”, “Oklahoma (state) actors”, etc.

After retrieving that information, all the terms (tokens) appearing in the name and categories of the entry (which we will henceforth refer to as entry terms) are morphologically compared with the names of the ontology classes (by the name of a class we mean all the possible textual forms of the class, assuming a class-label mapping is available, as is usually the case). Applying singularisation and stemming mechanisms, and computing the Levenshtein distance (Levenshtein, 1966), only the entry terms that match some class name above a certain similarity threshold, are kept, and the rest are discarded. For instance, suppose that “action”, “actor”, “film”, “people”, and “television” are the entry terms sufficiently close to some ontology class name.

To select the most appropriate ontology class among the matching ones, we firstly create a vector whose components correspond to the filtered entry terms, taking as values the numbers of times each term appears in the entry and category names together. In the example, the vector might be as follows: {(action, 1), (actor, 6), (film, 3), (people, 1), (television, 1)}, assuming that “actor” appears in six categories of the Wikipedia entry “Brad Pitt”, and so forth. Once this vector has been created, one or more ontology classes are selected by the following heuristic:

- If a single component holds the maximum value in the vector, we select the ontology class that matches the corresponding term.
- In case of a tie between several components having the maximum value, a new vector is created, containing the matched classes plus their taxonomic ancestor classes in the ontologies. Then, the weight of each component is computed as the number of times the corresponding class is found in this step. Finally, the original classes that have the highest valued ancestor in the new vector are selected.

Here, “ontology class” and “ancestor” denote a loose notion admitting a broad range of taxonomic constructs, ranging from informally built subject hierarchies (such as the ones defined in the Open Directory tree or, in our experiments, the IPTC subjects), to pure ontology classes in a strict Description Logic sense.

In our example, the weight for the term “actor” is the highest, so we select its matching class as the category of the entry. Thus, assuming that the class matching this term was *Actor*, we finally define *Brad Pitt* as an instance of *Actor*.

Now suppose that, instead, the vector for Brad Pitt was {(actor, 1), (film, 1), (people, 1)}. In this case, there would be a tie in the matching classes, and we would apply the second case of the heuristic. We take the ancestor classes, which could be for example “cinema industry” for “actor”, “cinema industry” for “film”, and “mammal” for “person”, and create a weighted list with the original and ancestor classes. Then, we count the number of times each class appears in the previous list, and create the new

vector: $\{(actor, 1), (film, 1), (person, 1), (cinema\ industry, 2), (mammal, 1)\}$. Since the class *Cinema industry* has the highest weight, we finally select its sub-classes *Actor* and *Film* as the classes of the instance *Brad Pitt*.

We must note that our ontology population mechanism does not necessarily generate individuals following an “is-a” schema, but a more relaxed, fuzzier semantic association principle. This is not a problem for our final purposes in personalised content retrieval, since the annotation and recommendation methods in that area are themselves rooted on models of inherently approximated nature, for example regarding the relationships between concepts and item contents.

8.2 Item annotation

News@hand periodically retrieves news items from the websites of well-known news and media sources, such as ABC, BBC, CNN, NBC, The New York Times, and The Washington Post. These items are obtained via RSS feeds, and contain information of published news articles: their title, summary of contents, publication date, hyperlinks to the full texts and related on-line images.

The system analyses and automatically annotates the textual information (title and summary) of the RSS feeds with concepts (classes and instances) which exist in the domain ontologies, and have been previously indexed. Figure 8.3 depicts the workflow of the whole news item retrieving, indexing and annotation mechanism.

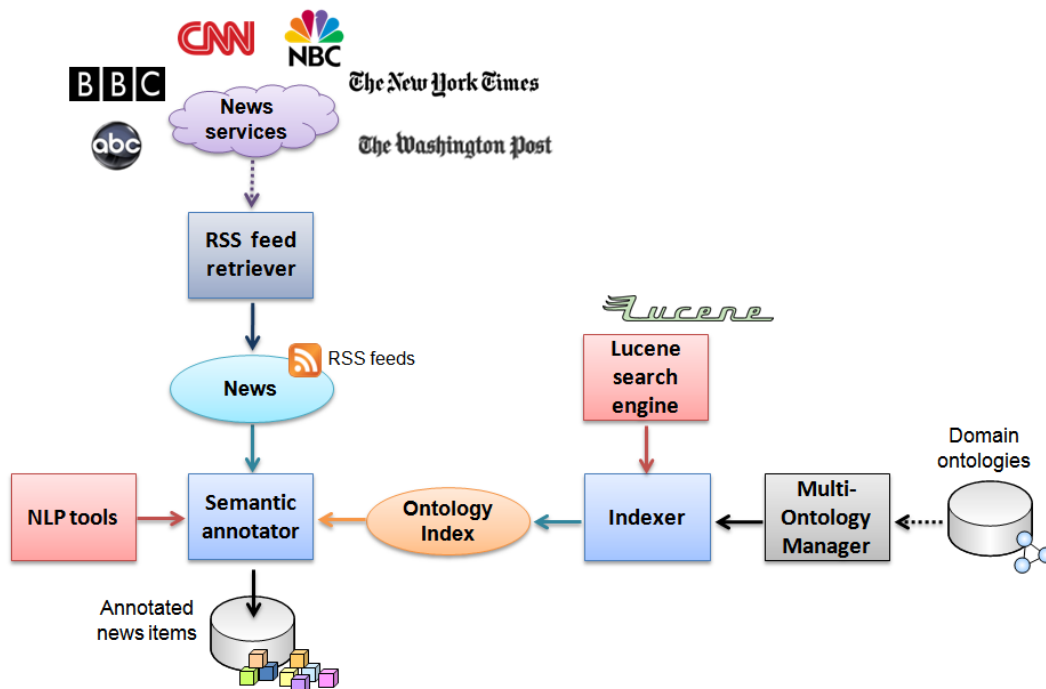


Figure 8.3 Automatic RSS feed extraction and semantic annotation in *News@hand*.

Using a set of Natural Language Processing tools (Alfonseca, Moreno-Sandoval, Guirao, & Ruiz-Casado, 2006), an annotation module removes stop words, and extracts relevant (simple and compound) terms, categorised according to their Part of Speech (PoS): nouns, verbs, adjectives, adverbs, pronouns, prepositions, etc. Then, nouns are morphologically compared with the names of the classes and instances of the domain ontologies. The comparisons are done using an ontology index created with Lucene, and according to fuzzy metrics based on the Levenshtein distance (Levenshtein, 1966). For each term, if similarities above a certain threshold are found, the most similar semantic concepts are chosen and added as annotations of the news items. After all the annotations are created, a TF-IDF technique computes and assigns weights to them.

Figure 8.4 shows a more detailed view of the annotation mechanism, which takes as input the HTML document to annotate, and the system ontology indices, and returns as output new entries for the annotation database.

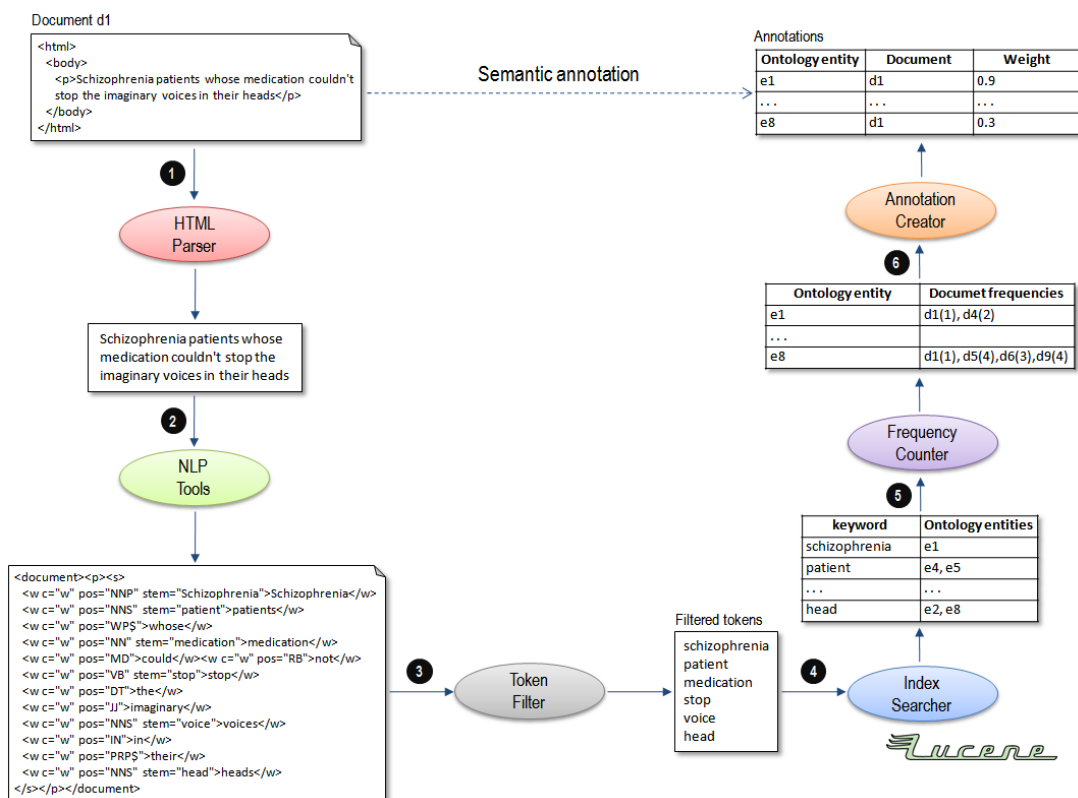


Figure 8.4 Semantic annotation mechanism.

The steps illustrated in the figure are:

- A web document is parsed removing HTML tags and meaningless textual parts (in terms of not having or being related to news contents).

- The remaining text is analysed by the *Wraetlic* linguistic-processing tools to extract the PoS and the stem of each term.
- The information provided by the linguistic analysis is used to filter the less meaningful terms (determinants, prepositions, etc.), and to identify those sets of terms that can operate as individual information units.
- The filtered terms are searched in the ontology indices, obtaining the subset of semantic entities to annotate.
- The annotations are weighted according to the semantic entity frequencies within individual documents and the whole collection.
- The annotations are added to a relational database.

The next subsections explain in more detail the previous steps and provide information about the gathered and annotated news contents.

8.2.1 Natural language processing of news contents

Once the on-line news items have been obtained from their corresponding websites via RSS, a Natural Language Processing (NLP) is made on their textual contents (titles and summaries) in order to detect which of their lexical structures (i.e., terms, or groups of terms) potentially represent ontological entities.

The NLP is carried out by means of the *Wraetlic* linguistic-processing tools (Alfonseca, Moreno-Sandoval, Guirao, & Ruiz-Casado, 2006), an XML suite for processing texts which performs the following tasks:

- **Segmentation:** the identification of lexical units in the texts. It is done by two components: a *tokeniser* which finds word boundaries, and a *sentence splitter* which locates the sentence boundaries. The tokeniser makes use of a list of regular expressions that define the different types of “tokens” appearing in the sentences, such as words, numbers or punctuation symbols. The sentence splitter analyses the words followed by a dot to decide whether they are abbreviations or not, and uses this information to get the sentence boundaries.
- **Part-of-Speech (PoS) tagging:** the assignment of a PoS to each token. A *PoS tagger* labels each token with its corresponding PoS. *Wraetlic* tools utilise the PoS tags of the Penn Treebank corpus²⁶, and take into consideration the grammatical context of a word (i.e., its surrounding terms) to infer its PoS.
- **Morphological analysis:** the study of the inner structure of the words. For each token, a *morphological analyser* identifies the root (stem), which contains the basic meaning of the word, and the bound morphemes (prefixes and suffixes),

²⁶ The Penn Treebank Project, <http://www.cis.upenn.edu/~treebank/>

which vary the basic meaning, e.g., by pluralizing a noun (e.g., “parent” and “parents”), or by changing an adjective into a noun (e.g., “wide” and “width”).

An example

Suppose the following text as the content of a news item to analyse (and annotate):

Schizophrenia patients whose medication couldn't stop the imaginary voices in their heads gained some relief after researchers repeatedly sent a magnetic field into a small area of their brains.

The NLP performed by *Wraetlic* produces the following XML output:

```
<document>
  <p>
    <s>
      <w c="w" pos="NNP" stem="Schizophrenia">Schizophrenia</w>
      <w c="w" pos="NNS" stem="patient">patients</w>
      <w c="w" pos="WP$">whose</w>
      <w c="w" pos="NN" stem="medication">medication</w>
      <w c="w" pos="MD">could</w>
      <w c="w" pos="RB">not</w>
      <w c="w" pos="VB" stem="stop">stop</w>
      <w c="w" pos="DT">the</w>
      <w c="w" pos="JJ">imaginary</w>
      <w c="w" pos="NNS" stem="voice">voices</w>
      <w c="w" pos="IN">in</w>
      <w c="w" pos="PRP$">their</w>
      <w c="w" pos="NNS" stem="head">heads</w>
      <w c="w" pos="VBD" stem="gain">gained</w>
      <w c="w" pos="DT">some</w>
      <w c="w" pos="NN" stem="relief">relief</w>
      <w c="w" pos="IN">after</w>
      <w c="w" pos="NNS" stem="researcher">researchers</w>
      <w c="w" pos="RB">repeatedly</w>
      <w c="w" pos="VBD" stem="send">sent</w>
      <w c="w" pos="DT">a</w>
      <w c="w" pos="JJ">magnetic</w>
      <w c="w" pos="NN" stem="field">field</w>
      <w c="w" pos="IN">into</w>
      <w c="w" pos="DT">a</w>
      <w c="w" pos="JJ">small</w>
      <w c="w" pos="NN" stem="area">area</w>
      <w c="w" pos="IN">of</w>
      <w c="w" pos="PRP$">their</w>
      <w c="w" pos="NNS" stem="brain">brains</w>
    </s>
  </p>
</document>
```

Figure 8.5 XML output provided by *Wraetlic* after the NLP of a text.

As shown in Figure 8.4, the NLP tools parse the document, recognise its paragraphs, sentences and tokens, and provide information about the PoS and the semantic stem of each token. This information will be used afterwards by the annotation module to discard meaningless tokens such as determinants, prepositions, etc., and to identify lexical structures (tokens or groups of tokens) which may potentially match with ontology entities, and may be included in semantic annotations.

8.2.2 Automatic semantic annotation

The semantic annotator identifies ontology entities (classes and instances) within the text documents, and generates links between the identified ontology entities and the documents using index structures. The process can be seen as a traditional IR indexing process where the basic units to create document indices are ontology entities (word senses) instead of plain keywords.

It is important to highlight that the annotation process carried out here does not populate ontologies with new instances appearing in the texts, but identifies already existing ontology entities, thus allowing to maintain the semantic information decoupled from the textual contents.

In contrast to other large scale annotation frameworks, our system has been designed to support annotation in open domain environments where any document can be associated or linked to any ontology without having any restriction. In order to do so, the system has to deal with the scalability problem and the increase of uncertainty in the correct semantic meanings of the annotations.

The scalability problem

The exploitation of a potential unlimited number of ontologies by the annotation process may result in efficiency and scalability limitations. To address them, we propose to use ontology indices and non-embedded annotations. In contrast to systems where annotations are inserted in the ontologies or documents, our mechanism generates non-embedded annotations, and stores them into a relational database, increasing thus the speed of the retrieval algorithms. The following are the steps conducted by our proposal:

- **Generation of the ontology index**

We envision a scenario where a user may need to interact with hundreds of KBs structured in tens of ontologies. To successfully manage such amount of information on real time, the ontologies are analysed and stored into one or more inverted indices using Lucene.

The indexation is based on a mapping between each ontology entity and a set of keywords that represent the meaning of the former. By default, these

keywords are extracted from the entity (local) name and *rdfs:label* meta properties. Optionally, they could be obtained from any other ontology property. The mapping thus allows the generation of inverted indices where each keyword can be associated to several semantic entities belonging to different ontologies.

To retrieve the semantic information stored in the indices we make use of the advantages that Lucene provides for approximate (fuzzy) searches.

- **Construction of the annotation database**

As mentioned before, the created annotations are stored using a relational database in order to increase the efficiency of the retrieval phase. For each annotation, an entry is generated in the database to gather the identifiers of the corresponding semantic entity and document, as well as a weight indicating the degree of relevance of the semantic entity within the document.

In traditional IR indexing systems, keywords appearing in a document are assigned weights reflecting the fact that some words are better at discriminating between documents. Similarly, in our system, semantic annotations are assigned weights that reflect how well the ontology entities represent the meaning of the document. Weights are automatically computed by an adaptation of the TF-IDF algorithm, and based on the frequency of the occurrences of each ontology entity within the document.

Initially, the frequency of occurrences of an entity in a document was defined as the number of times any of its associated “mappings” appears in the document text. However, in preliminary experiments, we realised that quite a number of occurrences were missed, since we were not considering pronouns as entity occurrences. To slightly overcome this limitation, we included a modification in the algorithm to also count pronoun occurrences in a sentence if an entity was previously identified. This modification does not help to increase the annotation accuracy or incorporate new annotations, but enhances the preciseness of the annotation weights that will be later used during the recommendation and ranking processes.

As explained in Section 2.2, the weight $w_{k,n}$ in the annotation of a document d_n with an ontology entity c_k is computed as:

$$w_{k,n} = \text{TF-IDF}_{k,n} = \frac{\text{freq}_{k,n}}{\max_j \text{freq}_{j,n}} \cdot \log \frac{N}{N_k},$$

where $\text{freq}_{k,n}$ is the number of occurrences in d_n of the keywords attached to c_k , $\max_j \text{freq}_{j,n}$ is the frequency of the most repeated ontology entity in d_n , N_k is the number of documents annotated with c_k , and N is the total number of documents.

The relational model designed to store the above annotations is composed by the following tables:

- *Annotation table*. This table stores the annotations, linking documents with ontology entities through weights.

Entity ID	Document ID	Weight
1829048176	3614522287	0.54
1829048179	3614522287	0.21

- *Ontology entity table*. This table stores index information about ontology entities. Each entity is identified by its ontology, URI and type (class, instances, property, literal), and has associated a set of text labels.

Entity ID	Entity URI	Entity type	Entity labels	Ontology ID
1829048176	0#Teide	instance	teide	45
1829048179	1#boat	class	boat, ship	46

- *Document table*. This table stores information about the textual documents. Each document is identified by its URI and repository or media source.

Document ID	Document URI	Repository ID
3614522287	24#CNN_D1	21
3614522289	24#CNN_D2	24

- *Prefix table*. This table was designed to optimise the storage of namespaces in the database.

Prefix	Namespace
0	http://geography.com/spain/mountain
1	http://transports.net/watercraft
24	http://www.cnn.com/travel

The increase of the uncertainty degree of the annotations

The use of a potentially unlimited number of domain ontologies increases the uncertainty of the annotations as more morphological similar concepts (with divergent semantic meanings) can be found. To address this limitation, we propose to exploit the PoS information provided by *Wraetlic* NLP tools in order to identify and discard those words that typically do not provide significant semantic information. Moreover, we group sets of words that can operate as individual semantic information units. The following are some examples of the considered word group patterns.

- *Noun + noun*. E.g., “tea cup”.
- *Proper noun + proper noun*. E.g., “San Francisco”.
- *Proper noun + proper noun + proper noun*. E.g., “Federico García Lorca”.
- *Abbreviation + proper noun + proper noun*. E.g., “F. García Lorca”.
- *Abbreviation + abbreviation + proper noun*. E.g., “F. G. Lorca”.
- *Participle + preposition*. E.g., “located in”, “stored in”.
- *Modal verb + participle + preposition*. E.g., “is composed by”, “is generated with”.

8.2.3 Annotation database

We have run our semantic annotation approach on a set of 9,698 news items daily retrieved during two months. The ontological KB from which we obtained the semantic concepts appearing in the annotations is the one explained in Section 8.1. A total of 66,378 annotations were created. Table 8.5 describes the information gathered and annotated for each news section. A preliminary evaluation of the generated annotations is presented in Section 8.4.2.

News section	Retrieved/generated data		
	#news items	#annotations	Avg. #annotations/item
<i>Headlines</i>	2,660	18,210	7
<i>World</i>	2,200	17,767	8
<i>Business</i>	1,739	13,090	8
<i>Technology</i>	303	2,154	7
<i>Science</i>	346	2,487	7
<i>Health</i>	803	4,874	6
<i>Sports</i>	603	2,453	4
<i>Entertainment</i>	1,044	5,343	5
	9,638	66,369	7

Table 8.5 Average number of annotations per news item.

8.3 User profiles

Recent works show an increasing interest in using social tagging information to enhance personalised content retrieval and recommendation. *FolkRank* (Hotho, Jäschke, Schmitz, & Stumme, 2006) is a search algorithm that exploits the structure of folksonomies to find communities, and organise search results. The system presented in (Niwa, Doi, & Honiden, 2006) suggests web pages available on the Internet, by using folksonomy and social bookmarking information. The movie recommender proposed in (Szomszor, et al., 2007) is built on keywords assigned to movies via collaborative tagging, and demonstrates the feasibility of making accurate recommendations based on the similarity of item keywords to those of the user's rating tag-clouds.

News@hand also exploits folksonomy information to make collaborative recommendations, but in contrast to the above approaches, it makes use of a controlled ontological representation of social tags. Thus, the tags introduced by the users have to correspond to semantic concepts existing in the system domain ontologies. To do this, we provide two alternatives: a profile editor that allows searching and selecting semantic concepts in the ontologies, and an automatic mechanism that transforms freely-defined social tags into ontology concepts.

8.3.1 Manual definition of semantic preferences

The user profile editor of *News@hand* allows the users to manually create and update their semantic preferences. An ontology browser lets to explore the ontology hierarchies, easily search for concepts through on-line auto-complete widgets (Figure 8.6), and add selected concepts into the profile assigning weights to them.

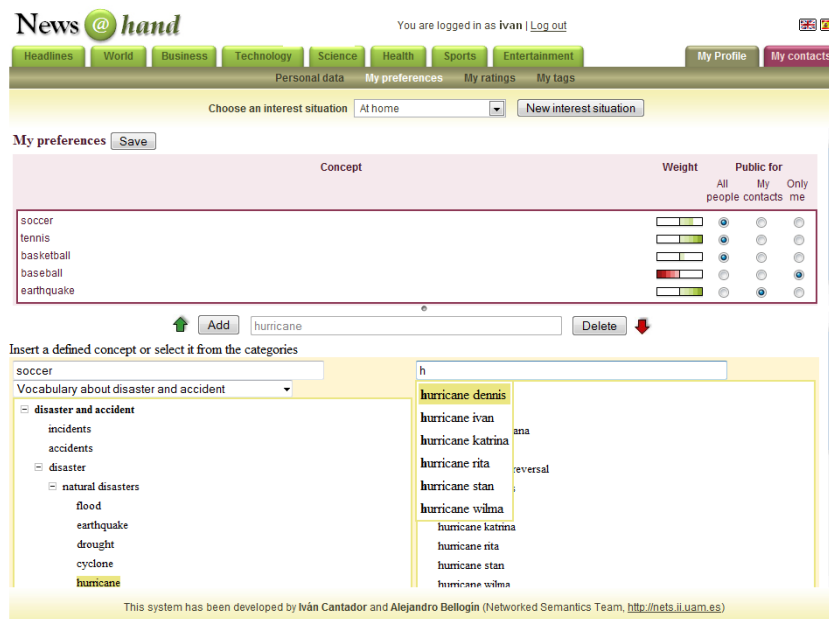


Figure 8.6 *News@hand* ontology browser with auto-complete search functionalities.

8.3.2 Automatic transformation of social tags into semantic preferences

Parallel to the proliferation and growth of social tagging systems, the research community is increasing its efforts to analyse the complex dynamics underlying folksonomies, and investigate the exploitation of this phenomenon in multiple domains. Results reported in (Cattuto, Loreto, & Pietronero, 2007) suggest that users of social systems share behaviours which appear to follow simple tagging activity patterns. Understanding, predicting and controlling the semiotic dynamics of online social systems are the basic pillars for a wide variety of applications.

For these purposes, the establishment of a common vocabulary (set of tags) shared by users in different social systems is a desirable situation. Thus, recent works have focused on the improvement of tagging functionalities to generate tag datasets in a controlled, coordinated way. For instance, *P-TAG* (Chirita, Costache, Handschuh, & Nejdl, 2007) is a method that automatically generates personalised tags for web pages, producing keywords relevant both to their textual content, and data collected from the user's browsing. In (Jäschke, Marinho, Hotho, Schmidt-Thieme, & Stumme, 2007), an adaptation of user-based collaborative filtering and a graph-based recommender is presented as a tag recommendation mechanism that eases the process of finding good tags for a resource, and consolidating the creation of a consistent tag vocabulary across users.

The integration of folksonomies and the Semantic Web has been envisioned as an alternative approach to the collaborative organisation of shared tagging information. The proposal presented in (Specia & Motta, 2007) uses a combination of pre-processing strategies, and statistical techniques, together with the exploitation of knowledge provided by ontologies, for making explicit the semantics behind the tag space in social tagging systems.

In the context of the ontology-based knowledge representation and recommendation models presented in this thesis, and integrated in *News@hand*, we propose the use of knowledge structures defined by multiple domain ontologies as a common semantic layer to unify and classify social tags from several Web 2.0²⁷ sites. More specifically, we propose a mechanism for the creation of ontology instances from gathered tags, according to semantic information collected from the Web. Tagging information is linked to ontological structures by our method through a sequence comprising three processing steps:

²⁷ Web 2.0 is a term which describes the trend in the use of WWW technology and web design that aims to enhance creativity, information sharing, and, most notably, collaboration among users. These concepts have led to the development and evolution of web-based communities and hosted services, such as social-networking sites, wikis, blogs and folksonomies.

- *Filtering social tags*: To facilitate the integration of information from different social sources as well as the subsequent translation of that information into ontological knowledge, a pre-processing of the tags is needed, associating them to a common vocabulary, shared by the different involved applications. Morphologic and semantic transformations of tags are performed at this stage based on the WordNet English dictionary (Miller, 1995), the Wikipedia encyclopaedia, and the Google²⁸ web search engine.
- *Obtaining semantic information about social tags*: The shared vocabulary is created with the use of Wikipedia, which provides semantic information about millions of concepts.
- *Categorisation of social tags into ontology classes*: Once the tags have been filtered and mapped to a shared vocabulary, they are automatically converted into instances of classes of domain ontologies. Semantic categorisation information available in Wikipedia is exploited in this process.

The second and third steps are the same to those performed in the ontology population strategy described in Section 8.1.2. For this reason, in the following, we only explain the first step.

Filtering social tags

Raw tagging information can be noisy and inconsistent. When manual tags are introduced with a non-controlled tagging mechanism, people often make grammatical mistakes (e.g., *barclona* instead of *barcelona*), tag concepts indistinctly in singular, plural or derived forms (*blog*, *blogs*, *blogging*), sometimes add adjectives, adverbs, prepositions, pronouns or verbs to the main concept of the tag (*beautiful car*, *to read*), or use synonyms and acronyms that could be converted into a single tag (*biscuit* and *cookie*, *ny* and *new york*). Moreover, the tag encoding and storage mechanisms used by social systems often alter the tags introduced by the users: they may transform white spaces (*san francisco*, *san-francisco*, *san_francisco*, *sanfrancisco*) and special characters in the tags (*los angeles* for *los ángeles*, *zurich* instead of *zürich*), etc.

Thus, while it is possible to gather information from multiple folksonomy sites, such as Flickr or del.icio.us, inconsistency will lead to confusion and loss of information when tagging data is compared. For example, if a user has tagged photos from a recent holiday in New York with *nyc*, but also bookmarked relevant pages in del.icio.us with *new_york*, the correlation will be lost.

In order to facilitate the folksonomy data analysis and integration, tags have to be filtered and mapped to a shared vocabulary. Here, we present a tag filtering architecture that makes use of different external knowledge resources such as the

²⁸ Google web search engine, <http://www.google.com/>

WordNet English dictionary, Wikipedia encyclopaedia, and Google web search engine. Broadly, the filtering architecture can be divided into four sections, as depicted in Figure 8.6:

- *Tag Reader*, which reads different social tagging datasets (e.g., from Flickr or del.icio.us), and converts them into an internal representation.
- *Tag Filtering Module*, comprising a number of subcomponents that are responsible for the different stages of the filtering process. These components can be split into two categories: morphological and semantic filters. Tags are maintained, merged or discarded according to the proposed filtering criteria.
- *External Resource Access Module*, providing a communication framework to the external knowledge resources.
- *Data Management Module*, supplying a database for tags, and managing the results from the various filtering steps.

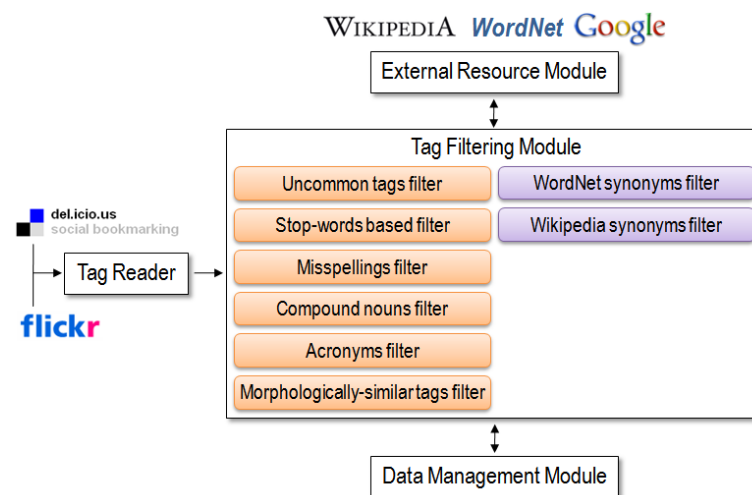


Figure 8.7 The tag filtering architecture.

The filtering process is a sequential execution where the output from one filtering step is used as input to the next. The output of the entire filtering process is a set of new tags (and their frequencies within the user profiles) that correspond to an agreed representation. As will be explained below, this is achieved by correlating tags to entries in two large knowledge resources: WordNet and Wikipedia. WordNet is a lexical database and thesaurus that groups English words into sets of cognitive synonyms called “synsets”, providing definitions of terms, and modelling various semantic relations between concepts: synonym, hypernym, hyponym, among others. Wikipedia is a multilingual, open-access, free-content encyclopaedia on the Internet. Using a wiki-style of collaborative content writing, it has grown to become one of the largest reference Websites with around 90,000 active contributors, maintaining

approximately 2,500,000 articles in over 250 languages (as of October 2008). Wikipedia contains collaboratively generated categories that classify and relate entries, and also supports term disambiguation and dereferencing of acronyms.

Figure 8.7 provides a visual representation of the filtering process where a set of raw tags are transformed into a set of filtered tags, and a set of discarded tags. Each of the numbers in the diagram corresponds to a step outlined below.

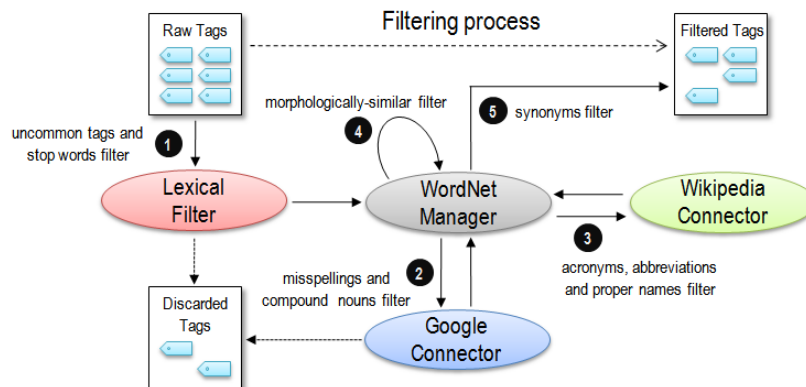


Figure 8.8 The tag filtering process.

For preliminary testing and input parameter setting, tags from public available user accounts from Flickr and delicio.us sites have been collected and filtered. A total of 1,004 user profiles have been gathered from these two systems, providing 149,529 and 84,851 distinct tags respectively. Initially, the intersection between both datasets was 28,550 common tags.

Step 1: Lexical filtering

After the raw tags are loaded by the *Tag Reader*, they are passed to the *Lexical Filter*, which applies several filtering operations. Tags that are too small (with length = 1) or too large (length > 25) are removed, resulting in a discarding rate of approximately 3% of the initial dataset. In addition, considering the discrepancies in the use of special characters (such as accents, dieresis and caret symbols), we convert such special characters to a base form (e.g., the characters à, á, â, ä, ã, å are converted to a), as shown in Table 8.7.

Tags containing numbers are also filtered based on a set of custom heuristics. For example, to maintain salient numbers, such as dates (2006, 2007, etc), common references (911, 360, 666, etc), or combinations of alphanumeric characters (7 up, 4 x 4, 35 mm), we discard unpopular tags below a certain global tag frequency threshold. Finally, common stop-words, such as pronouns, articles, prepositions and conjunctions are removed. After syntactic filtering, tags are passed on to the *WordNet Manager*. If a tag has an exact match in WordNet, we pass it on directly to the set of filtered tags, to save further unnecessary processing.

Pre-filtering	Post-filtering
á, à, â, ã, ä, å	a
é, è, ê, ë	e
í, ì, î, ï	i
ó, ò, ô, õ, ö, ø	o
ú, ù, û, ü	u
ý, ÿ	y
ç	c

Table 8.6 Conversion of special characters to a base form.**Step 2: Compound nouns and misspellings**

If a tag is not found in WordNet, we consider possible misspellings and compound nouns. Motivated by (Specia & Motta, 2007), to solve these problems, we make use of the Google “did you mean” mechanism. When a search term is entered, the Google engine checks whether more relevant search results are found with an alternative spelling. Because Google’s spell check is based on occurrences of all words on the Internet, it is able to suggest common spellings for proper nouns (e.g., names and places) that would not appear in a standard dictionary. By encapsulating a remote call to Google’s web service, our *Google Connector* corrects and filters misspelled tags.

The Google “did you mean” mechanism also provides an excellent way to resolve compound nouns. Since most tagging systems prevent users from entering white spaces into the tag value, users create compound nouns by concatenating nouns together or delimiting them with a non-alphanumeric character such as _ or -, which introduces an obvious source of complication when aligning folksonomies. By sending compound nouns to Google, we easily resolve the tag into its constituent parts). This mechanism works well for compound nouns with two terms, but is likely to fail if more than two terms are used. For example, the tag *sanfrancisco* is corrected to *san francisco*, but the tag *unitedkingdomsouthampton* is not resolved by Google.

We have thus developed a complementary novel algorithm that quickly and accurately splits compound nouns of three or more terms. The main idea is to firstly sort the tags in alphabetical order, and secondly process the generated tag list sequentially. By caching previous lookups, and matching the first shared characters of the current tag string, we are able to split it into a prefix (previously resolved by Google) and a postfix. A second lookup is then made using the postfix to seek further possible matches. The process is iteratively repeated until no splits are obtained from the *Google Connector*. Compared to a bespoke string-splitting heuristic, the proposed process has a very low computational cost. This mechanism successfully recognises long compound nouns such as *war of the worlds*, *lord of the rings*, and *martin luther king jr*. Figure 8.8 shows the pseudocode of the explained algorithm.

```

// Sort the tags alphabetically
sort(tags)

// Filter each tag
for each tag in tags {
    suggestion = Google.didYouMean(tag)

    // CASE 1: Compound noun
    if ( suggestion != tag AND suggestion.isCompoundNoun() ) {
        accept(tag, suggestion)
        lastPrefix = suggestion.firstTerm()
        lastTag = tag
        lastSuggestion = suggestion
    }

    // CASE 2: Misspelling
    else if ( suggestion != tag AND !suggestion.isCompoundNoun() ) {
        if ( levenshteinDistance(tag, suggestion) <= 2 ) {
            accept(tag, suggestion)
        }

        // Possible compound noun
        else if ( tag.startsWith(lastPrefix) ) {
            newTag = tag.substring(lastPrefix)
            newSuggestion = Google.didYouMean(newTag)

            if ( levenshteinDistance(newTag, newSuggestion) <= 2 )
                accept(tag, lastPrefix + ' ' + newSuggestion)
            else
                discard(tag)
        }
        else {
            discard(tag)
        }
    }

    // CASE 3: Exact matching or keyword not found
    else {
        // Possible compound noun
        if ( tag.startsWith(lastPrefix) ) {
            newTag = tag.substring(lastPrefix)
            newSuggestion = Google.didYouMean(newTag)

            if ( levenshteinDistance(newTag, newSuggestion) <= 2 )
                accept(tag, lastPrefix + ' ' + newSuggestion)
            else
                accept(tag, suggestion)
        }

        // Possible compound noun of more than 2 tokens
        else if ( tag.startsWith(lastTag) ) {
            newTag = tag.substring(lastTag)
            newSuggestion = Google.didYouMean(newTag)
            accept(tag, lastTag + ' ' + newSuggestion);
        }
        else {
            accept(tag, suggestion)
        }
    }
} // end for

```

Figure 8.9 Pseudocode of the compound noun and misspelling detection mechanism.

Similarly to Step 1, after using Google to check for misspellings and compound nouns, the results are validated against the *WordNet Manager*. Unprocessed tags are added to the pending tag stack, and unmatched tags are discarded.

Step 3: Wikipedia correlation

Many of the popular tags occurring in community tagging systems do not appear in grammar dictionaries, such as WordNet, because they correspond to proper names (such as famous people, places, or companies), contemporary terminology (such as *web2.0* and *podcast*), or are widely used acronyms (such as *asap* and *diy*).

In order to provide an agreed representation for such tags, we correlate tags to their appropriate Wikipedia entries. For example, when searching the tag *nyc* in Wikipedia, the entry for New York City is returned. The advantage of using Wikipedia to agree on tags from folksonomies is that Wikipedia is a community-driven knowledge base, much like folksonomies are, so that it rapidly adapts to accommodate new terminology.

Apart from consolidating agreed terms for the filtered tags, our *Wikipedia Connector* retrieves semantic information about each obtained entry. Specifically, it extracts ambiguous concepts (e.g., “java programming language” and “java island” for the entry “java”), and collaboratively generated categories (e.g., “living people”, “film actors” and “American actors” for the entry “Brad Pitt”). This information is also exploited by the ontology population and semantic annotation processes already described in Sections 8.1.2 and 8.2.2.

Step 4: Morphologically similar terms

An additional issue to be considered during the filtering process is that users often use morphologically similar terms to refer to the same concept. One very common example of this is the no discrepancy between singular and plural terms, such as *blog* and *blogs*, and other morphological deviations (e.g., *blogging*).

In this step, using a custom singularisation algorithm, and the stemming functions provided by the Snowball library²⁹, we merge morphologically similar tags into a single tag. Figure 8.9 provides the pseudocode of the implemented algorithm. Firstly, the tags are reduced to their stem, base or root form. Then, those tags that share the same stem are grouped. Finally, for each group of similar tags, the shortest term found in WordNet is used as the representative tag of the group. If no term of a formed group is found in WordNet, the shortest term is selected as the group representative.

²⁹ Snowball string-handling language, <http://snowball.tartarus.org/>

```

// 1st step: singularisation and stemming
mappings = createHashTable()

for each tag in tags {
    singular = singularisation(tag)
    if ( singular != tag ) {
        mappings.put(tag, singular)
    }
    stem = stemming(singular)
    if ( stem != singular ) {
        mappings.put(singular, stem)
    }
}

// 2nd step: create groups of similar tags
groups = createGroupsWithTheSameMapping(mappings)

// 3rd step: set the representative term of each group
for each group in groups {
    representative = null
    foundInWordNet = false

    for each term in group {
        candidate = mappings.get(term)

        if( foundInWordNet = true ) {
            if ( WordNet.search(candidate) != null AND
                length(candidate) < length(representative) ) {
                representative = candidate
            }
        }
        else {
            if ( WordNet.search(candidate) != null ) {
                representative = candidate
                foundInWordNet = true
            }
            else if ( length(candidate) < length(representative) ) {
                representative = candidate
            }
        }
    }
}

```

Figure 8.10 Pseudocode of the morphologically similar term group technique.

Step 5: WordNet synonyms

When people communicate a certain concept, they often use synonyms, i.e., terms that have the same meaning, but with different morphological forms. A natural filtering step is the simplification of the tag sets by merging pairs of synonyms into single terms.

WordNet provides synonym relations between synsets of the terms. However, due to ambiguous meanings of the tags, not all of them can be taken into consideration, and the filtering process must be very carefully executed. Our merging process comprises three stages. In the first stage, a matrix of synonym relations is created by using WordNet. In the second stage, according to the number of synonym relations found for each tag, we identify the non-ambiguous synonym pairs, and finally, stage three replaces each of the synonym pairs by the term that is most popular. Examples of thus processed synonym pairs are *android* and *humanoid*, *thesis* and *dissertation*, *funicular* and *cable railway*, *stein* and *beer mug*, or *poinsettia* and *christmas flower*. Figure 8.10 shows the pseudocode of the proposed algorithm.

```
// 1st step: create the matrix of synonym relations
mappings = createHashTable()

synonyms = createMatrix(numTagsFoundInWordNet, numTagsFoundInWordNet)

for each tag in tagsFoundInWordNet {
    indexTag = getIndexOf(tag)
    tagSynonyms = WordNet.getSynonyms(tag)
    for each synonym in tagSynonyms {
        indexSynonym = getIndexOf(synonym)
        synonyms[indexTag][indexSynonym] = 1
        synonyms[indexSynonym][indexTag] = 1
    }
}

// 2nd step: find the non-ambiguous synonyms, i.e., those with only
// one '1' in their corresponding row/column of the synonyms matrix
synonymsPairs = createArray()

for each tag in tagsFoundInWordNet {
    indexTag = getIndexOf(tag)
    if( getNumberOfSynonyms(matrix, indexTag) = 1 ) {
        synonym = getSynonym(indexTag)
        synonymsPairs.add(tag, synonym)
    }
}

// 3rd step: replace the tags of each synonyms pair by that which is
// most popular
for each pair in synonymPairs {
    representative = getMostPopular(pair.get(1), pair.get(2))
    replace(pair.get(1), representative)
    replace(pair.get(2), representative)
}
```

Figure 8.11 Pseudocode of the WordNet synonym merging technique.

8.4 Experiments

In this section, we present an evaluation of the effectiveness achieved by our ontology population and item annotation mechanisms, tag filtering and matching strategies, and semantic-based recommendation models, once they have been integrated in *News@hand*.

8.4.1 Evaluation of the ontology population mechanism

In order to evaluate the ontology population process, we asked twenty users to randomly select, and manually assess thirty instances of each ontology. The users were undergraduate and PhD students of our department, half of them with experience on ontological engineering. They were requested to declare whether each instance was assigned to its correct class, to a less correct class but belonging to a suitable ontology, or to an incorrect class/ontology. Table 8.7 shows the average accuracy values for all the users considering correct class and correct ontology assignments.

Ontology	Average population accuracies			
	#classes	#instances	Class instantiation	Ontology instantiation
<i>Arts, culture, entertainment</i>	87	33,278	78.7	93.3
<i>Crime, law, justice</i>	22	971	62.7	73.3
<i>Disasters, accidents</i>	16	287	74.7	84.0
<i>Economy, business, finance</i>	161	25,345	69.3	80.0
<i>Education</i>	20	3,542	57.5	76.7
<i>Environmental issues</i>	41	20,581	72.0	85.3
<i>Health</i>	26	1,078	65.3	89.3
<i>Human interests</i>	6	576	64.0	84.0
<i>Labour</i>	6	133	70.7	78.7
<i>Lifestyle, leisure</i>	29	4,895	72.0	90.7
<i>Politics</i>	54	3,206	60.0	81.3
<i>Religion, belief</i>	31	3,248	84.0	90.7
<i>Science, technology</i>	50	7,869	68.0	86.7
<i>Social issues</i>	39	8,673	70.7	85.3
<i>Sports</i>	124	5,567	72.0	86.7
<i>Unrests, conflicts, wars</i>	23	1,820	61.3	80.0
<i>Weather</i>	9	66	69.7	89.5
	744	121,135	69.9	84.4

Table 8.7 Average class and ontology population accuracies.

These preliminary results demonstrate the feasibility of our ontology population mechanism. The average accuracy for class assignment is 69.9%, and the average accuracy for ontology assignment arises to 84.4%. Improvements in our mapping heuristics can be investigated. Nevertheless, we presume they are good enough for our recommendation goals. In general, the main common concepts are correctly instantiated, and the effect of an isolated incorrect annotation in a news item is mitigated by the domain/s of the rest of the correct annotations.

8.4.2 Evaluation of the item annotation mechanism

For two months we were daily gathering RSS feeds. A total of 9,698 news items were stored. For this dataset, we run our semantic annotation mechanism, and a total of 66,378 annotations were created. Table 8.8 shows a summary of the average number of annotations per news item generated with our system. Similarly to the experiments conducted for our ontology population strategy, we asked twenty students to evaluate the annotations created for ten randomly selected news items from each of the 8 news sections of *News@hand*, giving ratings with values from 0 to 10. The annotation accuracies for each topic are also presented in Table 8.8. An item was considered to be correctly annotated if it received a user rating greater than 5.

News section	Retrieved/generated data			
	#news items	#annotations	Avg. #annotations/item	Avg. accuracy
<i>Headlines</i>	2,660	18,210	7	71.4
<i>World</i>	2,200	17,767	8	72.7
<i>Business</i>	1,739	13,090	8	79.2
<i>Technology</i>	303	2,154	7	76.3
<i>Science</i>	346	2,487	7	74.1
<i>Health</i>	803	4,874	6	73.1
<i>Sports</i>	603	2,453	4	75.8
<i>Entertainment</i>	1,044	5,343	5	76.0
	9,698	66,378	7	74.8

Table 8.8 Average number of annotations per news item, and annotation accuracies.

An average annotation accuracy of 74.8% was obtained. During the experiment, we found out that further improvements can be done in the annotation process. The main problem we noticed is the lack of term disambiguation in this step. The evaluators identified items with “duplicated” instances, having the same name but different URIs (i.e., belonging to different ontology classes). Several sources of information could be exploited to attempt to disambiguate annotations, such as co-occurrences of terms within news contents and ontology concepts.

8.4.3 Methodology for evaluating the recommendation models

Chapter 6 gave independent empirical evaluations of our semantic group-oriented and multilayer hybrid recommendation mechanisms. In this chapter, making use of *News@hand* platform, we complement those experiments with evaluations of the personalised and context-aware recommendation strategies.

After integrating all the models in *News@hand*, we conducted experiments combining and/or comparing the above approaches. These new evaluations were carried out with real users in search and recommendation scenarios which are similar to a natural environment. In the next subsections, we present the experiments in detail. But before that, we identify the evaluation cases or tests that should be investigated in order to cover the validation of the above recommendation functionalities. These functionalities are alternately activated and deactivated, in order to discriminate, observe and measure the effect of each other separate from the rest. We also list general steps that should be followed in the identified evaluation cases.

Activation/deactivation of functionalities

Excluding group-oriented recommendation and preference learning mechanisms³⁰, the following are identified as the significant comparisons to be investigated in order to properly assess the performance of the personalisation and recommendation functionalities.

- **Test 1.** Evaluation of personalised ontology-based content retrieval (activating the semantic expansion mechanism) against a keyword-based approach.
- **Test 2.** Evaluation of the semantic context-aware content retrieval approach within the personalised ontology-based model.
- **Test 3.** Evaluation of the hybrid recommendation approach against the content-based technique.
- **Test 4.** Evaluation of the hybrid recommendation approach against a collaborative filtering technique.

Table 8.9 indicates the involved functionalities in each of the four proposed testing cases. If we consider the basic content-based approach (i.e., without semantic expansion) as a form of simple keyword-based technique, cases 1 and 3 have already

³⁰ Using *News@hand*, we discarded the study of the group-oriented recommendation model because the current version of the system focuses on the automatic single user-oriented presentation of news contents. The preference learning module was not evaluated either. As mentioned in Section B.6, this functionality is already integrated in the system, but is out of the scope of this thesis.

been fulfilled by the experiments presented in Section 6.2. Case 4 was also tested in Section 6.3, but using a synthetic dataset of semantic user profiles. In the rest of this section, we mainly deal with cases 2 and 4 using user profiles manually created in *News@hand*.

		Personalisation functionalities			Recommendation functionalities	
		Keyword-based content retrieval	Ontology-based content retrieval	Semantic context-aware personalisation	Collaborative filtering	Hybrid recommendation
Evaluation of personalisation	1	X	X			
	2		X	X		
Evaluation of collaborative recommendation	3		X			X
	4				X	X

Table 8.9 Functionalities to be evaluated in each testing case.

Execution of evaluation tasks

We propose an experimental approach where every user performs several personalisation and recommendation tasks. Each pair of tasks is aimed to evaluate a specific testing case. A user does not have to deal with all the testing cases, but only a subset evenly distributed (according to latin square design) so that users and tasks do not introduce any bias in the performance of the different configurations. An average result is finally obtained for each evaluation case from the corresponding tasks performed by the users.

The following is a general scheme about how the experimentation has to be conducted.

- N specific search tasks are defined. We suggest $N \geq 6$.
- Each user performs $2M$ tasks (with $M \leq N/2$). We set $M = 2$.
- The tasks of each user will be used to evaluate M testing cases: a pair of tasks addresses a specific case activating or deactivating the involved functionalities.
- We could take into consideration only $M-1$ cases per user, if the results of the first case (i.e., the first two tasks) were omitted. We may presume that the first case is not valid because the user has to learn how to use the application.
- Average precision/recall results are measured for each case.

The experiments described in the next subsections have been designed following the proposed evaluation methodology. The definition of the tasks and the computation of the precision/recall values will be different depending on which recommendation functionality is tested.

8.4.4 Evaluating personalised and context-aware recommendations

We conducted an experiment to evaluate the precision of the personalisation and context-aware recommendation functionalities available in *News@hand* (explained in Sections 4.2 and 4.3). With this experiment we also wanted to investigate the influence of each mechanism in the integrated system, measuring the precision of the recommendations when a combination of both models is used.

The experiment was done with sixteen subjects, recruited among members of our department. In this case, they were undergraduate/graduate students, and lecturers. The experiment consisted of two phases, each composed of two different tasks.

- In the first phase, only the personalisation module was active, and its tasks were different in having the semantic expansion enabled or disabled.
- In the second phase, the contextualisation and semantic expansion functionalities were active. On its second task we also enabled the personalised recommendations.

Search tasks

A task was defined as finding out and evaluating those news items that were relevant to a given goal. Each goal was framed in a specific domain. We considered three domains: telecommunications, banking and social care issues. For each domain, a user profile and two search goals were manually defined (see below). Table 8.10 shows a summary of the involved tasks.

Domain	Section	Query		Task goal
Telecommunications	World	Q _{1,1}	pakistan	News about media: TV, radio, Internet
	Entertainment	Q _{1,2}	music	News about software piracy, illegal downloads, file sharing
Banking	Business	Q _{2,1}	dollar	News about oil prices
	Headlines	Q _{2,2}	fraud	News about money losses
Social care	Science	Q _{3,1}	food	News about cloning
	Headlines	Q _{3,2}	internet	News about children, young people, child safety, child abuse

Table 8.10 Summary of the search tasks performed in the experiment.

To simplify the searching tasks, they were defined for pre-established sections and queries. For example, the task goal of finding news items about software piracy, illegal downloads and file sharing, $Q_{1,2}$, was reduced to evaluate those articles existing in *Entertainment* section that were retrieved from the query “music”.

Table 8.11 shows the tasks performed by the sixteen users. The configuration and assignment of the tasks were set according to the following principles:

- A user should not repeat a query during the experiment.
- The domains should be equally covered by each experiment phase.
- A user has to manually define a user profile once in the experiment.

User	Personalised recommendations		Context-aware recommendations	
	Without expansion	With expansion	With expansion	
	$w_p=1$ $w_c=0$	$w_p=1$ $w_c=0$	$w_p=0$ $w_c=1$	$w_p=0.5$ $w_c=1$
1	* $Q_{1,1}$	$Q_{2,1}$	$Q_{3,1}$	$^A Q_{1,2}$
2	$Q_{2,2}$	* $Q_{3,2}$	$^A Q_{2,1}$	$Q_{1,2}$
3	$Q_{3,1}$	$^A Q_{3,2}$	* $Q_{1,1}$	$Q_{2,1}$
4	$^A Q_{1,1}$	$Q_{1,2}$	$Q_{2,2}$	* $Q_{3,2}$
5	$Q_{1,2}$	* $Q_{2,2}$	$Q_{3,2}$	$^A Q_{2,1}$
6	$Q_{2,1}$	$Q_{3,1}$	* $^A Q_{3,2}$	$Q_{1,1}$
7	$Q_{3,2}$	$^A Q_{1,1}$	$Q_{1,2}$	* $Q_{2,2}$
8	* $^A Q_{2,2}$	$Q_{1,1}$	$Q_{2,1}$	$Q_{3,1}$
9	$Q_{1,1}$	$Q_{2,1}$	* $Q_{3,1}$	$^A Q_{3,2}$
10	$Q_{2,2}$	$Q_{3,2}$	$^A Q_{1,1}$	* $Q_{1,2}$
11	* $Q_{3,1}$	$^A Q_{2,2}$	$Q_{1,1}$	$Q_{2,1}$
12	$^A Q_{3,1}$	* $Q_{1,2}$	$Q_{2,2}$	$Q_{3,2}$
13	$Q_{1,2}$	$Q_{2,2}$	$Q_{3,2}$	* $^A Q_{1,1}$
14	* $Q_{2,1}$	$Q_{3,1}$	$^A Q_{2,2}$	$Q_{1,1}$
15	$Q_{3,2}$	* $^A Q_{3,1}$	$Q_{1,2}$	$Q_{2,2}$
16	$^A Q_{1,2}$	$Q_{1,1}$	* $Q_{2,1}$	$Q_{3,1}$

Table 8.11 Experiment tasks configurations.

For each phase, the combination of personalised and context-aware recommendations was established as a linear combination of their results using two weighs $w_p, w_c \in [0,1]$:

$$\text{score}(i_n, u_m) = w_p \cdot \text{pref}(i_n, u_m) + w_c \cdot \text{pref}(i_n, u_m, \text{context}).$$

In the personalisation phase, the contextualisation was disabled (i.e., $w_c = 0$). Its first tasks were performed without semantic expansion, and its second tasks had the semantic expansion activated. In the contextualisation phase, w_c was set to 1 and the expansion was enabled. Its first tasks were done without personalisation ($w_p = 0$), and its second tasks were a bit influenced by the corresponding profiles ($w_p = 0.5$).

User profiles

As mentioned before, fixed user profiles were used for each domain. Some of them were common predefined profiles, and others were created by the users (those marked with ‘*’ in Table 8.11) during the experiment using the profile editor of *News@hand*. In addition, some tasks were done with user profiles containing concepts belonging to all the three domains. They are marked with an ‘A’ in the table.

Table 8.12 lists those concepts included in the predefined domain-driven user profiles. Each domain was described with six semantic concepts, appearing in a significant number of item annotations. Note that each domain may be described by concepts belonging to different ontologies, and may be covered with news items of different news sections.

Domain	Concepts
Telecommunications	internet, network, satellite, technology, telecommunications, website
Banking	bank, banking, business, economy, euro, dollar
Social care	drug, health, immigration, safety, social abuses, terrorism

Table 8.12 Topics and concepts allowed for the predefined user profiles in the evaluation of personalised and context-aware recommenders.

Analogously to the predefined user profiles, those manually created by the evaluators using the profile editor of *News@hand* contained semantic concepts of the above three domains. However, in this case, the evaluators were free to select their preferences from concepts available in the entire system KB. No restriction was placed on the number, type (classes or instances) and ontology of the concepts. Table 8.13 shows the concepts included for each domain, and the average size (in number of preferences per user) of the sixteen profiles. For instance, in *Telecommunication* domain, 55 preferences were declared using 30 different semantic concepts, producing an average of 3.4 preferences per user. On average, each profile

contained 3.2 preferences of each domain.

Domain	Concepts	#preferences	Avg. #pref./user
Telecommunications (30 concepts)	blackberry, cell phone, computer programming, computer sciences, computing and information technology, digital voice, email, encryption, file sharing, free downloads, internet, internet history, mobile network operator, network theory, networks, router, search engine, signal processing, social search, software, technology, telecommunications, television, tfidf, video arcade, video call, video game, voice over internet, web crawler, web search	55	3.4
Banking (25 concepts)	bank, bank charges, bank machine, bank of america, banker, banking, business, cash, credit card, dollar, economy, euribor, euro, euro interbank offered rate, finance, foreign exchange market, funds, ibank, macroeconomics, microfinance, money, payment system, stock, stock broking, trade policy	46	2.9
Social care (26 concepts)	abstinence, abuse, adoption, charity, children, civil society, drug, drug trafficking, family, gay, health, homophobia, homosexuality, immigration, pornography, safety, sexuality, smoking, social abuses, social change, social development, social groups, teenagers, terrorism, victims, volunteerism	51	3.2

Table 8.13 Topics and concepts of the manually-defined user profiles in the evaluation of personalised and context-aware recommenders.

Steps for the evaluation of the personalised recommendation

The objective of the two tasks performed in the first experiment phase was to assess the importance of activating the semantic expansion in our recommendation models. The following are the steps the users had to do in these tasks.

- Launch the query with the personalisation module deactivated.
- Rate the top 15 news items. The allowed rating values were: 1 if the item was not relevant to the task goal, 2 if the item was relevant to the task goal, and 3 if the item was relevant to the task goal and the user profile. These ratings are considered as our baseline case.
- Launch the query with the personalisation module activated (and the semantic expansion enabled/disabled depending on the case).

- Rate the new top 15 news items as explained before. If a news item had previously been rated, rate it again with the same value.

Steps for the evaluation of the context-aware recommendation

The objective of the two tasks performed for the second experiment phase was to assess the quality of the results when the contextualisation functionality is activated and combined with personalisation. The steps done in this phase are the following:

- Launch the query with the contextualisation deactivated.
- Rate the top 15 news items as explained before, and evaluate as relevant (clicking the title) the first two items which were related to the task goal. Doing this the current semantic context is updated.
- Launch the query with the contextualisation activated (semantic expansion enabled, and personalisation enabled/disabled depending on the case).
- Rate again the top 15 news items as explained before. If a news item had previously been rated, rate it again with the same value.

Results

Once the two evaluation phases were finished, we computed the precision values for the top $N = 5, 10, 15$ news items as follows:

$$P@N = \frac{\#\{\text{relevant items in the top } N \text{ news items}\}}{N}.$$

Figures 8.12 shows the average results for the sixteen users, taking into account those items evaluated as relevant to the task goal, and also the user profile.

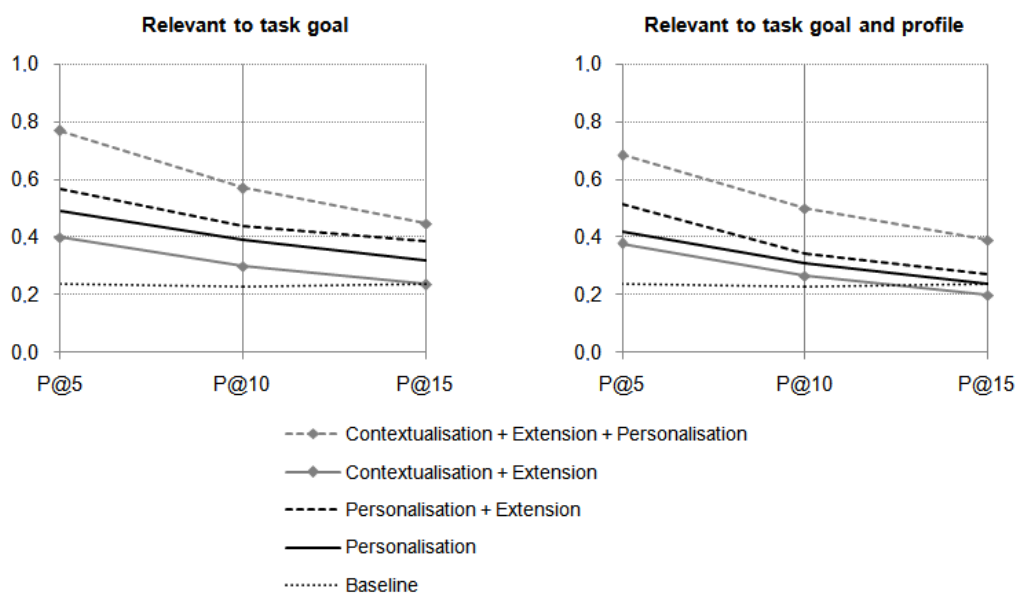


Figure 8.12 Average precision values for the top 5, 10 and 15 news items, taking into account those items evaluated as relevant to the task goal and the user profile.

In both cases, the recommendation models outperformed the baseline case, especially for the five top items. The $P@5$ values increased from 20% of the baseline case to almost 40% and 50% when contextualisation and personalisation functionalities were enabled. The semantic expansion seemed to be an essential component within the recommendation processes. It provided an improvement of 10% in the personalisation precision. Finally, the combination of personalised and context-aware recommendations (plus semantic expansion) gave the best results, achieving a $P@5$ value of 80%.

Apart from the computation of the precision values, we also asked the evaluators to provide comments and suggestions about the system. The most remarkable feedback we obtained can be summarised in the following points:

- **The contextualisation of recommendations is a useful functionality.** The users noticed and positively assessed how news items relevant to the current search goal move up to the top positions of the ranked lists when the context-aware recommender is activated.
- **A disambiguation mechanism should be included within the annotation process.** The users found out semantic annotations whose terms appeared in their profiles but having different meanings. This not only worsened the generated recommendations, but also the users' evaluations.
- **A collaborative approach to enrich the semantic profiles may be beneficial.** Several users declared some preferences assuming that related ones (e.g., synonyms) were going to be implicitly taken into account. A mechanism to exploit co-occurrences among preferences of different users could be useful to automatically add related semantic concepts into the semantic profiles.
- **The incorporation of a user preference recommender would be helpful.** Despite the facilities offered by the ontology browser and the auto-complete concept search boxes of *News@band*, several users missed the fact of having concept suggestions (e.g., in the form of “related preferences are...”) when they had to create their profiles.

8.4.5 Evaluation of hybrid recommendations

A second experiment was conducted with *News@band* to evaluate the semantic multilayer hybrid recommenders. As the experiment explained in Section 6.3, which merged and exploited information from MovieLens and IMDb repositories, the objective of this experiment was to compare the recommendations provided by our

hybrid models with those obtained using a classic collaborative filtering approach.

Again, an off-line execution of the recommendation strategies over a set of user profiles and ratings was performed in order to compute accuracy measures. However, in this case, users were asked to provide such information using the system.

The sixteen members of our department who participated in the previous experiment were again requested to take part of the evaluation presented herein. Three phases were followed by each user, assessing news recommendations for three news sections: *Business*, *Sports* and *World* (see below why we selected these sections). For each phase, two tasks were defined:

- In the first task, the users had to rate a number of news items from a random list.
- In the second task, the users had to rate several news items from a list generated with the personalisation functionality activated.

Search tasks

A task was defined as finding out and rating those news items that were “related to” a personal user profile. By “related to” we mean that a news item contains semantic annotations whose concepts appear in the user’s profile.

Note that a concept could be assigned negative or positive weights within a profile, so the evaluation of an item might have a low (close or equal to 1 star) or a high (close or equal to 5 stars) rating values.

User profiles

Similarly to the experiment described in Section 8.4.4, the evaluators were asked to choose their preferences. However, in this case, they could only select preferences from a given list of semantic concepts. They were provided a form with a list of 128 semantic concepts, classified in 8 different domains. From this list the users had to select a subset of concepts, and assign them negative/positive weights according to personal interests. Table 8.14 shows the concepts available for each domain, and the average number of preferences per user.

On average, each profile was created with 7.8 preferences per domain, duplicating the preferences introduced by the users when they had to manually search the concepts in the ontology browser (see Section 8.4.4).

Once the user profiles were created, we identified which news sections contained news items annotated with the most popular (i.e., the most used) preferences. The goal was to define an item set from which the recommenders could provide a significant number of personalised recommendations. Finally, we selected the news

sections mentioned previously: *Business*, *Sports* and *World*.

Domain	Concepts	#preferences	Avg. #pref./user
Computers Technology Telecommunications	computer, digital, ebay, google, ibm, internet, mass, media, microsoft, networking, online, satellite, software, technology, video, website	135	8.4
Wars Armed conflicts	al-qaeda, army, battle, combat, crime, kidnapping, kill, memorial, military, murder, peace, prison, strike, terrorism, war, weapons	104	6.5
Social issues	aids, assassination, babies, children, death sentence, divorce, drugs, family, health, hospital, immigration, love, obesity, smoking, suburb, suicide	115	7.2
Television Cinema Music	actor, bbc, cinema, cnn, film, grammy, hollywood, movie, music, musician, nbc, radio, rock, oscar, singer, television	129	8.1
Sports	baseball, cricket, football, lakers, nascar, nba, new england patriots, new york giants, nfl, olympics, premier league, running, sports, soccer, super bowl, tennis	168	10.5
Politics	george bush, condolezza rice, congress, democracy, elections, government, hillary clinton, john maccain, barack obama, parliament, politics, president, senate, senator, voting, white house	104	6.5
Banking Economy Finance	banking, business, cash, companies, earnings, economy, employment, finance, fraud, gas price, industry, marketing, markets, money, oil price, wall street	120	7.5
Climate Weather Natural disasters	air, climate, earth, earthquake, electricity, energy, fire, flood, forecast, fuel, gas, pollution, sea, storm, weather, woods	128	8.0

Table 8.14 Topics and concepts allowed for the user profiles in the evaluation of the hybrid recommenders.

Steps for the evaluation of the collaborative filtering and hybrid recommendations

As mentioned before, the users had to perform three tasks, each of them in one of the following news sections: *Business*, *Sports* and *World*. Successively, for each section, a user had to:

- Deactivate the personalisation functionality, and display the news items of the section. The goal is to present to all the users the same set of news items, in order to obtain a “shared” group of rated items.
- Rate 20 news items that are related (with negative or positive weights) to the user profile. Taking into account the similarities between item annotations with user preferences, assign a 1-5 star rating to the selected news items. No restriction is placed on which items have to be rated.
- Activate the personalisation functionality, and display again the news items of the section. This time the order (ranking) of the news items is different to the one shown previously. The goal here is to present to each user a set of news items that might be related to his semantic profile. Thus, content-based similarities could be found among profiles of different users.
- Rate (as explained before) 50 news items not evaluated previously.

With this strategy, the sixteen users provided a total of 3,360 ratings for 859 different news items.

Results

The purpose of the experiment was to compare the accuracy values obtained with our semantic multilayer hybrid recommendation model UP- q , with those achieved by a classic item-based collaborative filtering strategy.

Analogously to previous evaluations already presented in the thesis, in this experiment, we computed the accuracy of the recommendations using different percentages of the user ratings to build (train) and evaluate (test) off-line the models. In this case, we computed the MSRE (defined in Section 2.6) between the actual ratings introduced by the users, and the values predicted by the above recommenders. Figure 8.13 shows separately the average results for the items belonging to the three considered news sections.

In *Business* and *World* sections, the accuracy values of both models seem to be very similar. For the *World* section, the UP- q strategy performs slightly better than CF when 10% to 50% of the ratings were used to build the recommenders. For the *Business* section, however, there is no significant difference. Checking the news items profiles, we noticed that there was a relative small number of semantic annotations about banking, economy and finance. This could be the reason of having such results with our semantic-based approach.

In *Sports* section, the UP- q model provides better recommendations for almost of the training rating levels. The user profiles created with semantic concepts of this domain were richer, facilitating the discovery of similarities among the user interests.

In the general case where items of the three sections were taken into account, the hybrid model seems again to give more accurate recommendations when few ratings are available. Specifically, utilising 10%, 20% and 30% of the rating information, the UP- q error is lower than the error obtained with the CF strategy.

Once more, the hybrid recommender seems to successfully address the related cold-start and sparsity problems successfully. With the obtained results, we presume that a combination of CF and our semantic-based recommendation model could be an optimal solution. In fact, this is the approach that is executed in *News@hand* when the collaborative recommendation functionality is enabled.

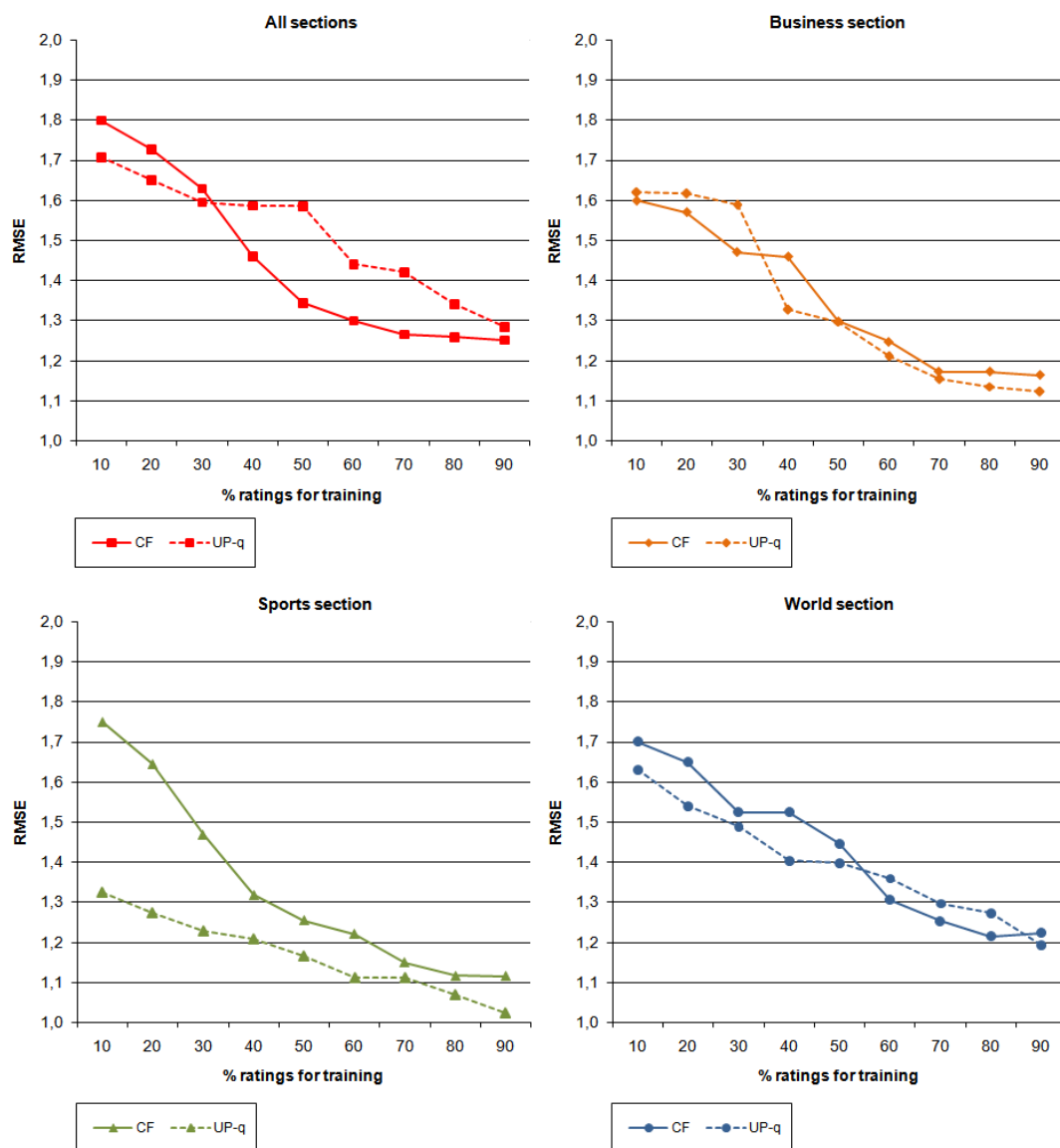


Figure 8.13 Average Mean Squared Error of item-based collaborative filtering (CF) and semantic multilayer hybrid (UP- q) recommendation strategies using

10%, 20%, ..., 90% of the available ratings for building (training) the models, and the rest for testing.

Apart from the computation of accuracy metrics, we gathered more subjective assessments of the system. We asked the evaluators to provide us comments about the recommendations obtained during the experiment. The most remarkable observations were the following:

- **Very similar news items were closely shown.** The non-diversity problem (see Section 2.2.1) has not been addressed in this thesis. In the current version of the system, a certain news item can be retrieved from different RSS sources, and might be recommended to the user several times. Various users did not rate some news items because they had already evaluated very similar ones.
- **A disambiguation mechanism should be included within the annotation process.** As noticed in the evaluation of the personalised and context-aware recommenders (Section 8.4.4), the users found out semantic annotations with wrong meanings.
- **The contextualisation of recommendations is a desirable functionality even when collaborative item suggestions are provided.** Several users missed the activation of the context-aware recommender for this experiment. They also suggested us to consider additional sources of context, such as the semantics of news items linked through spatial (location) and temporal relations.
- **The rating of news items according to the user profile seemed to be difficult in some cases.** Several users found difficult to rate some news items because they could not easily distinguish between interesting and pleasant-reading articles.

8.4.6 Evaluation of recommendations using semantic preferences obtained from social tags

Preliminary experiments have been conducted to evaluate our personalised content-based recommender when it is executed with semantic preferences obtained from social tags, and through the mechanism explained in Section 8.3.2.

Twenty experimenters were requested to evaluate news recommendations according to 10 semantic profiles obtained from the 1,004 Flickr and del.icio.us tag sets introduced in Section 8.3.2. Running the personalised recommendation algorithm of *News@hand* with the assigned 10 user profiles, each evaluator had to assess the 5 top ranked news items of the 8 news sections, specifying whether a

recommended item would be relevant or not for the 10 anonymous users according to their semantic profiles. The 10 profiles assigned to each evaluator were randomly selected from the original tag sets.

Table 8.15 shows the average results for the twenty experimenters. Each value represents the percentage of evaluated news items that were marked as relevant. The results of our ontology-based approach are compared with those obtained with a classic keyword-based content retrieval algorithm, which computes cosine similarities between item annotations and tag-based user profiles. Although more significant experimentation has to be performed, our approach to recommending items based on semantic transformations of social tags provides acceptable results, achieving an average relevance accuracy close to 70%. Analogously to previous experiments, in this evaluation we noticed the need of incorporating a semantic disambiguation mechanism within the annotation process in order to improve the recommendations.

News section	Personalised recommendations	
	Keyword-based content retrieval	Ontology-based content retrieval
<i>Headlines</i>	46.3	57.0
<i>World</i>	34.3	53.2
<i>Business</i>	39.0	72.8
<i>Technology</i>	43.5	94.0
<i>Science</i>	35.9	60.9
<i>Health</i>	21.1	40.6
<i>Sports</i>	58.0	98.2
<i>Entertainment</i>	33.5	60.4
	39.0	67.1

Table 8.15 Average relevance values for the top 5 ranked news items recommended by *News@hand* when using semantic profiles obtained from social tags.

The reduction on the size of the user profiles when they are transformed from social tag clouds into semantic concept sets is illustrated in Figure 8.14.

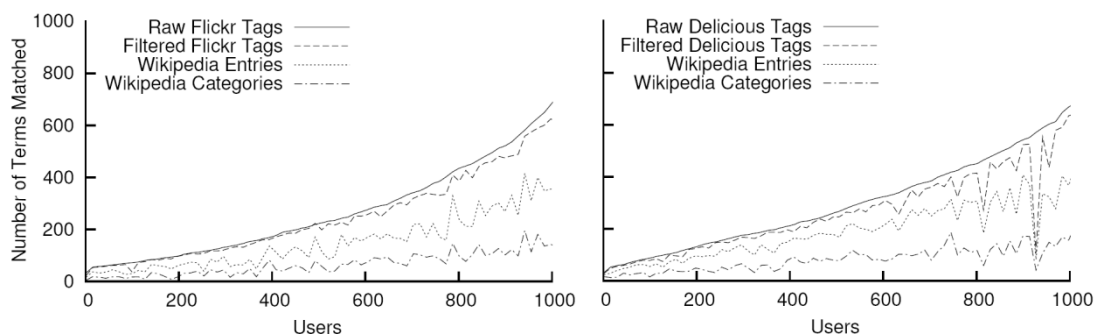


Figure 8.14 Matching Flickr and del.icio.us tags to Wikipedia ontology. Graphs show how many tags each user had in the raw tag cloud, how many tags were filtered, how many corresponded to a Wikipedia entry, and finally how many categories were selected to represent the given tag cloud.

The percentage of matched Wikipedia entries conform approximately the 50% of the original social tags. However, this does not correspond to the final size of the user profiles. The graphs were plotted considering the terms matched with Wikipedia entries, but not with those that were previously found in WordNet, and did not have to be matched with Wikipedia concepts.

8.5 Conclusions

News@hand, our on-line news recommender system, has allowed us to evaluate the semantic content-based and collaborative recommendation models presented in this thesis, executing them in parallel and combining their output recommendations.

The obtained results have reinforced conclusions that were previously observed, and have provided additional findings which could not be detected by isolated evaluations of the models. The personalised recommendations help the users to find relevant news articles, and the semantic expansion of user preferences eases the matching between user and item profiles, improving precision values for the top suggested items, and mitigating the well-known cold-start and sparsity problems. The incorporation of contextualisation within the personalisation mechanism speeds up the discovery of items related to current search goals, and has been highly appreciated by the users. Finally, the consideration of layer hybrid recommendations seems to enhance collaborative approaches when partial (interest-focused) comparisons of user profiles are computed. The establishment of relations among users at multiple interest layers reduces the effect of the grey sheep problem.

In addition to these conclusions, the implementation of a novel recommender system which is based on the semantic representation of user preferences and item content features has forced us to confront interesting and diverse research challenges. Firstly, we had to build from scratch a knowledge base comprising different domains. For that purpose, we have proposed an automatic ontology population mechanism that exploits semantic information from several public information sources such as WordNet and Wikipedia. Next, we had to annotate news contents with classes and instances existing in the domain ontologies. To do that, we have developed an automatic semantic annotator that makes use of NLP tools to analyse and process texts, retrieving their semantic concepts. Finally, we had to make it easier for users to create their semantic profiles. We have presented a mechanism that automatically transforms social tags into semantic concepts of a given set of ontologies.

The experimentation done has also provided us the opportunity of getting feedback from the users about the system functionalities and outputs. Among other issues, they realised the need of incorporating a semantic disambiguation step in the annotation process, and addressing of the non-diversity problem, as very similar (or even the same) news items were presented closely in the recommendation pages. Moreover, they suggested additional improvements in the personal profile editor, such as the integration of a real-time semantic preference recommender which takes into account concepts similar to the ones already introduced (synonyms, co-occurrences, etc.).

Chapter 9

Conclusions

Aiming to address limitations existing in current recommender systems, this thesis elaborates on the incorporation and exploitation of a conceptual space describing and connecting user preferences and item contents in a general way. Building upon this view, the following specific goals are set out:

- The definition of a formal (ontology-based) knowledge model, supporting the expression of explicit semantic relations between concepts.
- The design of flexible content-based models, allowing the contextualisation of the recommendations, and their extension to multiple users.
- The design of hybrid models, drawing further benefit from collaborative filtering approaches.
- Building a recommender system, allowing the joint evaluation of all the above proposals.

In the first part of the thesis, we reviewed and related the two research fields in which this work is framed: recommender systems, and semantic-based information representation and retrieval. In the second part of the thesis, we presented our knowledge representation and recommendation model proposals, and we reported on two independent sets of experiments where the models are evaluated in controlled scenarios with a few user profiles, and with larger synthetic datasets. Finally, in the third part of the thesis, we described the developed recommender system, which was used to conduct further evaluations in a more realistic setting (complementary to the lab experiments). An additional purpose of this experience was to check, face, and study, from a comprehensive perspective, the general feasibility, difficulties and limitations involved in the implementing a semantic-based system.

In this chapter, we present the conclusions and summarise the contributions achieved in this research work (in Section 9.1), and we discuss the limitations of the proposals, along with future research directions to address them (in Section 9.2).

9.1 Summary and contributions

The final result of this thesis is a set of recommendation models building upon a rich semantic representation of the domain of discourse in order to address known problems and limitations of recommender systems. The proposed models are integrated and demonstrated in a recommender system, which relates tastes and interests of users for a wide range of items through an ontology-based knowledge representation. The semantic relations defined in the ontologies are used by the above strategies to provide recommendations which are oriented to single and multiple users, which take into account the current semantic context within the content retrieval process, and which, according to several layers of tastes and interests shared by the users, discover and exploit collaborative content-based relations among the user preferences.

In the next subsections, we motivate and summarise the proposals, and detail the achieved contributions, highlighting their benefits in comparison to other approaches reported in the literature.

9.1.1 Ontological knowledge representation

Content-based recommender systems (Lang, 1995; Pazzani & Billsus, 1997; Krulwich & Burkey, 1997; Mooney, Bennett, & Roy, 1998; Billsus & Pazzani, 1999) usually use term vectors (lists of weighted keywords) to describe the user preferences and item contents. Using term-based annotation and indexing techniques (e.g. TF-IDF approaches), and classic information retrieval algorithms (Salton & McGill, 1986; Baeza-Yates & Ribeiro Neto, 1999), such as the vector-space and probabilistic models (see Chapter 2), these systems compute similarities between user vectors and item vectors to provide an estimation measuring the potential interest of users for items.

This representation approach responds to the requirement of being efficiently processable, but entails a *loss of information* due to two main reasons. The first reason is related to the non-disambiguation of terms. A term can have several meanings, and the user might be interested in only one of them. Without taking into consideration the meaning of the term in each case, all the items where that term appears could be recommended to the user, whereas only some, those which do have the term with the meaning preferred by the user, would be relevant. The rest of the items would comprise wrong, not useful recommendations. The second reason is the term independence assumption. The fact of an item not having user interest terms explicitly does not necessarily imply that the item is not relevant for the user. Other related terms (by synonymy, hypernymy, hyponymy, etc., relations) could be taken into account to determine the importance of the item for the user.

The previous limitations imply that in most of the current recommender systems there is a:

Lack of understanding and exploitation of the semantics underlying the user tastes and interests, and the recommended item contents

To address this problem, we have proposed a knowledge representation in which both user profiles and item contents are described by means of vectors of concepts (classes and instances) that belong to one or more domain ontologies. In the vector associated to a user profile, each component is assigned a weight measuring the (positive or negative) interest that concept is predicted to elicit from the user. In an item annotation vector, the weight of each component reflects the degree in which the corresponding concept is relevant (informative) within the item contents and/or in comparison to the contents of the rest of the items.

The contribution of the thesis on this issue is:

The definition of a formal knowledge representation of user preferences and item contents, which is not ambiguous, and takes into account arbitrary (i.e., not pre-established) semantic relations between concepts.

The use of such a conceptual representation, in contrast to other common approaches based on keywords or items, offers the following benefits:

- *Semantic richness.* Preferences and annotations are more accurate, and reduce ambiguity. This enables a better understanding and exploitation of the meanings involved in personalised information retrieval and recommendation processes.
- *Hierarchical representation.* Ontological concepts are represented in a hierarchical way through standard relations such as “subClassOf” or “instanceOf”. Ancestors and descendants of a certain concept can provide additional valuable information about the semantics of the latter.
- *Inference.* Standard ontology definition languages, like RDF or OWL, support inference mechanisms for the discovery of knowledge that can be used to enhance the recommendations.

In addition to the characteristic benefits of an ontology-based representation, the proposal provides the following advantages, in comparison to classic recommendation models:

- *Portability.* Using XML-based standards, domain knowledge, item annotations, and user preferences can be easily distributed, adapted or integrated in different recommender systems for different applications.
- *Domain independence.* Regardless of the domain in which they may be applied, knowledge structures for user and item profiles consist of semantic networks

with interconnected concepts. Recommendation models, designed upon those structures, are built in a generic way, without any domain constraint.

- *Media independence.* Assuming the existence of manual or automatic semantic annotation mechanisms, recommendation models using the proposed knowledge representation can be used with no a priori restriction on the nature of items (texts, images, videos, audios, etc.).

Classic user profile representations based on keyword or rating lists are prone to suffer a **“shortage” of preferences**. In systems where preferences are set manually, users tend to not spend a lot of time creating their profiles, and in systems in which preferences are determined automatically from user action records, learning algorithms tend to recognise very generic user interests. This fact entails two main problems. The first problem is related to the *sparsity* of information in the knowledge structures used by the recommender systems, which makes it difficult to find similarities or correlations among users and items (Billsus & Pazzani, 1998; Sarwar, Karypis, Konstan, & Riedl, 2000). The second problem is the difficulty of recommending items to a new user when he begins to use the system, and has none or few declared preferences (Schein, Popescul, & Ungar, 2001). Apart from strategies that give the users an incentive to build their profiles, the two above problems can be addressed by techniques that extend or enrich the user profiles. Thus, we state the:

Need for enriching the user and item profiles

In order to satisfy this need, we have proposed a strategy which spreads the weights of the ontological concepts available in user and item profiles towards other concepts that are connected through semantic relations of the domain ontologies. The semantic propagation strategy proposed herein is based on CSA techniques (Cohen & Kjeldsen, 1987; Crestani, 1997), considering the attenuation of weights as the expansion grows away from the initial set, with loop control in the propagation paths, and the possibility to bound the expansion distance.

The contribution of the thesis in this area is:

The design of a novel mechanism which extends the semantic descriptions of user preferences and item contents through the ontological relations of the involved concepts.

The main direct benefits of the proposal are:

- *Mitigation of the sparsity problem.* By applying a semantic expansion, user and item profiles become larger, covering more areas of the conceptual space, and resulting in a higher likelihood of finding user and item similarities and correlations.

- *Coping with the cold-start problem.* The semantic expansion of new user profiles and item annotations eases their early incorporation and better exploitation in the recommendation processes. It may also be used as a complementary assistance for preference suggestion in the manual creation and edition of user profiles.

9.1.2 Semantic content-based recommendations

Current recommender systems are acknowledged to leave substantial room for improvement and extensions of their capabilities (Adomavicius & Tuzhilin, 2005). One of the most significant possible extensions is the ***contextualisation of recommendations*** (Räck, Arbanowski, & Steglich, 2006; Anand & Mobasher, 2007; Vallet, Castells, Fernández, Mylonas, & Avrithis, 2007). The context can be defined in many and very diverse ways:

- Based on facts directly related to the system, such as the last actions and evaluations done by the user, the current date and time, etc.
- According to information coming from other applications, such as the scheduled events of an electronic agenda, the last received messages in an email client, the favourite websites stored in a web browser, etc.
- Regarding external factors such as the current location, time, environment, companion, or mood of the user.
- Others.

In any case, the incorporation of context into the recommendation processes is known to be a complex task and, to some extent, is further hindered by a lack of flexibility in the content retrieval models.

Another relevant extension in recommender systems is the execution of ***group-oriented recommendations***. The suggestion of an item to a group of people is a desired feature that has been identified in multiple applications, such as the collective recommendations of musical compositions (McCarthy & Anagnost, 1998), movies (O'Connor, Cosley, Konstan, & Riedl, 2001), tourist attractions (Ardissono, Goy, Petrone, Segnan, & Torasso, 2003), or television shows (Ali & Van Stam, 2004). Again, traditional models are not flexible enough to hold such type of recommendations, and ad-hoc strategies, dependent on the application domain, have been proposed.

There are other possible enhancements (see Section 9.2), which in general and similarly to the two extensions explained before, are caused by the:

**Need for extensions in personalised recommendation models to provide
context-aware and group-oriented item suggestions**

Based on the proposed ontology-based user and item profile models, we have defined a personalised recommendation approach which is based on an adaptation of the vector-space IR model. In this proposal, the user interest for an item is computed as the cosine of the angle between their respective concept vectors, once they have been extended with the semantic expansion technique discussed in the previous subsection.

Analogously, we have defined the notion of semantic context as the set of ontological concepts present in the annotations of the items recently browsed or evaluated by the user. The context is described by a vector representation, so it can be easily combined with the basic personalised model. In particular, we have studied the linear combination of both models scores, but other alternatives would be feasible.

The vector representation not only allows the combination of a user profile with the semantic context, but also merging multiple profiles in order to build a single profile which somehow takes into account the preferences of a group of users. This group profile can then be used by the basic recommendation model. The development of an effective strategy to combine the profiles of a group has been investigated in this thesis, and we have shown the feasibility of applying certain techniques drawn from social choice theory (Masthoff, 2004).

The contribution of our work with regards to the flexibility issue in recommender systems can be summarised as:

Building an ontology-based personalised recommendation model which allows the incorporation of semantic context, and can be adapted to hold the preferences of one or more users.

The main benefit of the proposed personalised recommendation model is its flexibility for being adapted to:

- *Contextualised recommendations.* Adding semantic context into the personalised recommendation process allows casting the user's preferences into the scope of the ongoing user activity. Usually, not all the preferences available in a user profile are related to the current search or short-term user priorities, and only those preferences which are within the present scope should be taken into consideration.
- *Group-oriented recommendations.* The proposed group modelling strategies, apart from allowing a straightforward execution, and going beyond the mere aggregation of preferences (by using techniques based on social choice theory), is open to its generic application in any domain, provided of course that the knowledge representation exposed in this work is used.

9.1.3 Semantic hybrid recommendations

A content-based recommender system suggests items to a user taking into account only the preferences defined in his profile. Such recommendations, while accurate, can be counterproductive in certain circumstances. In general, content-based strategies entail a risk of **over-specialisation** of the recommended items, which share a limited set of content features. A **lack of diversity and novelty**, undesirable and negatively valued by the users, may result from this.

These problems are addressed by collaborative filtering strategies, which recommend items to the user based on evaluations of other people with whom he shares certain preferences. (Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994; Hill, Stead, Rosenstein, & Furnas, 1995; Shardanand & Maes, 1995; Konstan, Miller, Maltz, Herlocker, Gordon, & Riedl, 1997; Pennock, Horvitz, Lawrence, & Giles, 2000). Thus, the user receives suggestions of items whose contents are not directly related to his profile and former choices, but to profiles and choices of “similar” users. The effectiveness of these strategies is supported by their success in actual commercial applications, such as *Amazon.com* (Linden, Smith, & York, 2003), yet several limitations remain. One of such difficulties is recommending items to **users with unusual preferences** (known in the literature as “grey sheep”). To establish the similarity between users, various measures have been proposed (Adomavicius & Tuzhilin, 2005). However, in general, all of them are based on global comparisons of the profiles. In this thesis, we advocate for splitting the profiles according to significant groups of preferences shared among users, and establishing user similarities based on each of the obtained sub-profiles. Thus, coincidences of unusual preferences have further chances of being found when dealing with smaller profiles, focused on specific yet cohesive semantic areas of interest and taste.

Summing up, collaborative recommender systems have a:

Difficulty in recommending items to users with unusual preferences, or sharing interests only in specific semantic areas

The above observation calls for an underlying need in recommendation environments to distinguish different levels or layers within the user profiles. Depending on the current layer, only a specific subset of the user’s preferences should be considered to establish his similarities with other people when recommendations are to be made.

To meet that need, this thesis presents a strategy which builds upon the proposed ontology-based knowledge representation. By taking advantage of the semantic relationships between concepts, and of the (weighted) preferences of the users for such concepts, the strategy clusters the semantic space in terms of correlations between concepts of the user profiles. Thus, the created concept clusters

can be understood as sets of preferences shared by significant sets of users. By projecting these clusters onto the user profiles, the latter are divided into several segments. Based on such segments (or sub-profiles), users are compared at different levels, allowing more than one (weighted) relationship between any two users. The relations between users at the different semantic levels represent different latent communities of interest, and can be used to provide recommendations in more focused or specialised conceptual areas, even when the whole user profiles are globally fairly dissimilar.

By the above semantic multilayer communities of interest, an additional contribution of this work is:

Building hybrid models which combine user profiles collaboratively at various semantic levels, in response to different groups of shared preferences.

The hybrid recommendation models based on multiple semantic layers bring the following advantages:

- *Mitigation of content over-specialisation, and lack of content novelty and diversity effects.* By to the collaborative combination of user profiles, these problems of pure content-based approaches can be avoided. A user may receive novel and diverse recommendations that do not have to be strictly related to his preferences, but to preferences of other similar people.
- *Mitigation of the “grey sheep” effect.* Through the contextualisation of the recommendations into different semantic layers based on tastes and interests shared by users, we reinforce significant occurrences between unusual preferences when user profiles are compared.

9.1.4 Evaluation of the recommendation models

Unlike other disciplines, the evaluation of recommender systems is not simple. In the literature, several metrics which attempt to objectively estimate the accuracy of the recommendations have been defined (Herlocker, Konstan, Terveen, & Riedl, 2004). The main idea of these metrics is to average the difference between actual assessments provided by users, and predictions provided by the system, for a set of reference items. Although they are often used as a standard method for comparing recommendation models, in many cases, they seem insufficient because they do not contemplate more subjective, but important magnitudes, such as the novelty, diversity or coverage (of the item space) provided by the recommendations (Sarwar, Konstan, Borchers, Herlocker, Miller, & Riedl, 1998; Good, et al., 1999; Herlocker, Konstan, Borchers, & Riedl, 1999; Herlocker, Konstan, & Riedl, 2000; Sarwar, Karypis, Konstan, & Riedl, 2001; Schein, Popescul, & Ungar, 2001).

Using accuracy metrics in different experiments, the recommendation models proposed in this thesis were evaluated with both real users, and artificial datasets created from external sources and standard collections. Each of these independent (and we might say isolated) experiments provided positive results backing the feasibility and validity of the proposals. Nonetheless, we saw the need for carrying out additional, integrative experimentation in an environment which articulated the different models, which was not as controlled and closed as the isolated evaluations, and which considered subjective user assessments. In other words, we found it necessary to provide an:

Evaluation of the ontology-based knowledge representation and recommendation models with a prototype system

Thus, as the last part of the thesis, we implemented *News@hand*, a news recommender system in which all the proposed recommendation models were integrated, and where the textual contents of the news are annotated with concepts belonging to a set of ontologies that cover various general domains of interest.

The results obtained with the system confirm and extend the conclusions that were previously reached in the isolated experiments, providing additional findings. The personalised recommendations helped users find relevant items, and the semantic expansion of preferences eased the matching between user and item profiles, improving precision values for the top suggested items, and mitigating the cold-start and sparsity problems. The contextualisation of the personalisation mechanism speeded up the discovery of items related to current search goals, and was highly appreciated by the test subjects. Finally, evidence was shown that layered hybrid recommendations enhanced collaborative approaches when partial (interest-focused) comparisons of user profiles were computed, thus reducing the effect of the grey sheep problem.

This experimental work also brought the opportunity of getting feedback from users about the system functionalities and outputs. Among other issues, they raised the wish to have a semantic disambiguation step in the annotation process, as well as further support for the non-diversity problem, as similar items were often presented close to each other in the recommendation result pages. They further suggested improvements in the profile editor, such as the integration of a real-time preference recommender which would suggest similar concepts to the ones already introduced (synonyms, co-occurrences, etc.) when the user is manually editing his profile.

News@hand was useful not only to make joint evaluations of the recommendation strategies, but also to highlight the difficulties involved in the transition of the ontology-based models and strategies to a real application. While building the system, a number of research challenges emerged, for which additional innovative and original solutions had to be developed. Specifically, we needed to

implement a technique to populate (i.e., create instances in) the domain ontologies, an automatic mechanism to semantically annotate the articles, and a strategy to convert tags (keywords) to ontological concepts.

This final contribution of the thesis can be summarised as:

The implementation of a prototype system in which we have integrated and evaluated all the proposed recommendations models, and which provides a platform for the development and testing of future proposals addressing open research topics in personalisation and recommender systems.

The advantages brought by this recommender system have already been mentioned:

- *Obtaining more realistic empirical results.* *News@hand* enables more realistic experimental settings and results than those produced by the isolated evaluations of each of the studied models. Likewise, the system has facilitated the collection of subjective user evaluations which can be taken into account to improve the recommendation models.
- *Discovery, analysis and solutions for difficulties and problems in the actual implementation of a semantic recommender system.* The implementation of *News@hand* raised new challenges on its own, which had to be solved in this thesis, such as the population of ontologies, the semantic annotation of texts, and the semi-automatic generation of user profiles. While the proposed solutions leave room for the continuation of work, they contribute by themselves ideas of value for the scientific community.
- *Availability of a development and evaluation platform.* *News@hand* can be adapted to incorporate new personalisation and recommendation functionalities and models, thus providing a platform on which to evaluate future proposals.

9.2 Discussion and future work

In this thesis, we have presented several recommendation models that exploit the semantic description of user preferences and item contents to address common problems of current recommender systems. Though we have covered a considerable number of the most important problems, further relevant research topics which are not addressed here, but have a close relation to the ones addressed, are worth mentioning. Moreover, in addition to new lines of work, further improvements or alternatives to aspects of the presented proposals can be pointed out.

Unresolved limitations, possible courses of action to address them, and potential future research challenges are discussed in the following subsections.

9.2.1 Semantic resources

The effectiveness of semantic-based systems depends on the richness of the metadata representation in the knowledge bases, and the quality of the item annotations. In the case of personalisation and recommendation systems, the accuracy of the results is also influenced by the correctness and completeness of the description of user preferences in the profiles.

The design and construction of ontologies are outside the scope of the objectives of this thesis, and are subjects of extensive study in various disciplines of the Semantic Web area. Under the title *Ontological Engineering* (Gómez-Pérez, Fernández-López, & Corcho, 2003), different research lines are encompassed:

- Definition and development of methodologies (Uschold & Grüninger, 1996) and tools (Gennari, et al., 2003) to support the process of building ontologies.
- Implementation of strategies for the reuse of ontological knowledge (*Ontology Reuse*), by integrating various semantic sources (*Ontology Integration*) (Farquhar, Fikes, & Rice, 1996), or analysing the correlation between concepts (*Ontology Alignment* or *Ontology Matching*) (Euzenat & Shvaiko, 2007).
- (Semi)automatic generation of ontologies (*Ontology Learning*) (Maedche & Staab, 2001; Shamsfard & Barforoush, 2003) through the extraction of concepts and relationships from a text corpus or other types of databases.

All the works presented in this thesis started from a set of already built domain ontologies or other forms of semantic structures. For example, *News@hand* used adaptations of the IPTC ontology. Many of such ontologies contained the definition of class hierarchies, properties and relations, but did not contain any instance. For this reason, we needed to develop an automatic **ontology population** mechanism, i.e., a procedure whereby instances of a base corpus are identified and associated to ontological classes (Brewster, Ciravegna, & Wilks, 2001). The proposed method presents the idea of exploiting the Wikipedia categories. Given a term to instantiate, which for example is extracted from the text of a news item in the case of *News@hand*, we search for it in Wikipedia. If the term exists in that database, we obtain a web page containing a description and a number of pre-established categories of the concept. By linking these categories to ontological classes, a heuristic determines the most suitable class for the instance to create. The heuristic gave good results, but could be improved e.g. by also processing the descriptive texts of the concepts, in order to solve ambiguities between classes (Cucerzan, 2007), or extract further semantic relations between instances (Ruiz-Casado, Alfonseca, & Castells, 2006).

Once we have populated the domain ontologies, we can proceed with the **content annotation** (Uren, et al., 2006). The annotation task consists in identifying ontological concepts (classes and instances) within the item contents. It is a difficult

problem to solve, which is being widely studied in research areas such as Information Retrieval, Natural Language Processing, and the Semantic Web. In this thesis, the annotation problem has been addressed by an adaptation of the *Wraetlic* linguistic processing tools (Alfonseca, Moreno-Sandoval, Guirao, & Ruiz-Casado, 2006). These tools process texts at morphologic and syntactical levels, and extract all their nouns, including proper and compound nouns. Then, we apply a new heuristic that identifies classes and instances related to the extracted nouns by computing morphological similarities. In this approach, there is no semantic-level analysis, and because of that there were ambiguity situations in which we chose the wrong meaning of the concepts. Similarly to the ontology population process, in this case, a semantic disambiguation of the identified concepts would improve the accuracy of the annotations and hence the recommendations.

Apart from the ontological knowledge bases and semantic annotations, another resource exploited by the recommenders is related to the *user profiles*. The profiles used in this work were manually created by users. To facilitate this, in the experiments, we provided the evaluators a set of tools to create and edit their preferences. For example, *News@hand* includes an ontology browser that allows viewing the class hierarchies, expanding/collapsing taxonomic relations, list instances of each class, and search for concepts with the help of mechanisms that “auto-complete” query terms as they are being written. Users highly appreciated these facilities, but suggested several improvements, including the incorporation of a preference recommendation component. When a profile is being created, the system could suggest new preferences related to those already introduced. In this case, the relations might be automatically proposed based on semantic similarities or correlations between concepts co-occurring in a single item or in all users’ profiles (Jäschke, Marinho, Hotho, Schmidt-Thieme, & Stumme, 2007; Sigurbjörnsson & Van Zwol, 2008).

On the other hand, in addition to the facilities at the graphical user interface level, we have proposed in this thesis a strategy that automatically transforms social tags into ontological concepts. Thus, rather than having to search for existing concepts, the user can directly introduce terms that describe his tastes and interests, and the system takes care of seeking them in the ontologies. This kind of strategy, which is not simple as it has to consider misspellings, acronyms, synonyms, etc., represents a research topic of particular interest for social applications, and is becoming increasingly popular nowadays (Specia & Motta, 2007; Van Damme, Hepp, & Siorpaes, 2007; Hess, Maass, & Dierick, 2008; Van der Sluijs & J, 2008).

Finally, another possible approach is to relieve (in part or in whole) the user from declaring his preferences, and have the system infer or learn them by analysing the user’s actions in the system. Notwithstanding this issue being not addressed in this thesis, other researchers have already begun to work on the problem using *News@hand* (Picault & Ribière, 2008).

9.2.2 Recommendation models

The performed evaluations showed that *contextualised recommendation* improves the effectiveness of the basic personalised content retrieval model, focusing the current interests of the user. The notion of context considered here is defined as the set of all (weighted) semantic concepts belonging to the annotations of the most recently browsed or rated items. This description, though useful in practice, could be further enriched with semantic information from other external sources (Chirita, Firan, & Nejdl, 2006), such as upcoming events scheduled in an electronic agenda, recently received messages in an email client, or favourite websites stored in a web browser. In the proposal, the weights assigned to the context concepts decay over time, assuming the hypothesis that the user's focus of interest gradually evolves, drifts, or shifts to new targets. However, other approaches are plausible (White, Ruthven, Jose, & Van Rijsbergen, 2005), and would lead to new strategies for updating the semantic context. Once the mechanisms that create and update the context have been defined, they have to be integrated with the personal recommendation model. As a first approximation, we studied the linear combination of the models. However, again, other alternatives could be considered (Vallet, Castells, Fernández, Mylonas, & Avrithis, 2007).

Regarding the *group-oriented recommendation*, we realised the need for more exhaustive experimentation. In fact, the group-modelling strategies proposed in this thesis are the only ones that were not evaluated in *News@hand*, despite being integrated into the system. As future enhancements of the previous techniques, we propose the inclusion of new variables in the profile merging strategies, which might be related to different context sources, such as the current location, date and time, the users' age and gender, etc. (Ardissono, Goy, Petrone, Segnan, & Torasso, 2003). Thus, for example, it is not the same to recommend an afternoon TV show to a family with children, as to suggest a film to a couple after a romantic dinner.

The *multilayer hybrid recommendation* can be considered as the most significant contribution of the thesis, and hence it has been tested more thoroughly, both with real users in different scenarios, and with artificially created datasets. However, an aspect that has not been discussed so far is its computational cost. Although, analogously to collaborative filtering strategies, the user and item similarities can be recalculated with an off-line process, without affecting the system performance, the efficiency of the developed algorithms could be improved considerably. Specifically, the clustering technique which groups shared semantic preferences to create multilayered communities of interest makes use of hierarchical clustering strategies creating conceptual clusters at K levels, where K is the total number of concepts (Duda, Hart, & Stork, 2001). For this reason, we plan to apply more scalable clustering strategies based on SVD and LSI (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990; Landauer, Foltz, & Laham, 1998), or co-

clustering (George & Merugu, 2005). In addition to the scalability problem, other issue that could be studied is the exploration of new models for the collaborative comparison and combination of semantic preferences and contents. Recently, very similar approaches to this work have been emerged, sharing the proposed ontological knowledge representation (even including the idea of semantic expansion), but advocating alternative recommendation models. For instance, in (Mobasher, Jin, & Zhou, 2004), the authors present a collaborative filtering strategy in which the similarity between two items (see Section 2.3.2) is defined by means of a measure that takes into account the common concepts in both of their representations. In (Gauch, Chaffee, & Pretschner, 2003), on the contrary, the item similarity is based on distances between concepts within the ontological structures.

9.2.3 Evaluation framework

The construction of *News@hand* had a twofold motivation. On the one hand, it would be used as a platform for evaluating the recommendation models. The system would allow carrying out experiments less restricted than those conducted earlier. Users would interact with the models for longer periods of time, providing much information with which to measure more accurately the effectiveness of the proposals. Moreover, its implementation would be useful to highlight the problems and difficulties arisen from creating a system based on semantic technologies. In fact, those were the aspects that originated the above mentioned techniques to automatically populate ontologies and transform terms into ontological concepts.

The experience and empirical results obtained in the experiments, as well as the comments received from the evaluators will be used to correct errors found in the system, and to make changes and improvements in the followed evaluation methodology. Once *News@hand* has all its features ready, it will be made public on the Web. At this point, we will hopefully perform new ***larger-scale experiments***, with a significantly large number of users, and during periods of several months (Middleton, Shadbolt, & Roure, 2004).

Of course, future evaluations will not be limited to the proposals proposed in this work. We envision additional research to address other outstanding issues in the area of recommender systems. Specifically, we notice interesting the study of ***query-driven recommendation models*** (Adomavicius, Tuzhilin, & Zheng, 2005), and techniques that facilitate the ***understanding of recommendations*** (Tintarev & Masthoff, 2007). For the first case, we could design recommendation definition languages which would be extensions of ontology query languages (e.g., RDQL), or could combine recommendation models with semantic search mechanisms (Castells, Fernández, & Vallet, 2007). On the other hand, for the second case, we might evaluate techniques that would infer and explain the semantic concepts and relationships that shape the recommendations made to the user.

Appendix A

Acronyms

The following are the acronyms used throughout this document. For each of them, a brief description of its meaning is provided. In most cases, the presented descriptions have been obtained from Wikipedia (www.wikipedia.org).

AI	<i>Artificial Intelligence</i> : the intelligence of machines, and the branch of computer science that aims to create it.
API	<i>Application Programming Interface</i> : a set of declarations of the functions (or procedures) that an operating system, library or service provides to support requests made by computer programs.
CERN	<i>European Organization for Nuclear Research (originally stood, in French, for Conseil Européen pour la Recherche Nucléaire, i.e., the European Council for Nuclear Research)</i> : the world's largest particle physics laboratory, situated in the northwest suburbs of Geneva on the Franco-Swiss border, established in 1954.
CF	<i>Collaborative Filtering</i> : a method of making automatic predictions about the interests of a user by collecting rating information from many users.
CSA	<i>Constrained Spreading Activation</i> : a general processing technique of a network data structure, consisting of nodes interconnected by links. Spreading activation techniques are iterative in nature. Each iteration consists of one or more pulses and a termination check, which enable some form of control over the activation of the nodes in the network.
CoI	<i>Communities of Interest</i> : a collaborative group of users that exchange information in pursuit of their shared goals, interests, missions, or business processes, and therefore have a shared vocabulary for the information they exchange.

DARPA	<i>Defence Advanced Research Projects Agency</i> : an agency of the United States Department of Defence responsible for the development of new technology for use by the military.
FOAF	<i>Friend-Of-A-Friend</i> : a machine-readable ontology describing persons, their activities and relations to other people and objects. FOAF allows groups of people to describe social networks without the need for a centralised database.
HTML	<i>HyperText Markup Language</i> : the predominant markup language for web pages.
IPTC	<i>International Press Telecommunications Council</i> : a consortium of the world's major news agencies and news industry vendors. It develops and maintains technical standards for improved news exchange.
IR	<i>Information Retrieval</i> : the science of searching for information in documents, searching for documents themselves, searching for metadata that describe documents, or searching within databases.
IRC	<i>Internet Relay Chat</i> : a form of real-time Internet chat or synchronous conferencing. It is mainly designed for group communication in discussion forums called channels, but also allows one-to-one communication via private messages.
KB	<i>Knowledge Base</i> : a special kind of database for knowledge management, which provides the means for the computerised collection, organisation and retrieval of knowledge.
k-NN	<i>k-Nearest Neighbours</i> : one of the simplest Machine Learning algorithms where an object is classified by a majority vote of its neighbours; it is assigned to the class most common amongst its k nearest neighbours (where k is a positive integer, typically small).
LSA	<i>Latent Semantic Analysis</i> : a technique in natural language processing of analysing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms.
MAE	<i>Mean Absolute Error</i> : an average of the absolute errors $e_i = f_i - y_i$, where f_i is a prediction and y_i the true value.
ML	<i>Machine Learning</i> : a broad subfield of Artificial Intelligence which is concerned with the design and development of algorithms and techniques that allow computers to “learn”. Its major focus is to extract information from data automatically, by computational and statistical methods.

MSE	<i>Mean Squared Error</i> : one of many ways to quantify the amount by which an estimator differs from the true value of the quantity being estimated. MSE measures the average of the square of the “error” (the amount by which the estimator differs from the quantity to be estimated).
NLP	<i>Natural Language Processing</i> : a subfield of Artificial Intelligence and Computational Linguistics, which studies the problems of automated generation and understanding of natural human languages.
NN	<i>Nearest Neighbours</i> : see k-NN.
OOP	<i>Object-Oriented Programming</i> : a programming paradigm that uses “objects” and their interactions to design applications and computer programs, including features such as encapsulation, modularity, polymorphism and inheritance.
OWL	<i>Web Ontology Language</i> : a markup language for publishing and sharing data using ontologies on the World Wide Web.
QA	<i>Question Answering</i> : a type of information retrieval, in which given a collection of documents a system should be able to retrieve answers to questions posed in natural language.
RDF	<i>Resource Description Framework</i> : a World Wide Web Consortium specification for a metadata model and component in the Semantic Web proposal.
RDFS	<i>RDF Schema</i> : an extensible knowledge representation language, providing basic elements for the description of ontologies intended to structure Resource Description Framework (RDF) resources.
RDQL	<i>RDF Query Language</i> : a computer language able to retrieve and manipulate data stored in Resource Description Framework (RDF) format.
RSS	<i>Really Simple Syndication (a.k.a. RDF Site Summary, Rich Site Summary)</i> : a family of standard web formats used to publish frequently updated content such as blog entries, news headlines, and podcasts. An RSS document (which is called a “feed”) contains either a summary of content from an associated website or the full text.
SNA	<i>Social Network Analysis</i> : the mapping and measuring of relationships and flows between people, groups, organisations, computers, websites and other information/knowledge processing entities.

- SPARQL** *SPARQL Protocol and RDF Query Language (recursive acronym)*: a Resource Description Framework (RDF) query language and data access protocol for the Semantic Web. On 15th January 2008, SPARQL became an official W3C Recommendation.
- SQL** *Structured Query Language*: a database computer language designed for the retrieval and management of data in relational database management systems, database schema creation and modification, and database object access control management.
- SVD** *Singular Value Decomposition*: an important factorisation of a rectangular real or complex matrix, with several applications in signal processing and statistics. Applications which employ the SVD include computing the pseudo-inverse, least squares fitting of data, matrix approximation, and determining the rank, range and null space of a matrix.
- TF-IDF** *Term Frequency-Inverse Document Frequency*: a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.
- URI** *Uniform Resource Identifier*: a compact string of characters used to identify or name a resource on the Internet. A URI *may* be classified as a locator (URL) or a name (URN), or both. The URN defines an item's identity, while the URL provides a method for finding it. For example, ISBN 0486275574 (urn:isbn:0-486-27557-4) cites unambiguously a specific edition of Shakespeare's play "Romeo and Juliet", whilst a URL for this book in the Web could be <http://www.example.org/RomeoAndJuliet.pdf>.
- XML** *eXtensible Markup Language*: a general-purpose specification for creating custom markup languages.
- WWW** *World Wide Web*: a system of interlinked hypertext documents accessed via the Internet.
- W3C** *World Wide Web Consortium*: an international consortium where member organisations maintain full-time staff for the purpose of working together in the development of standards for the World Wide Web.

Appendix B

News@hand API

Section 7.3 provided a general view of *News@hand* architecture. This appendix summarises the Application Programming Interface (API) of the prototype system, briefly describing its main software components.

Section B.1 explains the database manager, which is able to handle multiple database connections in a flexible, easy-to-use way. Section B.2 describes the ontology plugin, a component that controls the access to ontologies stored in local/remote text files and databases. Section B.3 explains the user profile manager, which is composed of several modules that manage ontological user profiles. Sections B.4 and B.5 present the components that encapsulate the personalised and collaborative recommenders proposed in this thesis. Section B.6 introduces a preference learning module incorporated into the system, in order to infer long-term user preferences from recent user actions on the evaluation platform. Finally, Section B.7 summarises the log information generated and exploited by the system.

B.1 Database manager

The management of relational databases has been implemented in a three-layer JDBC³¹ framework (Figure B.1) in order to encapsulate the basic database access methods, and offer an easy-to-use upper-level API for controlling specific database components, such as MySQL or Jena MySQL managers.

The functionalities of these layers are the following:

- **Database connection.** A bottom layer that is formed by a set of Java classes encapsulating the basic operations provided by *java.sql* library: creation, opening and closing of JDBC connections, execution of SQL select, insert, delete and update operations, etc. At this level, specific information about the utilised database driver is needed.
- **MySQL database connection.** An upper layer built on top of the previous one that provides JDBC MySQL protocol and driver information, in order to connect to MySQL databases. At this level, only information about the database name, and the user's name and password has to be provided.
- **Jena MySQL database connection.** A top layer that generates MySQL connections to manage Jena ontology models. In addition to database parameters, this layer gathers additional information about the ontology model stored in the database.

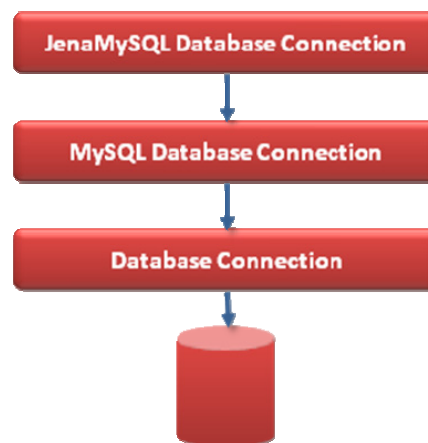


Figure B.1 The three-layer JDBC connection manager architecture.

The Java packages defined for the three previous layers are *es.uam.eps.nets.database*, *es.uam.eps.nets.database.mysql*, and *es.uam.eps.nets.database.jena.mysql*. The following tables show a brief description of the main classes existing in the database packages.

³¹ Java Database Connectivity (JDBC), <http://java.sun.com/javase/technologies/database/>

Package: es.uam.eps.nets.database	
Class	Description
<i>DatabaseConnectionBean</i>	Manages a generic JDBC connection with multiple readers.
<i>DatabaseConnectionPool</i>	Manages a pool of generic JDBC connections, each of them with multiple connectors.
<i>DatabaseConnector</i>	Manages a generic JDBC connection.

Package: es.uam.eps.nets.database.mysql	
Class	Description
<i>MySQLDatabaseConnectionBean</i>	Manages a MySQL connection with multiple readers.
<i>MySQLDatabaseConnectionPool</i>	Manages a pool of MySQL connections, each of them with multiple readers.
<i>MySQLDatabaseConnector</i>	Manages a MySQL JDBC connection.

Package: es.uam.eps.nets.database.jena.mysql	
Class	Description
<i>JenaMySQLDatabaseConnectionBean</i>	Manages a Jena MySQL connection with multiple readers.
<i>JenaMySQLDatabaseConnectionPool</i>	Manages a pool of Jena MySQL connections, each of them with multiple readers.

Table B.1 Main classes of the database manager component.

B.2 Ontology plugin

A general ontology management component has been implemented (Figure B.2). This component has a main class named *OntologyPlugin*, which defines an abstract framework with those functionalities that a specific ontology manager has to provide.

A more specific ontology plugin, *JenaOntologyPlugin*, which uses Jena library has been included in the component. This plugin temporally loads in memory an ontology model described in RDF or OWL languages.

To determine the logical device where the ontology model has to be permanently stored, a specific *JenaOntologyPlugin* subclass has to be implemented. In the current version of *News@hand* system, two different subclasses are included: *JenaMySQLOntologyPlugin*, which works with ontologies stored in MySQL databases, and *JenaURLOntologyPlugin*, which works with ontologies stored in (local or remote) text files.

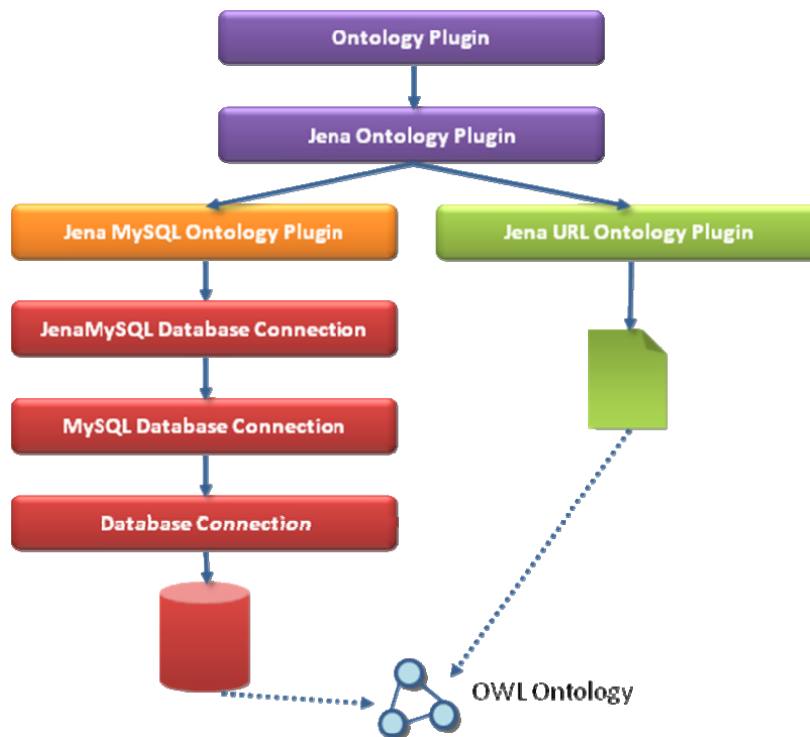


Figure B.2 The ontology manager architecture.

Furthermore, and more specifically, the ontology management component is composed by four different sets of Java classes:

- **Ontology entity classes.** A set of Java classes that wrap the information concerning the basic elements of an ontology: classes, instances, properties, literals, labels, triples (statements).

Package: es.uam.eps.nets.ontology	
Class	Description
<i>OntologyEntity</i>	Stores the information associated to an ontology entity (class, property, instance, literal).
<i>OntologyEntityLabel</i>	Stores the value and the language of an ontology label.
<i>OntologyEntityList</i>	Contains a list of ontology entities, not allowing duplicates.
<i>OntologyProperty</i>	Stores the information associated to an ontology property.
<i>OntologyStatement</i>	Stores the information associated to an ontology statement.
<i>URIEntity</i>	Stores the information associated to an URI.

Table B.2 Main ontology entity classes.

- **Ontology plugin classes.** A set of Java classes that provide access (read and write functionalities) to ontology models. The ontology entities are managed by the ontology plugin classes, which are defined in the package *es.uam.eps.nets.ontology.plugin*. A main abstract class *OntologyPlugin* has been extended by the class *JenaOntologyPlugin* to access to ontology models using Jena (packages *es.uam.eps.nets.ontology.plugin.jena*, *es.uam.eps.nets.ontology.plugin.jena.url* and *es.uam.eps.nets.ontology.plugin.jena.mysql*). A multi-ontology management class has also implemented to store several ontology plugins.

Package: <i>es.uam.eps.nets.ontology.plugin</i>	
Class	Description
<i>OntologyPlugin</i>	Interface that defines the basic functionalities of generic ontology management plugins.
<i>MultiOntologyPluginManager</i>	Manages a set of ontology plugins.

Package: <i>es.uam.eps.nets.ontology.plugin.jena</i>	
Class	Description
<i>JenaOntologyPlugin</i>	Implements <i>OntologyPlugin</i> with the Jena framework.

Package: <i>es.uam.eps.nets.ontology.plugin.jena.url</i>	
Class	Description
<i>JenaURLOntologyPlugin</i>	Extends <i>JenaOntologyPlugin</i> for ontologies read from a specified URL.

Package: <i>es.uam.eps.nets.ontology.plugin.jena.database.mysql</i>	
Class	Description
<i>JenaMySQLDatabaseOntologyPlugin</i>	Extends <i>JenaOntologyPlugin</i> for ontologies read from a specified MySQL database.

Table B.3 Main ontology plugin classes.

- **Ontology plugin repository classes.** A set of Java classes that manage configuration and storage information of a number of ontology plugins to be loaded. The information of the ontology plugins is stored in the so-called ontology plugin repositories. Basically, these repositories are XML files that contain the configuration and storage information of the plugins, i.e., file locations, database user names and passwords, types of ontology models (RDF, OWL), ontology access frameworks (Jena, Sesame), etc. The package that contains the Java classes related to ontology plugin repositories is *es.uam.eps.nets.ontology.plugin.repository*.

Package: es.uam.eps.nets.ontology.plugin.repository	
Class	Description
<i>OntologyPluginRepository</i>	Manages a repository with the information of a given ontology plugin.
<i>OntologyPluginRepositoryFileManager</i>	Manages XML files that store ontology plugin repository information.

Table B.4 Main ontology plugin repository classes.

- **Ontology annotation classes.** A set of upper level Java classes that associate annotation meta-information (labels, weights, etc.) to specific basic ontology entities. These classes will be used by the personalised content retrieval and the recommendation components, and could also be included in other components such as ontology search and annotation ones. The high-level classes, which allow the incorporation of additional information to ontology entities in the form of weighted annotations, have been defined in the package *es.uam.eps.nets.ontology.annotation*.

Package: es.uam.eps.nets.ontology.annotation	
Class	Description
<i>Annotation</i>	Stores and manages an annotation associated to a given ontology entity. Among other things, it contains a weighted ontology entity.
<i>AnnotationList</i>	Stores and manages a list of annotations.
<i>WeightedOntologyEntity</i>	Stores and manages a weight assigned to a given ontology entity.
<i>WeightedOntologyEntityList</i>	Stores and manages a list of weighted entities.

Table B.5 Main ontology annotation classes.

B.3 User profile manager

The main functionality of this component is based on the interaction with the client, and the handling of the user profile. It is the responsible of storing user preferences and personal information, and allows viewing, editing and deleting those data. Two kinds of user profile are stored by this component: the persistent user profiles, which contain steady or long-term user preferences, and the temporary user profiles, which contain the transient user preferences managed during a specific session or short time period.

The storage of persistent user preferences is carried out by augmenting the database and ontology management frameworks. As shown in Figure B.3, two new layers have been added to those explained in Sections B.1 and B.2. The first layer,

named *UserProfile*, consists of a set of Java classes that simply store the information of a user profile. To load those classes from an ontology, the second layer *UserProfileOntologyManager* directly accesses the database or text file that stores the RDF/OWL user profile model.

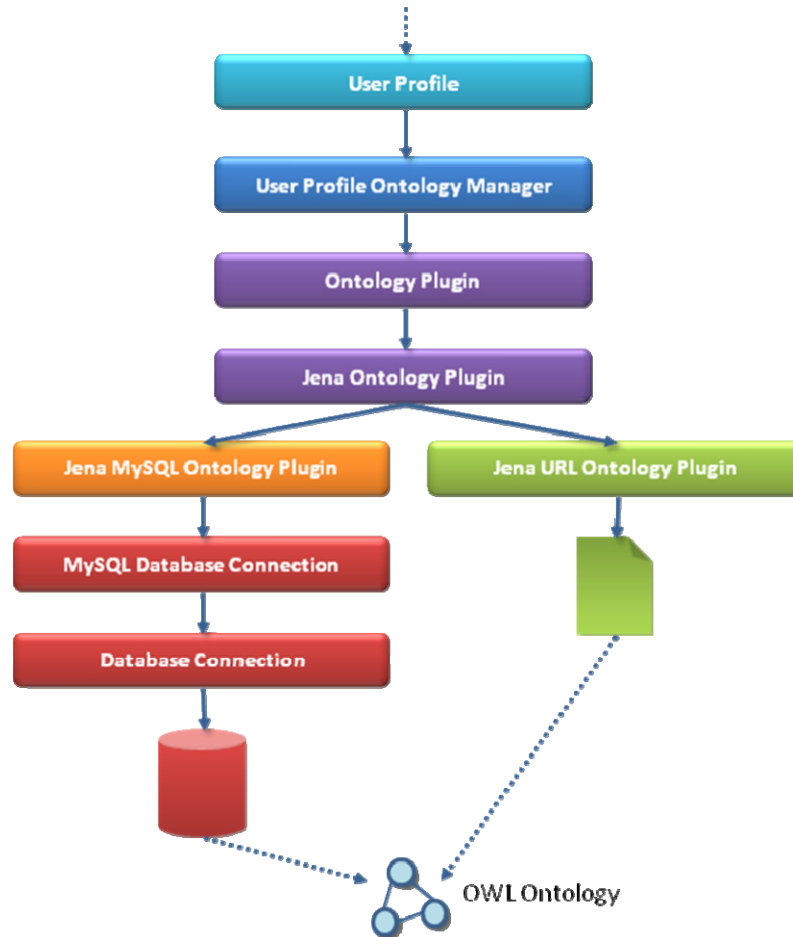


Figure B.3 The user profile management architecture.

In the following, we summarise the Java classes implemented to manage user profile information stored temporally in memory, and permanently in ontology repositories (databases or text files).

B.3.1 User profile memory storage

The package that contains all the classes destined to temporally store in memory the information of user profiles is *es.uam.eps.nets.personalisation.profile*. The following table briefly describes its main classes, which correspond to some of the elements defined for the user profile ontology of *News@band*.

Package: es.uam.eps.nets.personalisation.profile	
Class	Description
<i>UserProfile</i>	Stores all the information associated to the user.
<i>Login</i>	Stores the user's login.

Package: es.uam.eps.nets.personalisation.profile.demographic	
Class	Description
<i>DemographicProfile</i>	Stores the user's demographic profile. Based on (Pazzani, 1999).
<i>Address</i>	Stores the user's address.
<i>Birth</i>	Stores the user's birth.
<i>Contact</i>	Stores the user's contact information.
<i>Education</i>	Stores the user's education information.
<i>Job</i>	Stores the user's job information.

Package: es.uam.eps.nets.personalisation.profile.semantic	
Class	Description
<i>SemanticPreference</i>	Stores a semantic preference of the user.
<i>SemanticPreferences</i>	Stores all the semantic preferences of the user.

Package: es.uam.eps.nets.personalisation.profile.collaborative	
Class	Description
<i>Rating</i>	Stores a rating of the user.
<i>Ratings</i>	Stores all the ratings of the user.
<i>RatingCriterion</i>	Stores the information associated to a rating criterion, which is incorporated to each rating element.

Package: es.uam.eps.nets.personalisation.profile.social	
Class	Description
<i>SocialProfile</i>	Stores the user's social profile.
<i>PersonalCategory</i>	Stores a personal category of the user, which is used to categorise the social links.
<i>PersonalCategories</i>	Stores all the personal categories of the user.
<i>SocialLink</i>	Stores a social link (i.e., contact information of a known person) of the user.
<i>SocialLinks</i>	Stores all the social links of the user.

Table B.6 Main classes of the user profile memory storage component.

B.3.2 User profile ontology handling

In *News@hand*, user profiles are permanently stored in OWL databases or text files. As explained before, they are loaded in the wrapping classes of the package *es.uam.eps.nets.personalisation.profile*. To load and save those classes, a set of ontology handlers have been defined in the package *es.uam.eps.nets.personalisation.profile.ontology*, and its sub-packages *demographic*, *semantic*, *collaborative* and *social*, as shown in the following tables.

Package: <i>es.uam.eps.nets.personalisation.profile.ontology</i>	
Class	Description
<i>UserProfileOntologyHandler</i>	Handles user profiles stored in an ontology.

Package: <i>es.uam.eps.nets.personalisation.profile.ontology.demographic</i>	
Class	Description
<i>DemographicProfileOntologyHandler</i>	Handles demographic user profiles stored in an ontology.
<i>AddressOntologyHandler</i>	Handles addresses stored in an ontology.
<i>BirthOntologyHandler</i>	Handles births stored in an ontology.
<i>ContactOntologyHandler</i>	Handles contact information instances stored in an ontology.
<i>EducationOntologyHandler</i>	Handles education information instances stored in an ontology.
<i>JobOntologyHandler</i>	Handles job information instances stored in an ontology.

Package: <i>es.uam.eps.nets.personalisation.profile.ontology.semantic</i>	
Class	Description
<i>SemanticPreferenceOntologyHandler</i>	Handles semantic interest lists stored in an ontology.

Package: <i>es.uam.eps.nets.personalisation.profile.ontology.collaborative</i>	
Class	Description
<i>RatingOntologyHandler</i>	Handles rating lists stored in an ontology.

Package: <i>es.uam.eps.nets.personalisation.profile.ontology.social</i>	
Class	Description
<i>SocialProfileOntologyHandler</i>	Handles social user profiles stored in an ontology.
<i>PersonalCategoryOntologyHandler</i>	Handles personal category lists stored in an ontology.
<i>SocialLinkOntologyHandler</i>	Handles social link lists stored in an ontology.

Table B.7 Main classes of the user profile ontology handling component.

B.3.3 User profile management

For each user, the profile manager has to handle two instances of the user profile:

- a persistent (long-term) user profile, which contains stable user preferences that evolve relatively slowly over time; and
- a transient (temporary) user profile, which consists in a temporary alteration or update of this permanent profile (e.g., by re-weighting of concepts), based on short-term usage history (usage context).

The persistent user profile is the only instance which is physically stored; the transient user profile is the only one kept in memory during the duration of one session. Two main classes have been defined to handle both sources of user information, as shown in the table below.

Package: es.uam.eps.nets.personalisation.profile.management	
Class	Description
<i>UserProfileManager</i>	There is one user profile manager instance per user. The user profile manager takes care of two user profiles - a persistent user profile which contains the long-term user preferences - a transient user profile which is only instantiated during one session.
<i>UserProfileEditor</i>	Interface presented to the user so that he can view/modify/delete his user profile.

Table B.8 Main classes of the user profile management component.

B.4 Personalised recommenders

B.4.1 Semantic content-based recommendation

The personalised content retrieval model described in Section 4.2 that evaluates how interesting is an annotated item for a user according to his semantic preferences, has been implemented in a class named *ContentBasedRecommender*. The comparison between user and item profiles is delegated to the class *VectorMatcher*, which computes cosine-based and other vector similarity measures.

Package: es.uam.eps.nets.recommendation.cb	
Class	Description
<i>ContentBasedRecommender</i>	Encapsulates the computation of the personal relevance measure, which takes into account the user's preferences, and their semantic expansion and contextualisation to provide enhanced personalised recommendations.
<i>ContentBasedRecommenderEvaluator</i>	Evaluates a semantic content-based recommender.
<i>VectorMatcher</i>	Matches the concepts of the semantic user preferences and item annotations.

Table B.9 Main classes of the semantic content-based recommendation component.

In addition to the above components for personalised content recommendation, a set of classes that encapsulate the functionalities of semantic preference expansion and contextualisation have been included in *es.uam.eps.nets.personalisation.preference* package.

B.4.2 Semantic context-aware recommendation

In the case of the semantic context-aware recommendations, a context monitor successively receives “selected item” and “executed query” notifications in the form of lists of weighted concepts. The received weighted concepts are added into the current semantic context. Without considering the last context update, but taking into account the time (turn) in which the rest of the concepts were included in the context, the weights of the context concepts are progressively updated following the formulas given in Section 4.3.

Package: es.uam.eps.nets.personalisation.preference.context	
Class	Description
<i>ContextMonitor</i>	Monitors and dynamically builds a semantic context through implicit user feedback.
<i>PreferenceContextualiser</i>	Filters the output of a semantic matcher with the current semantic context information.

Table B.10 Main classes of the semantic contextualisation component.

B.4.3 Semantic preference expansion

The personalised relevance measure computation, with or without the activation of the context-aware component, can be enhanced including expanded preferences (see Section 4.1). The implemented preference expander takes as input a set of weighted concepts (preferences), and expands these concepts through semantic properties of a given ontology, in order to obtain new weighted concepts related to the former. Additionally to other parameters, such as the minimum weight threshold, and the maximum distance in which stop the expansion (see Table 4.1), the preference expander allows to declare which ontology properties and resources can be used to build a concept neighbourhood at any time.

Package: es.uam.eps.nets.personalisation.preference.expander	
Class	Description
<i>PreferenceExpander</i>	Expands a set of preferences (weighted resources) through the relations (weighted properties) of a given set of ontology plugins.
<i>PreferenceNeighbourhood</i>	Stores the resources and the properties associated to an “ontology preference neighbourhood”.

Table B.11 Main classes of the semantic preference expansion component.

B.5 Collaborative recommenders

B.5.1 Collaborative filtering recommendation

Two well-known user-based and item-based collaborative filtering approaches presented in Section 2.3 have been included in *News@band*. The package *es.uam.eps.nets.recommendation.cf* contains a set of classes that wrap the implementation of the above approaches by the Taste³² Java library.

The collaborative filtering component offers an API similar to that presented in the content-based approach (Section B.4). For each recommender, an evaluator has been developed.

Package: <i>es.uam.eps.nets.recommendation.cf</i>	
Class	Description
<i>CollaborativeFilteringRecommender</i>	Abstract class that declares the methods to be implemented by any type of collaborative filtering recommender (in the form of subclasses).
<i>ItemBasedCollaborativeFilteringRecommender</i>	Implements an item-based collaborative filtering recommender.
<i>ItemBasedCollaborativeFilteringRecommenderEvaluator</i>	Evaluates an item-based collaborative filtering recommender.
<i>UserBasedCollaborativeFilteringRecommender</i>	Implements a user-based collaborative filtering recommender.
<i>UserBasedCollaborativeFilteringRecommenderEvaluator</i>	Evaluates a user-based collaborative filtering recommender.

Table B.12 Main classes of the collaborative filtering component.

B.5.2 Semantic multilayer hybrid recommendation

In addition to the content-based and collaborative filtering approaches, the hybrid recommendation strategies UP, UP- q , NUP and NUP- q , based on semantic multilayer communities of interest, and explained in Chapter 5, have been incorporated into the system.

The package that contains the implementation of the above techniques is *es.uam.eps.nets.recommendation.hybrid.multilayer*. As done in other recommender packages, this implementation provides an evaluator of the hybrid recommendation models.

³² <http://taste.sourceforge.net/>

Package: es.uam.eps.nets.recommendation.hybrid.multilayer	
Class	Description
<i>MultilayerHybridRecommender</i>	Offers semantic multilayer hybrid (content-based collaborative) recommendations.
<i>MultilayerHybridRecommenderEvaluator</i>	Evaluates the hybrid (semantic content-based collaborative) recommendations. The recommendation models implemented are: <ul style="list-style-type: none"> • UP (based on a user profile) • UP-q (based on a user profile, considering a cluster q) • NUP (no user profile) • NUP-q (no user profile, considering a cluster q)

Table B.13 Main classes of the semantic multilayer hybrid recommendation component.

To build the concept and user clusters exploited by our semantic multilayered hybrid recommendation approaches, the clustering algorithms provided by the Weka³³ data mining framework have been encapsulated by a class named *WekaClusterer*, which is located in the package *es.uam.eps.nets.clustering*. This class provides several general clustering mechanisms (K-MEANS, X-MEANS, EM, COBWEB, etc.), each of them with different execution parameters.

Package: es.uam.eps.nets.clustering	
Class	Description
<i>WekaClusterer</i>	Wraps the clustering algorithms implemented in the Weka data mining framework.

Table B.14 Main classes of the clustering component.

B.6 Preference learner

The long-term user profile adaptation is in charge of adapting the semantic preferences of the user based on his content consumption. The preference adaptation mechanism is explained and evaluated by Picault and Ribière in (Picault & Ribière, 2008), who also have integrated it in *News@hand* system. Because this functionality is out of the scope of this thesis, we do not describe it in detail herein. For further information, the reader is referred to:

- Cantador, I., Fernández, M., Vallet, D., Castells, P., Picault, J., & Ribière, M. (2007). A Multi-Purpose Ontology-based Approach for Personalised Content Filtering and Retrieval. *Book chapter in "Studies in Computational Intelligence"*, vol. 93, pp. 25-51. Springer-Verlag. Edited by M. Wallace, M. Angelides, and P. Mylonas. ISBN: 978-3-540-76359-8.

³³ Data Mining Software in Java, <http://www.cs.waikato.ac.nz/ml/weka/>

The component takes care of tracking all potential interests of a user, and detects whether an interest is confirmed with time. It decides when to introduce a concept as a new semantic preference in the user profile. In order to do so, a number of Java classes have been defined in the package *es.uam.nets.personalisation.preference.learning*, as shown in the following table.

Package: es.uam.nets.personalisation.preference.learning	
Class	Description
<i>ConceptHistoryStack</i>	Describes a mechanism to store all concepts representing potential user interests (because they occurred in the consumed contents).
<i>ContentConsumptionAnalysisManager</i>	Takes care of the analysis of consumed content.
<i>ContentConsumptionElement</i>	Describes data related to one content item used by the long term adaptation process.
<i>LongTermAdaptationModule</i>	Takes care of updating the persistent user profile (long term preferences) of the user according to the content consumption of the user.

Table B.15 Main classes of the semantic preference learning component.

In particular, the *LongTermAdaptationModule* stores all concepts representing potential user interests. This module is implemented as a timer that wakes up periodically (according to a specified update period; typically one day, one week – the value could be customised for each user according to his average item consumption). When the long term adaptation process is triggered, the persistent user profile is retrieved from the user profile manager, and then the updated semantic preferences are saved into the persistent user profile through the user profile manager.

The process of insertion of concepts into the user profile uses a *ConceptHistoryStack* as a mean to detect if a concept that appears periodically in the consumed content has to be incorporated inside the user profile or not. The class *ContentConsumptionAnalysisManager* interacts with the log component to retrieve the set of *ContentConsumptionElement* that are used in the profile adaptation process.

The long-term adaptation mechanism can be completely configured at run time, enabling to specify:

- If the module triggers the adaptation process automatically or not: automatically triggering of long-term adaptation is the regular mode for *News@hand*, as the profile is supposed to be updated periodically. On the other hand, the manual mode is useful for testing purposes;
- The update period frequency, which is the time between two consecutive updates of the user profile;

- A threshold for insertion of concepts as new preferences;
- A threshold for removal of concepts from users' preferences;
- The maximum size of the concept history stack;
- Parameters of the preference weight update formula explained in (Picault & Ribière, 2008):
 - A decay factor that affects the decrease of preference weight over time;
 - A parameter that defines the impact in terms of the increase/decrease of preference weights according to positive/negative implicit feedback.

B.7 Log manager

The system monitors the actions a user performs, and gathers them in a log database. This not only has allowed us to identify *software bugs* during the implementation and testing phases, but also to make an off-line analysis of the results obtained in the experiments described in Chapter 8. All the accessing, browsing, rating, querying, and user profile updating actions, together with the corresponding system parameters and outputs, are recorded in the log database. Table 7.16 shows the main attributes of the log database entities.

Table	Attributes
<i>Browsing</i>	actionID , actionType , timestamp, sessionID, itemID , itemRankingPosition , itemRankingProfile, itemRankingContext, itemRankingCollaborative, itemRankingHybridUP, itemRankingHybridNUP, itemRankingHybridUPq, itemRankingHybridNUPq, topicSection, userProfileWeight, contextWeight, collaborativeWeight
<i>Context updates</i>	actionID , actionType , timestamp, sessionID, context , origin
<i>Queries</i>	actionID , actionType , timestamp, sessionID, keywords , topicSection
<i>Recommendations</i>	actionID , actionType , timestamp, sessionID, recommendationType, userProfileWeight , contextWeight , collaborativeWeight , topicSection
<i>User accesses</i>	actionID , actionType , timestamp, sessionID
<i>User evaluations</i>	actionID , actionType , timestamp, sessionID, itemID , rating , userFeedback, tags, comments, topicSection, duration
<i>User preferences</i>	actionID , actionType , timestamp, sessionID, concept , weight
<i>User profiles</i>	actionID , actionType , timestamp, sessionID, userProfile
<i>User sessions</i>	sessionID , userID , timestamp

Table B.16 Summary of the log database tables and attributes. The most relevant attributes are in bold fonts.

The database records share a **session identifier** (*sessionID*) that enables to recognise relationships among actions. For example, given a row from the *user evaluation* table, we extract the session identifier, the rated item, and the action timestamp. Then, we infer which system settings were set at that moment, as follows:

- Get all the *browsing actions* matching the given session identifier.
- Select the browsing actions with item identifier (*itemID*) previously extracted from the user evaluation table.
- Use the timestamp to obtain from the selected browsing actions the searched system settings, such as the user profile weight (0 if personalisation was off), the context weight, etc.

The **action type** (*actionType*), however, is an intra-table identifier whose value distinguishes between different actions a table can contain. For instance, the *user access* table records only get two values for the *actionType* attribute: ACCESS_LOGIN and ACCESS_LOGOUT. Analogously, the permitted action type values in the *user preference* table are PREFERENCE_CREATE, PREFERENCE_UPDATE and PREFERENCE_DELETE.

Appendix C

Introducción

Este capítulo ofrece al lector una visión general de la tesis, poniendo énfasis en la definición de los problemas que la motivaron, el enunciado de las propuestas desarrolladas para abordarlos, y los resultados que finalmente se obtuvieron.

La Sección C.1 expone la motivación que ha originado este trabajo, planteando los problemas tratados, y citando las limitaciones de las aproximaciones presentadas en la literatura. La Sección C.2 acota el alcance del estudio exponiendo los objetivos parciales a alcanzar. A continuación, la Sección C.3 resume las contribuciones de las propuestas desarrolladas en este trabajo. La Sección C.4 describe la estructura de este documento, y finalmente, la Sección C.5 lista las publicaciones resultantes de la investigación emprendida en esta tesis.

C.1 Motivación

A lo largo de las dos últimas décadas hemos alcanzado un punto en la era de las telecomunicaciones en el que la información disponible inunda nuestras actividades cotidianas. La cantidad de nuevos contenidos que se producen cada día (noticias, artículos científicos, películas, canciones, páginas web, etc.), venciendo a las capacidades humanas de procesamiento, así como la naturaleza no estructurada de la mayoría de esa información, originan importantes preguntas acerca de su uso efectivo y utilidad.

Esta sobrecarga de información planteó la necesidad de diseñar sistemas capaces de llevar a cabo una **búsqueda de información** eficiente sobre miles de millones de documentos. La información que estos sistemas manejan no sólo consiste en páginas web, sino también en otros formatos de documentos de texto, y en cualquier tipo de ficheros de imagen, video y audio, apropiadamente anotados con metadatos textuales. Los documentos a recuperar son anotados con palabras clave que representan resumidamente sus contenidos. Para documentos textuales las anotaciones consisten en aquellos términos que son más “informativos” (e.g., que aparecen más frecuentemente en documentos individuales, pero que son poco comunes en el conjunto de la colección). Para contenidos multimedia las anotaciones involucran conceptos que son declarados manualmente por los usuarios o que son extraídos automáticamente mediante alguna técnica avanzada de procesamiento de señal. A partir de las anotaciones obtenidas se generan tablas de índices que asocian de forma ponderada cada palabra clave con los documentos donde aparece, y que están contruidos con estructuras de datos que permiten recuperar los documentos correspondientes a una palabra clave de forma muy rápida (Baeza-Yates & Ribeiro Neto, 1999). De este modo, los diferentes motores de búsqueda se distinguen esencialmente por los mecanismos de generación de anotaciones e índices, y por los algoritmos desarrollados para obtener documentos a partir de palabras clave.

En este escenario el usuario suele conocer sus objetivos en cuanto a la información que desea obtener, y posibles descripciones de la misma mediante palabras clave. Por ello, es capaz de introducir consultas mediante listas de términos. Así, por ejemplo, un usuario que está planificando sus vacaciones, y está interesado en recopilar documentos con información sobre la República de Indonesia (la cual está compuesta por más de 17.000 islas del Océano Pacífico), podría emplear consultas como “Indonesia”, “República de Indonesia”, “islas de Indonesia”, etc.

No hay duda alguna acerca del éxito que los sistemas de recuperación de información han obtenido en los últimos años al ofrecer servicios de búsquedas de contenidos en Internet. A partir de una consulta dada, motores de búsqueda comerciales como *Google* y *Yahoo!* seleccionan y muestran de forma ordenada, ponderada (atendiendo a similitudes entre consultas y anotaciones), y en tiempo real, listas de decenas a millones de documentos potencialmente relevantes. En muchos casos los resultados deseados por el usuario están situados en las primeras posiciones

de las listas. Sin embargo, hay ocasiones en las que esos documentos se encuentran en posiciones tales que el usuario no alcanza a descubrirlos. Existen por tanto diversos aspectos que no han sido resueltos satisfactoriamente por los sistemas actuales. Entre ellos, uno de los más importantes es la **ambigüedad semántica**. Supongamos que el usuario del ejemplo anterior centra su búsqueda de información sobre Indonesia en una de sus islas: Java. Para ello introduce la consulta “Java” en un motor de búsqueda web. Esperando encontrar documentos sobre la citada isla, se encuentra con la sorpresa de que ese concepto no aparece en ninguna de las páginas web correspondientes a los primeros resultados obtenidos con la consulta. En su lugar, le son mostradas todo tipo de páginas web acerca del bien conocido lenguaje de programación que comparte el mismo nombre. Es en posiciones alejadas del comienzo de la lista de resultados donde comienzan a aparecer páginas web que tratan aspectos de la isla.

En el ejemplo descrito los resultados deberían haberse priorizado atendiendo al significado del término “Java” en cada caso. La desambiguación podría haber sido posible si el sistema hubiera tenido en cuenta el conjunto de consultas introducidas por el usuario con anterioridad acerca de Indonesia. De alguna manera, se podrían haber medido “distancias semánticas” entre términos de consultas anteriores (i.e., Indonesia, república, isla, etc.), y términos que apareciesen en los documentos indexados y que estuviesen relacionados con los dos significados de la palabra Java descritos anteriormente, i.e., la isla indonesia y el lenguaje de programación. Así, se hubiera podido deducir que con alta probabilidad el usuario en este “contexto” estaba interesado en obtener documentos asociados al primer significado. En el ámbito de la recuperación de información, la consideración del contexto (obtenido de acciones recientes del usuario en el sistema) ha sido denominada **búsqueda de información contextualizada**.

El contexto semántico, entendido como en el ejemplo anterior, puede considerarse como un conjunto de preferencias de usuario definidas a corto plazo durante la sesión del usuario en el sistema. En un principio estas preferencias son temporales, y podrían describirse como intereses u objetivos actuales del usuario. Sin embargo, si se repitiesen en el tiempo con cierta frecuencia (e.g., diariamente), podrían pasar a formar parte de una descripción de intereses permanentes, que se conoce en la literatura como *perfil de usuario*. De manera análoga al contexto, este perfil podría entonces ser usado para modificar el orden en el que los resultados de una consulta son mostrados. Por ejemplo, supongamos dos usuarios. El primero de ellos tiene un perfil que ha sido construido (manual o automáticamente) con conceptos relacionados con destinos y alojamientos turísticos, agencias de viajes, etc. El segundo, sin embargo, es un ingeniero en informática que ha definido su perfil usando conceptos relacionados con sistemas operativos, aplicaciones de ordenador, etc. Supongamos que los dos usuarios introducen la consulta “Java” en un mismo

motor de búsqueda web, cuyo algoritmo de recuperación de información subyacente recupera los contenidos atendiendo a las preferencias de usuario. Entonces, podría comprobarse que las listas de resultados proporcionadas a los dos usuarios son distintas. El primero recibiría una lista en la que los primeros documentos serían aquellos que hablasen sobre la isla indonesia, mientras que el segundo obtendría otra lista en la que los primeros resultados estarían relacionados con el lenguaje de programación. Este tipo de aplicaciones es referenciado en la literatura como sistemas de **búsqueda de información personalizada**.

Por supuesto, el contexto actual no tiene que necesariamente coincidir siempre con las preferencias del perfil de usuario. Siguiendo el ejemplo anterior, un ingeniero informático podría estar interesado en obtener información sobre la isla Java incluso por motivos profesionales al tener que asistir a alguna reunión o conferencia en la citada isla. Un equilibrio entre contextualización y personalización podría ser la clave para la obtención de resultados de búsqueda más precisos y relevantes al usuario.

En cualquier caso, hasta este punto, e independientemente del hecho de considerar contexto o preferencias personales, el usuario es consciente de sus necesidades y objetivos de búsqueda de información, y parece conocer la manera en la que reflejarlos mediante consultas basadas en palabras clave. Ahora bien, esto no siempre es así. Cada día, al salir a la calle, leer el periódico, ver la televisión, escuchar la radio, o simplemente charlar con un amigo, nos enteramos de hechos cuya existencia nos era desconocida, que no estábamos buscando, pero que son importantes o interesantes para nosotros, y que incluso pueden llegar a afectar de forma trascendental a nuestras propias vidas.

El “**boca a boca**” es una técnica que consiste en pasar información por medios verbales, especialmente recomendaciones, de una manera informal, personal, más que a través de medios de comunicación, anuncios, publicación organizada o *marketing* tradicional. Típicamente se considera una comunicación hablada, aunque los diálogos en Internet, por ejemplo, en blogs, foros o e-mails a menudo se incluyen en la definición. La promoción basada en el boca a boca es altamente valorada por los vendedores. Se siente que esta forma de comunicación tiene credibilidad valiosa a causa de la fuente de la que proviene. La gente está más inclinada a creer la palabra del boca a boca que medios más formales de promoción porque es poco probable que el comunicador tenga un interés ulterior (e.g., no intenta vender algo). También la gente tiende a creer a la gente que conoce.

En palabras de Jeffrey M. O'Brien, extraídas de su artículo “The race to create a ‘smart’ Google” publicado en CNN Money en noviembre de 2006:

Estamos abandonando la era de búsqueda y entrando en una de descubrimiento. ¿Cuál es la diferencia? La búsqueda es lo que uno hace cuando está intentando encontrar algo. El descubrimiento se da cuando algo maravilloso que uno no sabía que existía o por el que no sabía cómo preguntar, te encuentra.

Para afrontar este nuevo reto, a mediados de los noventa, los **sistemas de recomendación** surgen como un campo de investigación independiente de la Recuperación de Información y la Inteligencia Artificial. El objetivo de los investigadores se centra entonces en estimar la relevancia de aquellos ítems que todavía no han sido vistos por el usuario, sin necesidad de que este último los busque. La manera en que la estimación anterior es llevada a cabo permite distinguir dos tipos principales de estrategias de recomendación (Adomavicius & Tuzhilin, 2005): la basada en contenido y la basada en filtrado colaborativo.

Los sistemas de recomendación basados en contenido (del inglés *content-based recommender systems*) calculan la relevancia de un ítem para un usuario atendiendo a la relevancia que otros ítems “similares” tuvieron en el pasado para el usuario. Las medidas de similitud entre ítems están basadas en características de sus contenidos. Así, por ejemplo, un sistema de recomendación turístico podría sugerir alojamientos en diversos países de Oceanía a un usuario con historial de vuelos a Indonesia, ya que este país se encuentra en el citado continente.

En estos sistemas la ventaja inicial de que las recomendaciones proporcionadas a un usuario son un fiel reflejo de sus preferencias, obtenidas a partir de acciones y valoraciones personales pasadas sobre diversos ítems, puede convertirse en un gran inconveniente. Al tener en cuenta únicamente el perfil de usuario, el espacio de ítems novedosos potencialmente interesantes para el usuario se ve limitado a aquellos que comparten características con ítems ya vistos. La sobre-especialización (del inglés *content over-specialisation*) y falta de diversidad (en inglés, *portfolio effect*) en las recomendaciones son de hecho dos de los problemas más notables de este tipo de estrategias.

Para solventar estos problemas los sistemas de filtrado colaborativo (del inglés *collaborative filtering systems*) calculan la relevancia de un ítem para un usuario atendiendo a la relevancia que otros ítems tuvieron en el pasado para personas “similares”. En este caso las medidas de similitud entre usuarios se calculan a partir de correlaciones entre sus patrones de evaluación de ítems. Por ejemplo, supongamos que una gran mayoría de las personas que han viajado a Jakarta, la capital de Indonesia, también lo han hecho al país vecino Singapur, valorando positivamente sus estancias. Un sistema de filtrado colaborativo podría recomendar al usuario con historial de vuelos a Indonesia alojamientos en Singapur, a pesar de que nunca haya sido reflejado en su perfil cierto interés por este último país.

De este modo, el filtrado colaborativo no limita el espacio de recomendaciones, y evita la sobre-especialización y no diversidad de contenidos. Sin embargo, incorpora limitaciones propias, entre las cuales destaca el problema de las “ovejas negras” (en inglés, *grey sheep*, ovejas grises), que se define como la dificultad de recomendar ítems a usuarios particulares con preferencias (patrones de evaluación) poco comunes, muy diferentes a los del resto de usuarios.

El problema anterior se podría solventar incorporando una estrategia basada en contenido. De hecho, para abordar conjuntamente las limitaciones características de cada uno de los dos tipos de recomendación expuestos – basado en contenido y basado en filtrado colaborativo – se propone en la literatura la combinación de ambos en los denominados **sistemas de recomendación híbridos** (del inglés *hybrid recommender systems*).

En la actualidad el interés por los sistemas de recomendación está en alza, constituyendo una parte esencial de un gran número de importantes portales de comercio electrónico como *Amazon.com*, donde se ofrecen recomendaciones de libros, *FilmAffinity.com*, donde se recomiendan películas, *Last.fm*, que recomienda canciones y grupos musicales, o *Google News* (*news.google.com*), que proporciona recomendaciones personalizadas de noticias. En todos ellos el uso de modelos de recomendación clásicos ha sido muy exitoso. No obstante, la generación actual de sistemas de recomendación todavía requiere mejoras adicionales para hacer los algoritmos más eficaces y aplicables a una mayor gama de dominios. Estas mejoras incluyen, entre otras:

- La utilización de estrategias que aborden situaciones iniciales en las que se disponen de pocas preferencias o evaluaciones de los usuarios (problema del arranque frío, del inglés *cold-start problem*), y situaciones en las que hay poca densidad de correlaciones entre evaluaciones debido al elevado número relativo de usuarios o ítems (en inglés, *sparsity problem*).
- La adición de información contextual en los procesos de recomendación.
- El uso de algoritmos más flexibles, que puedan ser adaptados por el usuario, o que permitan hacer recomendaciones no sólo a un único usuario, sino también a un grupo de usuarios con gustos e intereses similares.

La manera en la que estos aspectos pueden ser parcial o totalmente resueltos de forma satisfactoria representa líneas de investigación abiertas en el área. Las dificultades planteadas por cada uno de los aspectos anteriores han sido abordadas hasta el momento de forma independiente, pero no se han establecido modelos de recomendación que permitan afrontarlas de forma conjunta y efectiva.

Esta tesis aboga que la principal razón de estas dificultades es la falta de comprensión y explotación de la semántica subyacente tanto en las preferencias de los usuarios como en las características de contenido de los ítems. Los modelos clásicos describen los perfiles de usuario e ítem como listas de palabras clave (aproximaciones basadas en contenido) o evaluaciones numéricas (aproximaciones basadas en filtrado colaborativo). Las componentes de estas listas aparentemente no están relacionadas entre sí, y su significado (semántico) no es tenido en cuenta a la hora de hacer recomendaciones.

En sistemas de recomendación la necesidad de una **representación semántica del conocimiento** que permita describir los dominios involucrados de forma sencilla, escalable y portable está siendo manifestada en recientes trabajos (Middleton, Roure, & Shadbolt, 2004; Mobasher, Jin, & Zhou, 2004; Anand & Mobasher, 2007; Sieg, Mobasher, & Burke, 2007; Shoval, Maidel, & Shapira, 2008). Debido a que los gustos e intereses de los usuarios son definidos sobre los contenidos de los ítems a recomendar, perfiles de usuario e ítem han de ser contruidos sobre una representación de conocimiento común. Esta representación debería ser comprensible por humanos, y procesable por máquinas (programas de ordenador). Además, debería ser fácilmente ampliable, y modificable a otros dominios. El ideal consistiría en que la información recogida por un sistema de recomendación dado pudiese ser explotada por otros sistemas diferentes, aunque manejasen ítems de naturaleza muy dispar. Para ello, sería conveniente el uso de lenguajes y modelos de representación de conocimiento estándares.

En esta tesis se propone el uso de **ontologías** como vehículo conductor a satisfacer la necesidad anterior. Tanto en ciencias de la computación como en ciencias de la información, una ontología es una representación formal de un conjunto de conceptos pertenecientes a un dominio, y de las relaciones existentes entre esos conceptos (Gruber, 1993). Se puede usar para definir el dominio en cuestión y/o para razonar sobre las propiedades del mismo. Las ontologías se emplean como forma de representación del conocimiento sobre el mundo o parte de él en campos tan diversos como la Inteligencia Artificial, la Web Semántica, la Ingeniería del Software, la Informática Biomédica o la Biblioteconomía. Algunos de los elementos fundamentales de una ontología son: los *individuos* (instancias u objetos básicos de información), las *clases* (categorías, conjuntos, tipos de objetos), los *atributos* (aspectos, propiedades, características que individuos y clases pueden tener), y las *relaciones* (atributos especiales que relacionan pares de clases y/o individuos).

Más específicamente, este trabajo propone un modelo de representación de conocimiento tricapa en el que se incorpora un espacio de conceptos semánticos interrelacionados (mediante ontologías, y describiendo uno o varios dominios de aplicación), entre los espacios de usuarios e ítems. En este modelo los perfiles de usuario e ítem son definidos mediante vectores cuyas componentes son conceptos ponderados del espacio ontológico. Sobre esa forma de representación del conocimiento se plantea y evalúa una serie de mecanismos de recomendación orientados a uno o varios usuarios, combinando estrategias basadas en contenido y filtrado colaborativo, e incorporando información contextual semántica obtenida de anotaciones de ítems involucrados en acciones y evaluaciones recientes del usuario. La implementación y puesta en marcha integrada de los mecanismos anteriores en un sistema de recomendación de noticias también son presentadas.

La oportunidad de incorporar meta-información a los perfiles de los usuarios y a las descripciones de los ítems recomendados, junto con la capacidad de inferir conocimiento a partir de las relaciones semánticas existentes entre los conceptos de las ontologías de dominio, serán los aspectos clave de las propuestas expuestas.

C.2 Objetivos

El objetivo final de esta tesis es la implementación y evaluación de una serie de modelos de recomendación asistidos por la incorporación de un espacio conceptual entre las preferencias de usuario y las características de contenido de los ítems a recomendar. Identificando y explotando las relaciones subyacentes entre usuarios e ítems, los modelos propuestos deberían abordar limitaciones existentes en los sistemas de recomendación actuales.

Procedentes de técnicas clásicas de recuperación de información, los sistemas de recomendación basados en contenido generalmente representan las preferencias de un usuario y las características de los ítems mediante vectores de términos. A partir de estas representaciones se calculan similitudes vectoriales (e.g., a través del coseno del ángulo formado por los vectores) que son usadas como medidas de relevancia personal de los diferentes ítems. Así, por ejemplo, supóngase que el perfil de un usuario viene dado por el vector $\mathbf{u} = (\text{indonesia} = 0,7; \text{java} = 0,9; \text{isla} = 0,2)$, donde cada término tiene asociado un peso en $[0,1]$ que da idea de la preferencia del usuario por ese concepto. Supóngase un ítem cuyo contenido está descrito mediante el vector $\mathbf{d} = (\text{java} = 0,6; \text{isla} = 0,5)$. Un modelo de recomendación sencillo que calculase el coseno entre los vectores \mathbf{d} y \mathbf{u} devolvería una preferencia de 0,38:

$$\text{pref}(\mathbf{d}, \mathbf{u}) = \cos(\mathbf{d}, \mathbf{u}) = (0,6 \cdot 0,9 + 0,5 \cdot 0,2) / \left(\sqrt{0,6^2 + 0,5^2} \cdot \sqrt{0,7^2 + 0,9^2 + 0,2^2} \right) = 0,38.$$

Este modelo conlleva dos principales problemas. El primero de los problemas está asociado a la *ambigüedad semántica* de los términos. En el ejemplo “java” se refiere a una preferencia del usuario por la isla indonesia. Ahora bien, considérense dos nuevos ítems $\mathbf{d}_1 = (\text{java} = 0,4; \text{hotel} = 0,8)$ y $\mathbf{d}_2 = (\text{java} = 0,4; \text{software} = 0,8)$. En \mathbf{d}_1 la componente “java” se refiere de nuevo a la citada isla, pero en \mathbf{d}_2 se refiere al lenguaje de programación que comparte el mismo nombre. Los significados subyacentes al término “java” son totalmente diferentes en los dos ítems. Sin embargo, al calcular las similitudes del perfil de usuario \mathbf{u} con los vectores \mathbf{d}_1 y \mathbf{d}_2 se obtiene que $\text{pref}(\mathbf{d}_1, \mathbf{u}) = \text{pref}(\mathbf{d}_2, \mathbf{u}) = 0,19$, dando de este modo la misma preferencia a los dos ítems, cuando el segundo potencialmente carece de interés para el usuario. En este caso la distinción entre los dos conceptos semánticos, e.g.,

$\mathbf{d}_1 = (\text{isla:java} = 0,4; \text{hotel} = 0,8)$, $\mathbf{d}_2 = (\text{programacion:java} = 0,4; \text{software} = 0,8)$ y $\mathbf{u} = (\text{indonesia} = 0,7; \text{isla:java} = 0,9; \text{isla} = 0,2)$, es esencial para no producir recomendaciones indeseables.

El segundo de los problemas es la *suposición de independencia* entre los términos. Supóngase ahora los siguientes dos ítems: $\mathbf{d}_1 = (\text{java} = 0,4; \text{hotel} = 0,8)$ y $\mathbf{d}_2 = (\text{java} = 0,4; \text{archipiélago} = 0,8)$. En este caso el término “java” se refiere a la isla en los dos ítems, y las preferencias dadas a ambos son de nuevo $\text{pref}(\mathbf{d}_1, \mathbf{u}) = \text{pref}(\mathbf{d}_2, \mathbf{u}) = 0,19$. Sin embargo, atendiendo a las preferencias del perfil $\mathbf{u} = (\text{indonesia} = 0,7; \text{isla:java} = 0,9; \text{isla} = 0,2)$, se puede asumir que el ítem \mathbf{d}_2 debería obtener una relevancia mayor, pues el concepto “archipiélago” (i.e., conjunto de islas) está más relacionado con la preferencia “isla” que el concepto “hotel”, incluido en el ítem \mathbf{d}_1 . La necesidad de considerar relaciones (semánticas) entre conceptos a la hora de recomendar ítems se hace evidente con este ejemplo.

La conclusión que se puede obtener de las dos limitaciones anteriores ya ha sido mencionada alguna vez en la literatura (Balabanovic & Shoham, 1997; Ungar & Foster, 1998): en muchos sistemas de recomendación actuales hay una ***falta de entendimiento y explotación de la semántica*** subyacente a los gustos e intereses de los usuarios y a los contenidos de los ítems recomendados. Para abordar este problema, el primer objetivo que se establece en la tesis es:

- O1. **La definición de una representación formal del conocimiento que no sea ambigua y que tenga en cuenta relaciones entre conceptos.** Se estudiarán propuestas basadas en ontologías. Tanto los perfiles de usuario como las descripciones de los ítems estarán compuestos de conceptos (clases e instancias) pertenecientes a múltiples ontologías de dominio. Las relaciones semánticas entre conceptos, que vendrán definidas en las ontologías, deberían ser explotadas por los diferentes modelos de recomendación que se planteen.

En una representación ontológica las relaciones semánticas enriquecen el significado de cada concepto. Así, por ejemplo, si un usuario muestra un interés genérico alto por aspectos relacionados con islas, con un perfil $\mathbf{u} = (\text{isla} = 0,9)$, se podría asumir que también compartiría cierta afinidad por islas específicas. De este modo, la extensión de su perfil a por ejemplo $\mathbf{u} = (\text{isla} = 0,9; \text{isla:java} = 0,1)$ no sólo podría resultar correcta, sino también beneficiosa para encontrar más ítems relevantes. En este caso la extensión de preferencias se ha realizado a través de la propiedad “instancia de” (del inglés “*instance of*”) que relaciona una clase (isla) con un individuo concreto de la misma (Java). Existen otros tipos de relaciones. Algunas de ellas son comunes a toda representación ontológica, como por ejemplo la relación “subclase de” (del inglés “*subclass of*”: “isla continental” e “isla oceánica” son subclases

de “isla”. Otras, sin embargo, están definidas de forma arbitraria en el dominio descrito por la ontología. Por ejemplo, en una ontología sobre Geografía podría estar definida la relación “capital de”: una “ciudad” es la capital de un “país”, “Jakarta” es la capital de “Indonesia”.

La extensión de preferencias hace que los perfiles de usuario estén menos dispersos en el espacio conceptual, al cubrir mayores áreas de este último. La “escasez” o poca densidad de preferencias y evaluaciones (del inglés *sparsity problem*) es un problema que ha sido abordado en diversos trabajos (Billsus & Pazzani, 1998; Sarwar, Karypis, Konstan, & Riedl, 2000). Está estrechamente relacionado con el problema del “arranque rápido” (del inglés *cold-start problem*), que consiste en la dificultad de recomendar ítems al usuario cuando éste comienza su actividad en el sistema teniendo ninguna o pocas preferencias declaradas (Schein, Popescul, & Ungar, 2001). Estos dos efectos no son sólo característicos de los modelos de recomendación basados en contenido, sino que también ocurren en las estrategias de filtrado colaborativo. Para afrontarlos la **necesidad de enriquecer las descripciones semánticas** ofrecidas por una representación del conocimiento basada en ontologías da lugar al segundo objetivo en la tesis:

O2. El enriquecimiento de los perfiles de usuario y las descripciones de ítem a través de la explotación de las relaciones entre sus conceptos.

Se investigarán estrategias que propaguen las preferencias de usuario y las características de contenido de los ítems hacia conceptos enlazados a través de relaciones existentes en las ontologías de dominio. La propagación deberá considerar aspectos como la atenuación de los pesos asociados a los conceptos expandidos, o la posibilidad de encontrar bucles en los caminos de propagación realizados. Además, se deberá evaluar el efecto producido por la propagación semántica sobre los resultados obtenidos con los modelos de recomendación que se propongan.

Aparte de enriquecer las descripciones semánticas de usuarios e ítems, una representación del conocimiento ontológica mejora el entendimiento de las mismas. Este hecho facilitaría la comprensión de los conceptos involucrados en el contexto actual de un entorno de recuperación o recomendación de contenidos. La *contextualización de recomendaciones* con modelos clásicos es una tarea compleja. Es de hecho una línea de investigación abierta, y ha empezado a ser estudiada en trabajos recientes (Räck, Arbanowski, & Steglich, 2006; Anand & Mobasher, 2007; Vallet, Castells, Fernández, Mylonas, & Avrithis, 2007). En la sección C.1 la contextualización se motivó con un ejemplo particular de desambiguación semántica del término “Java”. Los conceptos que anotaban resultados de consultas anteriores (e.g., Indonesia, república, isla, etc.) eran utilizados para inferir que “Java”, en el contexto actual, se refería a la isla indonesia, en vez de al lenguaje de programación. Otra posible aplicación de la contextualización es la focalización o reforzamiento de

preferencias de usuario. Aquellos conceptos que han sido referenciados recientemente (e.g., a través de evaluaciones de ítems) podrían ser más tenidos en cuenta que el resto por los modelos de recomendación.

La representación del conocimiento planteada también añade flexibilidad a la hora de recomendar ítems, permitiendo aplicar estrategias de *combinación de perfiles de usuario* de forma sencilla. Varios vectores que describan las preferencias de un conjunto de usuarios pueden ser agregados para generar un único perfil de grupo, empleado posteriormente para recomendar ítems de forma colectiva. Como ejemplo ilustrativo, sean u_1 y u_2 dos usuarios cuyos perfiles vienen dados por los vectores $u_1 = (\text{indonesia} = 0,6; \text{java} = 0,9)$ y $u_2 = (\text{java} = 0,1; \text{isla} = 0,4)$. Suponiendo que los dos vectores se combinan mediante la suma promedio de sus componentes, el perfil de grupo resultante sería $u_g = (\text{indonesia} = 0,3; \text{java} = 0,5; \text{isla} = 0,2)$. En la literatura, las recomendaciones orientadas a grupo se han propuesto para muy diversas aplicaciones, como por ejemplo, la recomendación colectiva de composiciones musicales (McCarthy & Anagnost, 1998), películas (O'Connor, Cosley, Konstan, & Riedl, 2001), atracciones turísticas (Ardissono, Goy, Petrone, Segnan, & Torasso, 2003), o programas de televisión (Ali & Van Stam, 2004).

Los dos aspectos anteriores son ejemplos que evidencian la ***necesidad de flexibilidad*** en los sistemas de recomendación, y motivan el tercer objetivo de esta investigación:

- O3. La creación de un modelo de recomendación personalizada que permita la incorporación de contexto semántico, y que pueda ser adaptado a las preferencias de uno o más usuarios.** Se propondrá un modelo de recomendación basado en contenido que haga uso de la representación del conocimiento ontológica propuesta. Este modelo deberá ser flexible para adaptarse a recomendaciones contextualizadas y orientadas a grupos. Se deberá evaluar el efecto que la adición de contexto semántico supone en los resultados del modelo básico, y se deberán estudiar diferentes estrategias para la combinación de perfiles de usuario.

Como ya se mencionó en la sección C.1, los sistemas de recomendación basados en contenido se centran en las preferencias de un usuario único, y no explotan los beneficios que ofrecen técnicas basadas en el fenómeno del “boca a boca” para descubrir ítems relevantes para el usuario, que no están explícitamente relacionados con sus preferencias, sino que son recomendados por personas con gustos e intereses similares. El hecho de centrarse en un único perfil conlleva a una sobre-especialización de los contenidos recuperados (del inglés *content over-specialisation*) y a una falta de diversidad en las recomendaciones (en inglés, *portfolio effect*).

Para solventar estos problemas se propusieron estrategias de filtrado colaborativo. Estas aproximaciones están basadas en el cálculo de similitudes

(correlaciones) entre perfiles de usuario e ítem, y su eficacia está demostrada por el éxito de su implantación en aplicaciones comerciales reales. Sin embargo, incorporan nuevas limitaciones. Una de ellas es la conocida como el problema de la “oveja negra” (en inglés, *grey sheep*, oveja gris), que consiste en la dificultad de recomendar ítems a personas con preferencias muy particulares, poco comunes en el resto de usuarios, y que no permiten encontrar correlaciones entre ellos. Modelos de recomendación híbridos que combinen características basadas en contenido y colaborativas pueden ser adecuados para afrontar el problema anterior.

En general, la comparación entre usuarios e ítems es realizada de forma global, de tal modo que similitudes parciales, pero fuertes y útiles pueden perderse. Por ejemplo, dos personas pueden tener una alta coincidencia en los lugares a visitar preferidos, pero pueden ser muy divergentes en cuanto al tipo de alojamientos frecuentados. Las opiniones de estas personas sobre destinos turísticos podrían ser altamente valiosas para cada una de ellas, pero podrían ser ignoradas por un sistema de recomendación de viajes al computar una similitud global entre perfiles relativamente baja. Sean de nuevo dos usuarios u_1 y u_2 cuyos perfiles se definen respectivamente mediante los vectores $\mathbf{u}_1 = (\text{java} = 0,4; \text{singapur} = 0,6; \text{hotel} = 0,8)$ y $\mathbf{u}_2 = (\text{java} = 0,5; \text{camping} = 0,7)$. La similitud basada en el coseno entre los dos vectores es 0,25:

$$\text{sim}(u_1, u_2) = \cos(\mathbf{u}_1, \mathbf{u}_2) = (0,4 \cdot 0,5) / \left(\sqrt{0,4^2 + 0,6^2 + 0,8^2} \cdot \sqrt{0,5^2 + 0,7^2} \right) = 0,25.$$

Ahora bien, supóngase que el sistema es capaz de identificar y agrupar por separado preferencias relacionadas con lugares turísticos y preferencias asociadas a tipos de alojamientos. Atendiendo a estos dos grupos de conceptos, los perfiles de usuario podrían dividirse en dos sub-perfiles diferentes. Para el usuario u_1 :

$$\mathbf{u}_1^{\text{lugares}} = (\text{java} = 0,4; \text{singapur} = 0,6), \mathbf{u}_1^{\text{alojamientos}} = (\text{hotel} = 0,8).$$

Para el usuario u_2 :

$$\mathbf{u}_2^{\text{lugares}} = (\text{java} = 0,5), \mathbf{u}_2^{\text{alojamientos}} = (\text{camping} = 0,7).$$

Calculando el coseno del ángulo formado por los vectores de los dos grupos de preferencias se encuentran nuevas similitudes entre los usuarios. En el caso del grupo relacionado con lugares turísticos, la similitud es más del doble que la global.

$$\text{sim}_{\text{lugares}}(u_1, u_2) = \cos(\mathbf{u}_1^{\text{lugares}}, \mathbf{u}_2^{\text{lugares}}) = (0,4 \cdot 0,5) / \left(\sqrt{0,4^2 + 0,6^2} \cdot \sqrt{0,5^2} \right) = 0,53.$$

En el caso del grupo relativo a tipos de alojamiento, la similitud es nula:

$$\text{sim}_{\text{alojamientos}}(u_1, u_2) = \cos(\mathbf{u}_1^{\text{alojamientos}}, \mathbf{u}_2^{\text{alojamientos}}) = 0 / \left(\sqrt{0,8^2} \cdot \sqrt{0,7^2} \right) = 0.$$

Si el sistema fuese capaz de discernir el contexto actual, podría proporcionar recomendaciones muy dispares, pero acertadas en cada caso. En el ejemplo anterior, si se tienen en cuenta sólo preferencias por destinos turísticos, al usuario u_2 se le podrían recomendar paquetes de viajes a Singapur, pues esta ciudad fue valorada positivamente por el usuario u_1 , con el que comparte interés por la isla de Java. Por el contrario, si se consideran únicamente las preferencias por tipos de alojamiento, al usuario u_2 no se le recomendaría ítem alguno en función del perfil del usuario u_1 .

Motivado por la ***dificultad de recomendar ítems a usuarios con preferencias poco usuales***, o a usuarios que comparten intereses sólo en determinados ámbitos semánticos, el cuarto objetivo de esta tesis es el siguiente:

- O4. **La creación de modelos híbridos que combinen los perfiles de usuario de forma colaborativa en varios contextos semánticos, atendiendo a diferentes grupos de preferencias compartidas.** Se definirán estrategias de recomendación híbrida que agrupen preferencias de usuario compartidas, y que a partir de los grupos generados, calculen similitudes entre usuarios e ítems basadas en la semántica de sus descripciones. Será necesario contrastar los resultados de recomendación obtenidos con los modelos propuestos contra los obtenidos con técnicas clásicas de filtrado colaborativo.

La evaluación de los sistemas de recomendación es también una línea de investigación abierta en la literatura (Herlocker, Konstan, Terveen, & Riedl, 2004; Adomavicius & Tuzhilin, 2005). En el caso de las propuestas expuestas en esta tesis, la puesta a punto de un entorno de experimentación plantea interrogantes en relación a la definición de las ontologías de dominio, la anotación semántica de ítems, y la creación de perfiles de usuario.

Con el propósito de llevar a cabo una ***evaluación de los modelos de representación del conocimiento y de recomendación*** basados en ontologías, el quinto y último objetivo en la tesis es:

- O5. **La integración y evaluación de todos los modelos de recomendación en un prototipo.** Se construirá un sistema de recomendación con el que se validen las propuestas de la tesis. En el proceso de implementación del sistema habrá que diseñar, desarrollar y evaluar técnicas que automáticamente creen las bases de conocimiento (i.e., procesos de instanciación de ontologías y de anotación semántica de ítems), y que faciliten la definición de perfiles a los usuarios.

C.3 Contribuciones

Los trabajos presentados en esta tesis contribuyen al desarrollo de modelos y

algoritmos que hacen uso de tecnologías basadas en semántica para abordar limitaciones existentes de los sistemas de recomendación actuales. Sus contribuciones más importantes se resumen en los siguientes puntos:

- **Explotación de capacidades ofrecidas por ontologías para enriquecer las funcionalidades de los sistemas de recomendación que conforman el estado del arte.** Se propone un modelo de representación del conocimiento que es más rico y menos ambiguo que modelos basados en palabra clave o ítem. La definición de las preferencias de usuario y de los atributos de ítem a través de conceptos semánticos pertenecientes a ontologías de dominio facilita al usuario final entender su perfil y las recomendaciones basadas en contenido obtenidas. El modelo proporciona una base adecuada para la representación de los intereses de usuario, tanto los más generales como los más refinados (e.g., intereses por ítems como un equipo de fútbol, un actor, o un valor en bolsa), y puede ser clave para tratar las sutilezas de las preferencias de usuario. Una ontología proporciona un significado de los conceptos más formal y procesable por máquinas (quién es el entrenador de un equipo de fútbol, la filmografía de un actor, los datos financieros de un valor en bolsa), y hace disponible este significado a un sistema de recomendación para que tome ventaja de él. Los lenguajes de descripción de ontologías estándar soportan mecanismos de inferencia que pueden ser usados para mejorar las recomendaciones. Así, por ejemplo, a un usuario interesado en películas sobre *hechos históricos* (superclase de *conflictos bélicos*), se le podrían recomendar películas sobre *guerras*. Similarmente, un usuario al que le gustan videos sobre *España* podría recibir recomendaciones de videos que traten de *Madrid*, a través de la relación transitiva *localizadoEn*. Los modelos de recomendación presentados en esta investigación harán uso de los tipos de inferencia semántica anteriores. Las primeras secciones del Capítulo 4 presentan el modelo de representación del conocimiento basado en ontologías propuesto, exponiendo con más detalle las ventajas que ofrece.
- **Desarrollo de novedosas aproximaciones a recomendación semántica colaborativa y basada en contenido.** Se proponen varios modelos de recomendación híbridos que combinan información semántica colaborativa y basada en contenido. En estos modelos las relaciones existentes entre conceptos de ontologías de dominio son explotadas para extender las preferencias de usuario y las anotaciones de ítem. En escenarios reales los perfiles de usuario suelen ser muy poco densos (con un número de preferencias/evaluaciones muy pequeño respecto al total de conceptos disponibles), particularmente en aquellos casos donde los usuarios tienen que declarar explícitamente sus intereses. Los usuarios no desean emplear tiempo describiendo al sistema sus preferencias detalladamente, y menos asignándoles

pesos, especialmente si no tienen un entendimiento claro de los efectos y resultados de sus elecciones. Por otra parte, aplicaciones en las que se utilizan algoritmos automáticos de aprendizaje de preferencias tienden a reconocer características muy generales de las preferencias del usuario, de este modo pudiendo producir perfiles que conlleven una falta de expresividad. Aparte de los perfiles de usuario, las descripciones de ítem también se enriquecerán. Sistemas de filtrado colaborativo sufren del bien conocido problema de arranque frío (Burke, 2002), en el cual un nuevo ítem no puede ser recomendado hasta que sea evaluado por al menos un usuario. En esta situación no existe información colaborativa alguna, el uso de aproximaciones basadas en contenido es esencial, y técnicas que mejoren las descripciones de los contenidos pueden resultar muy beneficiosas para encontrar correlaciones entre características de los ítems y los intereses de los usuarios. Por todas estas razones, los métodos de recomendación desarrollados en esta tesis hacen uso de una técnica que extiende las preferencias de usuario y las anotaciones de ítem de acuerdo a la semántica existente en las ontologías de dominio. Esta técnica estará basada en estrategias de activación de propagación restringida, del inglés *Constrained Spreading Activation* (Cohen & Kjeldsen, 1987; Crestani & Lee, 2000). Específicamente, los pesos de aquellas preferencias y anotaciones más relacionadas con el contexto actual son iterativamente propagadas a través de relaciones de las ontologías, generando versiones extendidas de perfiles de usuario y descripciones de ítem que serán usadas para proporcionar las recomendaciones personalizadas finales. La técnica de propagación semántica es presentada en el Capítulo 4, mientras que los modelos de recomendación híbridos son explicados detalladamente en el Capítulo 5. La evaluación de los modelos es descrita en el Capítulo 6.

- **Presentación de ideas novedosas para recomendación semántica contextualizada y orientada a grupos.** En general, los sistemas de recomendación no son flexibles en el sentido de que soportan un tipo de recomendaciones predefinido y fijo. La mayoría hacen uso de evaluaciones basadas en un único criterio, y sólo recomiendan ítems individuales a un usuario, sin tratar la agregación de ítems y/o usuarios. Por estos motivos, el usuario final no puede personalizar las recomendaciones de acuerdo a sus necesidades. Las representaciones del conocimiento y del perfil de usuario basadas en ontologías que se proponen en esta tesis han permitido el desarrollo de estrategias que proporcionan flexibilidad a los procesos de recomendación. En concreto, se ha usado un modelo que realiza consultas a una ontología para la recuperación personalizada de contenidos, se ha incluido información contextualizada en las recomendaciones, se han estudiado mecanismos que combinan varios perfiles de usuario para la

recomendación de ítems a grupos de usuarios, y se ha diseñado una técnica que hace uso de evaluaciones multi-criterio. Las últimas secciones del Capítulo 4 describen los mecanismos de recomendación anteriores, y el Capítulo 6 presenta experimentos que se han realizado para evaluarlos de forma independiente.

- **Implementación de un sistema de recomendación basado en ontologías.** Los modelos de recomendación propuestos en la tesis fueron evaluados con usuarios reales y conjuntos de datos artificiales creados a partir de fuentes externas. De forma aislada e independiente, cada experimento proporcionó resultados positivos que avalan la factibilidad de las propuestas. Sin embargo, se vio la necesidad de llevar a cabo experimentación adicional en un entorno que integrase los diferentes modelos combinando sus salidas, y con el que se estudiaran las dificultades originadas al trasladar los modelos a una aplicación realista. Por ello, se implementó *News@hand*, un sistema de recomendación de noticias en el que contenidos textuales de noticias son anotados con conceptos (clases e instancias) de un conjunto de ontologías que cubren diversos dominios de interés. Al construir el sistema surgieron retos de investigación para los cuales se han desarrollado novedosas soluciones. En particular, se tuvo que desarrollar una técnica de poblado (i.e., creación de instancias) de las ontologías de dominio, un mecanismo automático de anotación semántica de los artículos, y una estrategia de conversión de etiquetas (del inglés *tags*) o palabras clave a conceptos existentes. El Capítulo 8 describe la arquitectura e interfaz gráfica de usuario de *News@hand*, y el Capítulo 9 expone experimentos realizados para evaluar tanto las funcionalidades de recomendación del sistema como los mecanismos de creación de instancias, anotaciones y preferencias semánticas planteados.

C.4 Estructura de la tesis

El objetivo principal de esta tesis es la aplicación de modelos y técnicas basadas en semántica para afrontar algunas de las limitaciones existentes en sistemas de recomendación actuales. La multidisciplinariedad de este área de investigación implica abordar campos muy diversos, como el modelado de usuario y grupos de usuario, o la recuperación de información personalizada. Teniendo en cuenta que una amplia descripción del estado del arte en estos campos al comienzo de la tesis podría ser poco atractiva para el lector, la revisión de la literatura se ha distribuido en las diferentes partes que componen este documento. Sin embargo, para ofrecer una descripción inicial del contexto en el que se enmarca la tesis, sus dos primeros capítulos han sido dedicados a una exploración general de las principales áreas de investigación abordadas – sistemas de recomendación, y representación y

recuperación de información semánticas – y a una explicación más detallada de trabajos que pueden ser considerados la intersección de aquellas.

La tesis está dividida en tres partes. La primera parte proporciona conocimientos fundamentales a través de una revisión de la literatura en sistemas de recomendación, y modelos de representación y recuperación de información semánticos, identifica limitaciones actuales de los sistemas de recomendación, y describe aproximaciones recientes para afrontar algunas de esas limitaciones usando tecnologías basadas en semántica. La segunda parte contiene descripciones y evaluaciones de los modelos de recomendación semánticos propuestos en esta tesis. Finalmente, la tercera y última parte presenta la implementación y evaluación empírica de las propuestas anteriores en un prototipo de sistema de recomendación, explica las características novedosas y ventajas del sistema, y concluye con una discusión general y futuras líneas de investigación.

Adicionalmente, los contenidos de la tesis han sido distribuidos en capítulos de la siguiente manera:

Parte I. Contexto y trabajo relacionado

- El **Capítulo 2** ofrece una visión general del estado del arte en sistemas de recomendación distinguiendo entre aproximaciones basadas en contenido, de filtrado colaborativo e híbridas. Para cada una de ellas se describen sus fortalezas y debilidades, y se presentan varias aplicaciones representativas.
- El **Capítulo 3** motiva y define el uso de tecnologías semánticas en modelos de representación de conocimiento y recuperación de información. De las estrategias existentes el capítulo se centra en aquellas más relacionadas con los sistemas de recomendación. En concreto, describe trabajos relevantes en búsqueda semántica, y recuperación de contenidos personalizada basada en ontologías.

Parte II. Modelos de recomendación: una propuesta basada en ontologías

- El **Capítulo 4** introduce las representaciones de conocimiento y perfil de usuario basadas en ontologías que son subyacentes a las propuestas de esta tesis. A partir de estas representaciones en el capítulo se describe un modelo de recomendación basado en contenido, y extensiones de este modelo para ofrecer recomendaciones contextualizadas y orientadas a grupos.
- El **Capítulo 5** explica un mecanismo por el cual las representaciones de conocimiento y perfil de usuario descritas en el capítulo anterior son usadas para construir comunidades de interés semánticas multi-capas. Las relaciones sociales que emergen en estas comunidades son explotadas para ofrecer recomendaciones, motivando los modelos de recomendación híbridos que se detallan al final del capítulo.

- El **Capítulo 6** expone los experimentos llevados a cabo para evaluar los modelos de recomendación basados en contenido y colaborativos propuestos en el capítulo anterior, y da algunas conclusiones parciales.

Parte III. Evaluaciones adicionales: una experiencia de integración

- El **Capítulo 7** describe la implementación de los modelos de recomendación propuestos en una plataforma de evaluación web. La arquitectura y la interfaz gráfica de usuario del sistema también se presentan en el capítulo.
- El **Capítulo 8** expone las evaluaciones empíricas realizadas con el sistema de recomendación implementado, mostrando los beneficios de las aproximaciones basadas en ontologías.
- El **Capítulo 9** concluye la tesis con discusiones generales y posibles líneas de investigación a ser estudiadas mediante posteriores adaptaciones y extensiones del sistema prototipo.

Cada uno de los capítulos anteriores comienza con una breve introducción a los temas que trata, y un párrafo describiendo su estructura interna. Los capítulos que presentan resultados experimentales acaban con las correspondientes conclusiones parciales. El resto de capítulos, sin embargo, concluyen con secciones de resumen.

Además de los capítulos se incluyen varios apéndices con información adicional que es relevante aunque no central para los propósitos de la tesis:

- El **Apéndice A** lista todos los acrónimos usados en este documento.
- El **Apéndice B** proporciona la *API* del sistema prototipo desarrollado.
- El **Apéndice C** contiene la traducción a español del capítulo *Introduction*.
- El **Apéndice D** contiene la traducción a español del capítulo *Conclusions*.

C.5 Publicaciones

La base de la que surgen las propuestas de esta tesis es el modelo de representación de conocimiento basado en ontologías introducido en (Vallet, Fernández, & Castells, 2005). Este modelo ha sido explotado en diferentes aplicaciones como la búsqueda semántica (Castells, Fernández, & Vallet, 2007) y la recuperación de contenidos personalizada y contextualizada (Vallet, Castells, Fernández, Mylonas, & Avrithis, 2007). Como extensiones novedosas a estos trabajos, las publicaciones que han dado lugar a esta tesis son clasificadas en esta sección atendiendo al capítulo y tema de investigación con las que están relacionadas.

Capítulo 4

Recuperación de contenidos personalizada y contextualizada

La representación de conocimiento basada en ontologías y el marco de recuperación de contenidos personalizada y contextualizada del capítulo han sido usados para la generación de resúmenes personalizados de diferentes fuentes de contenidos multimedia. Una descripción de esta aplicación viene dada en:

- Dolbear, C., Hobson, P., Vallet, D., Fernández, M., Cantador, I., & Castells, P. (2007). Personalised Multimedia Summaries. *Book chapter in "Semantic Multimedia and Ontologies: Theory and Applications"*, pp. 165-183. Springer-Verlag. Edited by Y. Kompatsiaris, and P. Hobson. ISBN: 978-1-84800-075-9.

En este trabajo la explotación de la técnica de contextualización sugerida ofrece de manera consistente mejores resultados que los dados por la personalización simple. Los experimentos descritos muestran que la contextualización mejora la personalización eliminando los intereses de usuario que están fuera de contexto, y mantiene aquellos que realmente son relevantes para el resumen en curso.

Una segunda aplicación de los modelos de recomendación personalizada y contextualizada en la adaptación automática en entornos de recuperación de contenidos multimedia se presenta en:

- Cantador, I., López, F., Bescós, J., Castells, P., & Martínez, J. M. (2008). Enhanced Descriptions for Personalized Retrieval and Automatic Adaptation of Audiovisual Content Retrieval. *Book chapter in "Personalization of Interactive Multimedia Services: A Research and Development Perspective"*. Nova Science Publishers. Edited by J. J. Pazos-Arias, C. Delgado, and M. López. ISBN: 978-1-60456-680-2.

Este trabajo se centra en un conjunto de iniciativas y logros obtenidos en el ajuste automático de contenidos multimedia atendiendo a una amplia variedad de infraestructuras. La visión de adaptación multimedia propuesta comprende métodos de adaptación a bajo y alto nivel que abarcan desde la ordenación de unidades de contenido de acuerdo a intereses de usuario en diferentes escenarios (e.g., presencia o ausencia de una consulta explícita del usuario, existencia de uno o múltiples usuarios), hasta técnicas de adaptación a diferentes entornos de uso (terminales, redes, *codecs*, reproductores, preferencias de usuario, etc.).

Perfiles de grupo para recuperación de contenidos

Adicionalmente a las aplicaciones anteriores, la representación de perfil de usuario basada en ontologías ha sido adaptada para el diseño de novedosas estrategias de modelado de perfiles de grupo. Una descripción y evaluación de la propuesta se puede encontrar en:

- Cantador, I., Castells, P., & Vallet, D. (2006). Enriching Group Profiles with Ontologies for Knowledge-Driven Collaborative Content Retrieval. *Proceedings of the 1st International Workshop on Semantic Technologies in Collaborative Applications (STICA 2006), at the 15th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 2006)* (pp. 358-363). Manchester, UK: IEEE Computer Society Press, ISBN 0-7695-2623-3.

En este artículo, asumiendo que se dispone de perfiles semánticos asociados a usuarios con gustos e intereses compartidos, se estudia la factibilidad de aplicar estrategias basadas en teorías de decisión social (Masthoff, 2004) para la combinación de múltiples preferencias individuales en un sistema de recuperación de contenidos multimedia personalizada. Combinando varios perfiles con las estrategias de modelado de grupo consideradas, se busca establecer la manera en la que las personas recomiendan una ordenación óptima de elementos a un grupo, y miden la satisfacción de una ordenación de elementos dada. Los experimentos desarrollados demuestran los beneficios de usar preferencias semánticas y exhiben qué estrategias de combinación de perfiles podrían ser apropiadas en un entorno colaborativo.

Capítulo 5

Redes sociales y comunidades de interés

Una vez se estudiaron estrategias de modelado de grupos, el siguiente paso en la investigación fue el diseño de un algoritmo de agrupamiento (*clustering*) que encontrase aquellos conjuntos de perfiles con características similares:

- Cantador, I., & Castells, P. (2006). Building Emergent Social Networks and Group Profiles by Semantic User Preference Clustering. *Proceedings of the 2nd International Workshop on Semantic Network Analysis (SNA 2006), at the 3rd European Semantic Web Conference (ESWC 2006)*, (pp. 40-53). Budva, Montenegro.

El algoritmo propuesto está basado en la representación ontológica del dominio en el que se definen los intereses de los usuarios. El espacio ontológico toma la forma de una red de conceptos interconectados. Tomando ventaja de las relaciones existentes entre conceptos, y de las preferencias ponderadas de los usuarios por esos conceptos, se agrupa el espacio semántico obteniendo conjuntos de conceptos que representan temas de interés comunes. A continuación, se segmentan los perfiles de usuario proyectando los grupos de conceptos obtenidos sobre las preferencias de cada usuario. Los perfiles particionados son finalmente usados para comparar las preferencias individuales a diferentes niveles semánticos, y encontrar varias comunidades de usuarios compartiendo intereses.

Recomendación híbrida basada en multi-capas semánticas

De acuerdo a los diferentes subconjuntos de preferencias obtenidos con el algoritmo de *clustering* propuesto, los usuarios pueden ser comparados de tal manera que varios, en vez de uno sólo, enlaces (ponderados) son establecidos entre dos individuos. Estas relaciones sociales “multi-capa” fueron usadas para modelar una serie de técnicas de recomendación híbridas en:

- Cantador, I., & Castells, P. (2006). Multi-Layered Ontology-based User Profiles and Semantic Social Networks for Recommender Systems. *Proceedings of the 2nd International Workshop on Web Personalisation, Recommender Systems and Intelligent User Interfaces (WPRSIUI 2006), at the 4th International Conference on Adaptive Hypermedia (AH 2006)*. Dublin, Ireland.

Una discusión más detallada de los modelos anteriores, junto con experimentos más relevantes con usuarios, se proporciona en el siguiente trabajo:

- Cantador, I., & Castells, P. (2006). Multi-Layered Semantic Social Networks Modelling by Ontology-based User Profiles Clustering: Application to Collaborative Filtering. *Proceedings of the 15th International Conference on Knowledge Engineering and Knowledge Management – Managing Knowledge in a World of Networks (EKAW 2006)* (pp. 334-349). Podebrady, Czech Republic: Lectures Notes in Artificial Intelligence, 4248. Springer-Verlag, ISBN 3-540-46363-1.

Capítulo 6

Evaluación de los modelos de recomendación

Continuando los trabajos anteriores, evaluaciones adicionales de los modelos híbridos se exponen en:

- Cantador, I., Castells, P., & Bellogín, A. (2007). Modelling Ontology-based Multilayered Communities of Interest for Hybrid Recommendations. *Proceedings of the 1st International Workshop on Adaptation and Personalisation in Social Systems: Groups, Teams, Communities (SociUM 2007), at the 11th International Conference on User Modelling (UM 2007)*. Corfu, Greece.

En este caso, en vez de probar los modelos con un número bastante reducido de perfiles de usuario definidos manualmente, se generaron automáticamente cientos de perfiles combinando información de los repositorios MovieLens³⁴ e IMDb³⁵. Específicamente, se transformaron los *ratings* públicos de MovieLens en preferencias semánticas sobre características de películas en IMDb. Con los perfiles obtenidos se evaluaron los modelos de recomendación mostrando de nuevo su factibilidad.

³⁴ MovieLens repository, GroupLens Research, <http://www.grouplens.org/>

³⁵ Internet Movie Database, IMDb, <http://www.imdb.com/>

Todas las aproximaciones de recomendación contextualizada y orientada a grupos presentadas se reunieron en el siguiente artículo:

- Vallet, D., Cantador, I., Fernández, M., & Castells, P. (2006). A Multi-Purpose Ontology-based Approach for Personalized Content Filtering and Retrieval. *Proceedings of the 1st International Workshop on Semantic Media Adaptation and Personalisation (SMAP 2006)* (pp 19-24). Athens, Greece.

Este trabajo recibió una invitación a ser extendido y publicado como capítulo de libro:

- Cantador, I., Fernández, M., Vallet, D., Castells, P., Picault, J., & Ribière, M. (2007). A Multi-Purpose Ontology-based Approach for Personalised Content Filtering and Retrieval. *Book chapter in "Studies in Computational Intelligence"*, vol. 93, pp. 25-51. Springer-Verlag. Edited by M. Wallace, M. Angelides, and P. Mylonas. ISBN: 978-3-540-76359-8.

Finalmente, la aplicación de comunidades de interés multi-capa a modelado de grupos y sistemas de recomendación híbridos ha sido aceptada como dos artículos de revista:

- Cantador, I., & Castells, P. (2008). Extracting Multilayered Semantic Communities of Interest from Ontology-based User Profiles: Application to Group Modelling and Hybrid Recommendations. *Computers in Human Behaviour, special issue on Advances of Knowledge Management and Semantic Web for Social Networks*. Elsevier. In press.
- Cantador, I., Bellogín, A., & Castells, P. (2008). A Multilayer Ontology-based Hybrid Recommendation Model. *AI Communications, special issue on Recommender Systems*. IOS Press. In press.

Capítulo 7

Implementación de un sistema de recomendación basado en ontologías

A partir de la evaluación de los modelos de recomendación de forma aislada, se identificó la necesidad de integrar todos ellos en un sistema de recomendación con el fin de hacerlo público a la comunidad científica y permitir llevar a cabo experimentos más sofisticados y realistas. La presentación de tal sistema aparece en:

- Cantador, I., Bellogín, A., Castells, P. (2008). News@hand: A Semantic Web Approach to Recommending News. *Proceedings of the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2008)*. Hannover, Germany. Lecture Notes in Computer Science, vol. 5149, pp. 279-283. Springer-Verlag. ISBN 978-3-540-70984-8.

News@hand es un sistema de recomendación de noticias que aplica las propuestas de representación de conocimiento y técnicas de recomendación basadas en ontologías para describir y relacionar contenidos de noticias y preferencias de usuario, con el fin de producir sugerencias de noticias de forma personalizada.

Durante el desarrollo del sistema varios retos científicos surgieron: el poblado (instanciación) de las ontologías de dominio, la anotación semántica automática de ítems, y la obtención de preferencias de usuario a partir de etiquetas (del inglés *tags*) sociales. Las propuestas para abordar estos problemas se introducen en:

- Cantador, I., Szomszor, M., Alani, H., Fernández, M., & Castells, P. (2008) Enriching Ontological User Profiles with Tagging History for Multi-Domain Recommendations. *Proceedings of the 1st International Workshop on Collective Semantics: Collective Intelligence and the Semantic Web (CISWeb 2008), at the 5th European Semantic Web Conference (ESWC 2008)*. Tenerife, Spain. CEUR Workshop Proceedings, vol. 351, pp. 5-19, ISSN 1613-0073.

Este trabajo presenta una estrategia novedosa que filtra información colaborativa de etiquetas (i.e., “folcsonomías”) para incorporarla en una representación de conocimiento ontológica. Para alcanzar tal objetivo, se propone explotar información semántica disponible en recursos externos como WordNet (Miller, 1995) y Wikipedia³⁶. Evaluaciones preliminares de las técnicas propuestas también se explican en el artículo.

Capítulo 8

Evaluaciones con el sistema de recomendación implementado

Finalmente, experimentos con el sistema *News@hand* para evaluar la combinación de los modelos de recomendación personalizados se describen en:

- Cantador, I., Bellogín, A., Castells, P. (2008). Ontology-based Personalised and Context-aware Recommendations of News Items. *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2008)*. Sydney, Australia.

La combinación de un modelo que personaliza el orden en el que se muestran noticias atendiendo al perfil con intereses a largo plazo del usuario, y otro modelo que reordena las listas de noticias de acuerdo al contexto semántico de intereses actuales (a corto plazo) del usuario, mostró mejoras significativas en las pruebas experimentales realizadas.

³⁶ Wikipedia, the free encyclopaedia, <http://www.wikipedia.org/>

Contribuciones relacionadas

En paralelo a las publicaciones originadas por esta tesis ha habido contribuciones adicionales en aspectos relacionados con los sistemas de recomendación. En concreto, se han investigado: 1) mecanismos novedosos de recomendación multi-criterio, 2) estrategias de modelado de usuario a partir de fuentes de información folcsonómica cruzadas, y 3) técnicas de análisis de preferencias de usuario relevantes en un sistema de recomendación usando algoritmos de aprendizaje automático. La primera propuesta ha sido integrada en el sistema *News@band* descrito en el Capítulo 8, la segunda es una extensión del mecanismo de construcción de preferencias de usuario semánticas explicado en la sección 8.3.2, y la tercera ha sido realizada con información de registros de actividad recogidos en los experimentos llevados a cabo con *News@band* y que se describen en la sección 8.4.4.

Evaluación colaborativa y recomendaciones multi-criterio

La implementación de una herramienta para la evaluación y reutilización colaborativa de ontologías fue presentada en:

- Fernández, M., Cantador, I., & Castells, P. (2006). CORE: A Tool for Collaborative Ontology Reuse and Evaluation. *Proceedings of the 4th International Workshop on Evaluation of Ontologies for the Web (EON 2006), at the 15th International World Wide Web Conference (WWW 2006)*. Edinburgh, UK. CEUR Workshop Proceedings, vol. 179, ISSN 1613-0073.

Entre otras funcionalidades novedosas, esta herramienta proporciona un mecanismo de recomendación colaborativo basado en *ratings* multi-criterio. Debido a su relevancia en la comunidad de sistemas de recomendación, el algoritmo se explicó en detalle en otra publicación:

- Cantador, I., Fernández, M., & Castells, P. (2006). A Collaborative Recommendation Framework for Ontology Evaluation and Reuse. *Proceedings of the International Workshop on Recommender Systems, at the 17th European Conference on Artificial Intelligence (ECAI 2006)*, (pp. 67-71). Riva del Garda, Italy.

El marco de recomendación fue diseñado para afrontar el reto de evaluar aquellas características de una ontología que dependen de valoraciones humanas subjetivas y que por naturaleza son más difíciles de tratar por una máquina. Haciendo uso de técnicas de filtrado colaborativo, el sistema explota los *ratings* proporcionados por los usuarios para recomendar las ontologías más adecuadas a un dominio dado.

El sistema fue transformado en una aplicación web y modificado para incorporar nuevas capacidades colaborativas durante la definición del dominio del problema, y la ejecución de los procesos de recomendación:

- Cantador, I., Fernández, M., & Castells, P. (2007). Improving Ontology Recommendation and Reuse in WebCORE by Collaborative Assessments. *Proceedings of the 1st International Workshop on Social and Collaborative Construction of Structured Knowledge (CKC 2007), at the 16th International World Wide Web Conference (WWW 2007)*. Banff, Canada. CEUR Workshop Proceedings, vol. 273, ISSN 1613-0073.

En este artículo el algoritmo de recomendación multi-criterio es evaluado empíricamente, mostrando beneficios relevantes para la aplicación.

Modelado de usuario a partir de información folksonómica

Se propuso un método para la consolidación automática de perfiles de usuario cruzados de varias aplicaciones de redes sociales, y el posterior modelado semántico de intereses de usuario usando Wikipedia como modelo multi-dominio:

- Szomszor, M., Cantador, I., Alani, H. (2008). Correlating User Profiles from Multiple Folksonomies. *Proceedings of the 19th ACM Conference on Hypertext and Hypermedia (Hypertext 2008)*. Pittsburgh, Pennsylvania, USA. ACM 2008. ISBN 978-1-59593-985-2.
- Szomszor, M., Alani, H., Cantador, I., O'Hara, K., Shadbolt, N. (2008). Semantic Modelling of User Interests based on Cross-Folksonomy Analysis. *Proceedings of the 7th International Semantic Web Conference (ISWC 2008)*. Karlsruhe, Germany. Lecture Notes in Computer Science. Springer-Verlag.

En estos artículos se evalúa cuánto puede aprenderse sobre las preferencias del usuario a partir de la combinación de sus perfiles basados en *tags* definidos en diferentes sitios sociales, y en qué dominios se centran esas preferencias. Los resultados muestran que se pueden obtener perfiles enriquecidos cuando se combinan varios conjuntos de *tags*.

Análisis de preferencias relevantes en sistemas de recomendación

Adicionalmente a la propuesta de técnicas que proporcionan recomendaciones de ítems a partir de información de preferencias de usuario, o a la definición de estrategias que aprendan éstas últimas, también se investigó un mecanismo para descubrir qué preferencias son realmente relevantes para obtener recomendaciones precisas.

- Bellogín, A., Cantador, I., Castells, P., Ortigosa, A. (2008). Discovering Relevant Preferences in a Personalised Recommender System using Machine Learning Techniques. *Proceedings of the Preference Learning Workshop (PL 2008), at the 8th European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2008)*. Antwerp, Belgium.

En este trabajo se presenta una metodología de meta-evaluación que aplica técnicas de aprendizaje automático para analizar registros de actividad de *News@hand* con el fin de descubrir (y priorizar) las preferencias de usuario y parámetros del sistema que son adecuados para recomendaciones acertadas. Además, también muestra cómo la metodología propuesta puede ser usada para validar el propio proceso de evaluación del sistema.

Appendix D

Conclusiones

Con el fin de abordar limitaciones existentes en sistemas de recomendación actuales, esta tesis supone una propuesta ambiciosa para incorporar y explotar un espacio conceptual que describa y conecte de forma genérica descripciones de las preferencias de usuario y descripciones de los contenidos de los ítems a recomendar. Para ello se plantearon una serie de objetivos concretos:

- La definición de una representación del conocimiento formal (basada en ontologías) que permita expresar relaciones semánticas entre conceptos.
- La creación de modelos basados en contenido flexibles que permitan la contextualización y extensión a múltiples usuarios de las recomendaciones.
- La creación de modelos híbridos que permitan añadir a los modelos basados en contenido las ventajas del filtrado colaborativo.
- La implementación de un sistema de recomendación que permita la evaluación de todas las propuestas anteriores de forma conjunta.

En la primera parte de la tesis se revisaron y relacionaron las dos áreas de investigación en las que se enmarca este trabajo: los sistemas de recomendación, y la representación y recuperación de información semánticas. En la segunda parte de la tesis se presentaron las propuestas de representación del conocimiento y de recomendación, y se expusieron experimentos llevados a cabo para evaluarlos de forma independiente en escenarios controlados con pocos usuarios o con conjuntos de datos artificiales. Finalmente, en la tercera parte de la tesis se describió el sistema de recomendación implementado, que no sólo se utilizó para realizar evaluaciones más realistas de los modelos, sino también para poner de manifiesto las dificultades que conlleva la implantación de una aplicación basada en semántica.

En este capítulo se describen las conclusiones y contribuciones alcanzadas con el trabajo realizado (en la Sección D.1), y se plantean limitaciones de las propuestas, así como posibles líneas de investigación futura que las aborden (en la Sección D.2).

D.1 Resumen y contribuciones

El resultado final de esta tesis es un conjunto de modelos de recomendación que relacionan gustos e intereses de usuarios por ítems de diversa índole a través de una representación de conocimiento basada en ontologías. Las relaciones semánticas definidas en las ontologías del sistema son empleadas por varias estrategias de recomendación novedosas que están dirigidas a uno o varios usuarios, que tienen en cuenta el contexto semántico actual de la recuperación de contenidos, y que, a diferentes niveles de gustos e intereses de usuario compartidos, descubren y explotan relaciones colaborativas basadas en contenido entre las preferencias de los usuarios.

En los siguientes apartados se motivan y resumen las propuestas anteriores, y se detallan las contribuciones alcanzadas, destacando los beneficios que aportan en comparación a otras aproximaciones existentes en la literatura.

D.1.1 Representación del conocimiento ontológica

Los sistemas de recomendación basados en contenido (Lang, 1995; Pazzani & Billsus, 1997; Krulwich & Burkey, 1997; Mooney, Bennett, & Roy, 1998; Billsus & Pazzani, 1999) emplean en general vectores de términos (palabras clave) para describir las preferencias de usuario y los contenidos de los ítems. A través de técnicas de anotación e indexado (e.g., TF-IDF), y técnicas de recuperación de información clásicas (Salton & McGill, 1986; Baeza-Yates & Ribeiro Neto, 1999), como por ejemplo el modelo vectorial o el modelo probabilístico, estos sistemas calculan similitudes entre cada vector de usuario y cada vector de ítem para proporcionar una medida del potencial interés que ese usuario tiene por el ítem.

Esta propuesta de representación responde al requerimiento de ser procesable eficientemente por un sistema, pero implica la *pérdida de información* debido principalmente a dos motivos. El primer motivo está relacionado con la no desambiguación de los términos. Un término puede tener varios significados, y el usuario puede que sólo esté interesado por uno de ellos. Sin tener en cuenta el significado del término en cada caso todos los ítems que incluyan dicho término podrían ser recomendados al usuario, pero sólo algunos, aquellos que tengan el término con el significado preferido por el usuario, van a ser relevantes. El resto producirían recomendaciones erróneas, no útiles para el usuario. El segundo motivo es la suposición de independencia entre términos. El hecho de que la descripción de un ítem no tenga explícitamente términos de interés para el usuario no implica necesariamente que ese ítem no le sea relevante. Otros términos relacionados semánticamente (mediante sinonimia, antonimia, hiperonimia, hiponimia, y otras relaciones) podrían ser identificados y utilizados para determinar la importancia del ítem para el usuario.

Las limitaciones anteriores hacen que en muchos de los sistemas de recomendación actuales exista:

Falta de entendimiento y explotación de la semántica subyacente a los gustos e intereses de los usuarios y a los contenidos de los ítems recomendados

Para abordar este problema se ha propuesto una representación del conocimiento en la que tanto los perfiles de los usuarios como los contenidos de los ítems vienen descritos por vectores de conceptos (clases e instancias) ponderados que pertenecen a una o varias ontologías de dominio. En el vector asociado a un perfil de usuario, cada componente tiene asignado un peso, midiendo el interés (positivo o negativo) que el concepto correspondiente suscita al usuario. En el vector de anotaciones de un ítem, el peso de cada componente mide el grado en el que el concepto es relevante (informativo) dentro del contenido del ítem y/o en relación a los contenidos del resto de ítems.

La contribución que la tesis supone en este ámbito es:

La definición de una representación del conocimiento formal, acerca de preferencias de usuario y contenidos de ítems, que no es ambigua y que tiene en cuenta relaciones semánticas arbitrarias (i.e., no pre-establecidas) entre conceptos.

El uso de esta representación conceptual, en comparación con aproximaciones comunes basadas en palabras claves o en (relaciones explícitas entre) ítems, aporta los siguientes beneficios:

- *Riqueza semántica.* Las preferencias y anotaciones son más precisas, y reducen el efecto de ambigüedad. Esto permite el mejor entendimiento y explotación de la semántica involucrada en los procesos de recuperación de información personalizada y recomendación.
- *Representación jerárquica.* Los conceptos ontológicos están representados de forma jerárquica, a través de relaciones estándar como “sub-clase de” o “instancia de”. Ascendentes y descendientes de un concepto dado pueden proporcionar información adicional valiosa sobre la semántica de este último.
- *Inferencia.* Los lenguajes de descripción de ontología estándar, como RDF u OWL, soportan mecanismos de inferencia para el descubrimiento de conocimiento que puede ser usado para mejorar las recomendaciones.

Además de los beneficios característicos a una representación basada en ontologías, la propuesta aporta las siguientes ventajas no ofrecidas por los modelos de recomendación clásicos:

- *Portabilidad.* A través de estándares basados en XML, el conocimiento de dominio, las anotaciones de los ítems, e incluso las preferencias de los usuarios pueden ser fácilmente distribuidas, adaptadas o integradas en diferentes sistemas de recomendación para diferentes aplicaciones.
- *Independencia de dominio.* Independientemente del dominio en el que se usen, las estructuras de conocimiento para perfiles de usuario e ítem consisten en redes semánticas de conceptos interconectados. Los modelos de recomendación se construyen de forma genérica en base a las estructuras anteriores, sin tener que considerar restricción de dominio alguna.
- *Anotación de múltiples fuentes.* Asumiendo la existencia de mecanismos de anotación semántica manual o automática, los modelos de recomendación que empleen la representación de conocimiento propuesta pueden ser empleados para sugerir ítems de muy diversa naturaleza (texto, imagen, video, audio, etc.).

Representaciones clásicas del perfil de usuario a través de listas de palabras clave o evaluaciones numéricas (*ratings*) de ítems son propensas a la **“escasez” de preferencias**. En sistemas donde las preferencias son establecidas manualmente los usuarios no suelen emplear mucho tiempo en la creación de su perfil, y en sistemas donde las preferencias son determinadas de forma automática a partir de históricos de acciones los algoritmos de aprendizaje tienden a reconocer intereses del usuario muy genéricos. Este hecho conlleva dos problemas principales. El primer problema está relacionado con la poca densidad (del inglés *sparsity*) de información en las estructuras empleadas por los modelos de recomendación, que complica el encontrar similitudes o correlaciones entre usuarios e ítems (Billsus & Pazzani, 1998; Sarwar, Karypis, Konstan, & Riedl, 2000). El segundo problema es la dificultad de recomendar ítems a un nuevo usuario que comienza a usar el sistema y que tiene ninguna o pocas preferencias declaradas (Schein, Popescul, & Ungar, 2001). Aparte de estrategias que incentiven a los usuarios para crear sus perfiles, los dos problemas anteriores podrían ser abordados con técnicas que extiendan o enriquezcan los perfiles de usuario. De este modo, se plantea la:

Necesidad de enriquecer los perfiles de usuario e ítem

Para satisfacer esta necesidad se ha propuesto una estrategia que propaga los pesos de los conceptos ontológicos de los perfiles de usuario e ítem hacia otros conceptos enlazados a través de relaciones semánticas existentes en las ontologías de dominio. La propagación está basada en técnicas de CSA (Cohen & Kjeldsen, 1987; Crestani, 1997), considerando la atenuación de los pesos a medida que la expansión semántica avanza, tratando bucles en los caminos de propagación realizados, y permitiendo acotar el alcance de la extensión.

La contribución que la tesis aporta en el campo es:

El diseño de un mecanismo novedoso que extiende las descripciones semánticas de preferencias de usuario y contenidos de ítems a través de relaciones ontológicas de sus conceptos.

Los beneficios principales de la propuesta son:

- *Mitigación del problema de poca densidad de preferencias.* A través de la expansión semántica, los perfiles de usuario e ítem son más grandes, cubriendo más áreas del espacio conceptual, y por ello la probabilidad de encontrar similitudes y correlaciones entre usuarios e ítems a la hora de hacer recomendaciones es también mayor.
- *Apoyo al tratamiento del problema del arranque frío.* La expansión semántica de nuevos perfiles de usuario e ítem facilita su incorporación y mejor explotación en los procesos de recomendación. También podría ser usada como técnica de sugerencia de preferencias en los procesos de creación y edición de perfiles de usuario.

D.1.2 Recomendaciones semánticas basadas en contenido

Los sistemas de recomendación actuales son susceptibles de ser mejorados con extensiones de sus capacidades (Adomavicius & Tuzhilin, 2005). Una de las más representativas es el empleo de **recomendaciones contextualizadas** (Räck, Arbanowski, & Steglich, 2006; Anand & Mobasher, 2007; Vallet, Castells, Fernández, Mylonas, & Avrithis, 2007). El contexto puede ser definido de muchas y muy diversas formas:

- Atendiendo a hechos directamente relacionados con el sistema, como por ejemplo, las últimas acciones y evaluaciones realizadas por el usuario, la fecha y hora actuales, etc.
- En función de información procedente de otras aplicaciones, como por ejemplo, los eventos planificados en una agenda electrónica, los sitios web incluidos como favoritos en un navegador web, etc.
- A partir de factores externos, como por ejemplo la localización, la compañía o el estado de ánimo actuales del usuario.
- Otras.

En cualquier caso, la adición de contexto en los procesos de recomendación es una tarea compleja, que en muchas ocasiones se debe a la falta de flexibilidad en los modelos de recuperación de contenidos empleados.

Otra posible extensión muy importante de los sistemas de recomendación es el llevar a cabo **recomendaciones orientadas a grupo**. La sugerencia de ítems a un grupo de personas es un requerimiento que ha sido identificado en múltiples aplicaciones, como por ejemplo, la recomendación colectiva de composiciones musicales (McCarthy & Anagnost, 1998), películas (O'Connor, Cosley, Konstan, & Riedl, 2001), atracciones turísticas (Ardissono, Goy, Petrone, Segnan, & Torasso, 2003) o programas de televisión (Ali & Van Stam, 2004). De nuevo, los modelos tradicionales no son lo suficientemente flexibles para adoptar este tipo de recomendación, y en su lugar han de proponerse estrategias ad-hoc muy dependientes del dominio de aplicación.

Existen otras extensiones y mejoras posibles (ver Sección D.2), que en su mayoría, y al igual que las dos arriba explicadas, son originas por la:

**Necesidad de extender los modelos de recomendación personalizados
para proporcionar sugerencias de ítems contextualizadas
y orientadas a un grupo de usuarios**

A partir de la representación de perfiles de usuario e ítem basada en ontologías se ha propuesto un modelo de recomendación personalizado que es una adaptación del modelo de recuperación de información vectorial. En esta propuesta el interés de un usuario por un ítem se calcula mediante el coseno del ángulo formado por los respectivos vectores de conceptos, una vez han sido extendidos mediante la técnica de expansión semántica citada anteriormente.

De forma análoga, se ha definido la noción de contexto semántico como el conjunto de conceptos ontológicos presentes en las anotaciones de aquellos ítems recientemente visitados o evaluados por el usuario. La representación del contexto es de nuevo vectorial, por lo que es fácilmente combinable con el modelo de personalización básico. En esta tesis se ha estudiado la combinación lineal de ambos, pero otras alternativas podrían ser factibles.

La representación vectorial no sólo permite la combinación de un perfil de usuario y el contexto semántico, sino también la fusión de múltiples perfiles con el fin de generar un perfil único que tenga en cuenta de alguna manera las preferencias de un grupo de usuarios. Este perfil de grupo puede posteriormente ser utilizado por el modelo de recomendación básico. El desarrollo de una estrategia eficaz con la que combinar los perfiles de un grupo ha sido investigado en este trabajo, y se ha demostrado la factibilidad de aplicar ciertas técnicas extraídas de la teoría de elección social (Masthoff, 2004).

La contribución realizada en cuanto a flexibilidad de sistemas de recomendación se refiere se puede entonces resumir como sigue:

La creación de un modelo de recomendación personalizada basado en ontologías que permite la incorporación contexto semántico, y que puede adaptarse a las preferencias de uno o más usuarios.

El principal beneficio que aporta el modelo de recomendación personalizada propuesto es el de ser flexible para adaptarse a:

- *Recomendaciones contextualizadas.* El hecho de añadir contexto semántico en el proceso de recomendación personalizada permite la “focalización” de las preferencias de usuario. En ocasiones no todas las preferencias del perfil de usuario están relacionadas con el objetivo actual de búsqueda o recomendación, y sólo aquellas preferencias que están dentro del contexto presente deben ser consideradas.
- *Recomendaciones orientadas a grupo.* Las estrategias de modelado de grupos propuestas, aparte de ser muy sencillas de ejecutar, e ir más allá de la simple agregación de preferencias (al emplear técnicas basadas en la teoría de elección social), permiten su aplicación genérica en cualquier dominio, siempre que por supuesto se mantenga la representación del conocimiento ontológica expuesta.

D.1.3 Recomendaciones semánticas híbridas

Un sistema de recomendación basado en contenido sugiere ítems a un usuario atendiendo únicamente a las preferencias definidas en su perfil. Este tipo de recomendaciones, aún siendo preciso, puede ser contraproducente en determinadas circunstancias. En general, estas estrategias conllevan la **sobre-especialización** de los ítems recomendados, que comparten las mismas características de contenido. Como consecuencia, pueden tender a una **falta de diversidad y novedad**, indeseada y valorada negativamente por el usuario.

Estos problemas son solventados por estrategias de filtrado colaborativo que recomiendan ítems al usuario en base a evaluaciones de otras personas con las que comparte ciertas preferencias (Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994; Hill, Stead, Rosenstein, & Furnas, 1995; Shardanand & Maes, 1995; Konstan, Miller, Maltz, Herlocker, Gordon, & Riedl, 1997; Pennock, Horvitz, Lawrence, & Giles, 2000). De este modo, el usuario recibe sugerencias de ítems cuyos contenidos no están directamente relacionados con su perfil, sino con los perfiles de usuarios afines. La eficacia de estas estrategias queda avalada por su éxito en aplicaciones comerciales reales, como por ejemplo *Amazon.com* (Linden, Smith, & York, 2003), pero todavía muestran ciertas limitaciones. Una de ellas es la dificultad de recomendar ítems a **usuarios con preferencias poco usuales** (conocidos en la literatura como “ovejas negras”; en inglés “ovejas grises”, *grey sheep*). Para establecer la similitud entre usuarios

se han propuesto diferentes medidas (Adomavicius & Tuzhilin, 2005). Sin embargo, en general, todas ellas se basan en comparaciones globales de los perfiles. En esta tesis se aboga por segmentar los perfiles a partir de grupos de preferencias compartidas entre los usuarios, y establecer similitudes a partir de cada uno de los segmentos obtenidos. De este modo, coincidencias de preferencias poco usuales se verían reforzadas al tratar con perfiles más pequeños y focalizados en ámbitos de gustos e intereses específicos.

Resumiendo, en sistemas de recomendación colaborativos existe una:

Dificultad de recomendar ítems a usuarios con preferencias poco usuales, o a usuarios que comparten intereses sólo en determinados ámbitos semánticos

Según lo anterior, en entornos de recomendación subyace una necesidad de distinguir diferentes niveles o capas dentro de los perfiles de los usuarios. Dependiendo del contexto actual, sólo un subconjunto específico de las preferencias de un usuario debería ser considerado para establecer sus similitudes con otras personas cuando se tienen que hacer recomendaciones.

Para satisfacer la necesidad anterior este trabajo presenta una estrategia que parte de la representación del conocimiento ontológica propuesta. Tomando ventaja de las relaciones semánticas entre conceptos, y de las preferencias (ponderadas) de los usuarios por tales conceptos, la estrategia agrupa el espacio semántico en función de correlaciones entre conceptos existentes en los perfiles de usuario. De este modo, los grupos de conceptos creados pueden ser entendidos como conjuntos de preferencias compartidas por varios usuarios. Proyectando esos grupos de conceptos sobre los perfiles de usuario, éstos son divididos en varios segmentos. Atendiendo a estos segmentos (o sub-perfiles) los usuarios son comparados a diferentes niveles, permitiendo encontrar más de una relación (ponderada) entre dos usuarios cualesquiera. Las relaciones entre usuarios en los diversos niveles o capas semánticas constituyen diferentes comunidades de interés, y pueden ser empleadas para proporcionar recomendaciones en áreas conceptuales más focalizadas o especializadas, incluso cuando los perfiles de usuario completos son muy diferentes.

A partir de las comunidades de interés semánticas multi-capa una contribución adicional de este trabajo es:

La creación de modelos híbridos que combinan los perfiles de usuario de forma colaborativa a diversos niveles semánticos, atendiendo a diferentes grupos de preferencias compartidas.

Los modelos de recomendación híbridos basados en múltiples capas semánticas ofrecen las siguientes ventajas:

- *Disminución del efecto de los problemas de sobre-especialización y falta de diversidad y novedad de contenidos.* Gracias a la combinación colaborativa de perfiles de usuario se evitan problemas derivados de aproximaciones basadas en contenido puras. Un usuario recibe recomendaciones diversas y novedosas que no necesariamente están explícitamente relacionadas con sus preferencias, sino con otras de personas afines.
- *Afrontamiento del efecto de “ovejas negras”.* A través de la contextualización de las recomendaciones en diferentes capas semánticas que atienden a gustos e intereses compartidos entre usuarios se potencian las coincidencias de preferencias poco usuales a la hora de comparar perfiles de usuario.

D.1.4 Evaluación de los modelos de recomendación

A diferencia de otras disciplinas, la evaluación de sistemas de recomendación no es sencilla. En la literatura se han definido métricas que tratan de estimar de forma objetiva la precisión de las recomendaciones (Herlocker, Konstan, Terveen, & Riedl, 2004). La idea principal de estas métricas es promediar la diferencia existente entre evaluaciones reales (proporcionadas por usuarios) y predicciones (proporcionadas por el sistema) para un conjunto de ítems de referencia. Aunque suelen ser usadas como método estándar de comparación de modelos de recomendación, en muchas ocasiones resultan insuficientes, pues no contemplan magnitudes más subjetivas, pero muy importantes, como por ejemplo la novedad, la diversidad o la cobertura (del espacio de ítems) proporcionadas por las recomendaciones (Sarwar, Konstan, Borchers, Herlocker, Miller, & Riedl, 1998; Good, et al., 1999; Herlocker, Konstan, Borchers, & Riedl, 1999; Herlocker, Konstan, & Riedl, 2000; Sarwar, Karypis, Konstan, & Riedl, 2001; Schein, Popescul, & Ungar, 2001).

Usando métricas de precisión en diferentes experimentos, los modelos de recomendación propuestos en la tesis fueron evaluados con usuarios reales y conjuntos de datos artificiales creados a partir de fuentes externas. De forma aislada e independiente cada experimento proporcionó resultados positivos que avalan la factibilidad de las propuestas. Sin embargo, se vio la necesidad de llevar a cabo experimentación adicional en un entorno que integrase los diferentes modelos combinando sus salidas, que no fuese tan controlado y cerrado como el empleado en las evaluaciones aisladas, y que permitiese obtener valoraciones subjetivas de los usuarios. En otras palabras, se consideró necesaria la:

Evaluación de los modelos de representación del conocimiento y de recomendación basados en ontologías en un sistema prototipo

De este modo, como última parte de la tesis, se implementó *News@hand*, un sistema de recomendación de noticias en el que se integraron todos los modelos

presentados, y en el que los contenidos textuales de las noticias son anotados con conceptos de un conjunto de ontologías que cubren diversos dominios generales de interés.

Los resultados obtenidos con el sistema reforzaron las conclusiones observadas previamente en los experimentos aislados, y proporcionaron nuevos hallazgos. Las recomendaciones personalizadas ayudaron a los usuarios a encontrar ítems relevantes, y la expansión semántica de preferencias facilitó las concurrencias entre perfiles de usuario y de ítem, mejorando la precisión sobre los ítems recomendados más relevantes, y mitigando los problemas de “arranque frío” y poca densidad de preferencias. La contextualización de los mecanismos de personalización aceleró el descubrimiento de ítems relacionados con los objetivos actuales de búsqueda, y fue altamente apreciada por los evaluadores. Finalmente, la consideración de recomendaciones híbridas multi-capa pareció mejorar aproximaciones colaborativas al calcular comparaciones parciales (focalizadas a intereses) de perfiles de usuario, reduciendo de este modo el efecto del problema de la “oveja negra”.

La experimentación realizada también proporcionó la oportunidad de recibir opiniones y sugerencias de los evaluadores sobre las funcionalidades y salidas del sistema. Entre otros aspectos, percibieron la necesidad de incorporar una fase de desambiguación en el proceso de anotación, y de abordar el problema de la no diversidad de recomendaciones, pues ítems muy similares se presentaron cercanos en las páginas de recomendaciones. Adicionalmente, sugirieron mejoras en el editor de perfiles, como la integración de un módulo de recomendación de preferencias en tiempo real que tuviese en cuenta conceptos similares a los ya introducidos (sinónimos, co-ocurrencias, etc.).

News@hand no sólo sirvió para realizar evaluaciones conjuntas de las estrategias de recomendación, sino también para poner de manifiesto dificultades originadas al trasladar los modelos basados en ontologías a una aplicación real. Al construir el sistema surgieron retos de investigación para los cuales se desarrollaron novedosas y originales soluciones. En concreto, se tuvo que implementar una técnica de poblado (i.e., creación de instancias) de las ontologías de dominio, un mecanismo automático de anotación semántica de las artículos, y una estrategia de conversión de etiquetas (del inglés *tags*) o palabras clave a conceptos ontológicos existentes.

La contribución final de la tesis se resume como sigue:

La implementación de un sistema prototipo en el que se han integrado y evaluado todos los modelos de recomendación presentados, y que constituye una plataforma sobre la cual se desarrollen nuevas propuestas que aborden temas de investigación abiertos en los campos de la personalización y los sistemas de recomendación.

Las ventajas que este sistema de recomendación supone han sido ya mencionadas:

- *Obtención de resultados empíricos más realistas.* A través de *News@hand* se han realizado experimentos más realistas que los llevados a cabo en las evaluaciones aisladas de cada uno de los modelos estudiados. Así mismo, el sistema ha facilitado la obtención de valoraciones subjetivas de los usuarios que podrán tenerse en cuenta para mejorar los modelos de recomendación.
- *Descubrimiento, análisis y resolución (no definitiva) de dificultades y problemas que surgen al implantar un sistema de recomendación semántico.* La implementación de *News@hand* originó retos que han tenido que resolverse en esta tesis, como por ejemplo el poblado de ontologías, la anotación semántica de textos y la generación semi-automática de perfiles de usuario. Aunque las soluciones ofrecidas no son definitivas, representan ideas novedosas e interesantes para la comunidad científica.
- *Disponibilidad de una plataforma de desarrollo y evaluación.* *News@hand* puede ser adaptado para incorporar nuevas funcionalidades y modelos de personalización y recomendación, ofreciendo de este modo una plataforma con la que evaluar futuras propuestas.

D.2 Discusión y trabajo futuro

En esta tesis se han presentado una serie de modelos de recomendación que explotan la descripción semántica de preferencias de usuario y de contenidos de ítems para abordar algunos de los problemas existentes en los sistemas de recomendación actuales. Aunque se ha cubierto un considerable número de los problemas más importantes, aún se prevé que investigación relevante pueda llevarse a cabo en otros ámbitos del área. Por otra parte, además de nuevas líneas de trabajo, existen por supuesto aspectos de las propuestas presentadas que son susceptibles de ser revisados y mejorados.

Limitaciones no resueltas, posibles vías de actuación para solventarlas, y potenciales retos futuros son cuestiones que se plantean y comentan en las siguientes subsecciones.

D.2.1 Recursos semánticos

La eficacia de los sistemas basados en semántica depende de la riqueza de la representación de los metadatos en las bases de conocimiento, y de la calidad de las anotaciones de los contenidos. En el caso de sistemas de personalización y recomendación la precisión de los resultados también viene influenciada por la corrección y exhaustividad de las descripciones de las preferencias de los usuarios en sus perfiles.

El **diseño y construcción de ontologías** no se han abordado en esta tesis, pues estaban fuera del alcance de sus objetivos, y son temas de amplio estudio en diversas disciplinas de la Web Semántica. Bajo el epígrafe de Ingeniería Ontológica (del inglés *Ontological Engineering*) (Gómez-Pérez, Fernández-López, & Corcho, 2003), se engloban diferentes líneas de investigación:

- Definición y desarrollo de metodologías (Uschold & Grüninger, 1996) y herramientas (Gennari, et al., 2003) que asistan en el proceso de construcción de ontologías.
- Implementación de estrategias de re-utilización de conocimiento ontológico (*Ontology Reuse*), donde se integren varias fuentes semánticas (*Ontology Integration*) (Farquhar, Fikes, & Rice, 1996), o donde se determinen correspondencias entre conceptos (*Ontology Alignment* u *Ontology Matching*) (Euzenat & Shvaiko, 2007).
- La generación (semi)automática o aprendizaje de ontologías (*Ontology Learning*) (Maedche & Staab, 2001; Shamsfard & Barforoush, 2003) a partir de la extracción de conceptos y relaciones de un corpus u otros tipos de bases de datos.

En esta tesis se partió de ontologías de dominio ya construidas. Así, por ejemplo, para *News@hand* se usaron adaptaciones de la ontología IPTC. Estas ontologías tenían definidas las jerarquías, propiedades y relaciones de las clases, pero carecían de instancias. Por este motivo, se tuvo que desarrollar un mecanismo automático de **poblamiento de ontologías** (del inglés *Ontology Population*) (Brewster, Ciravegna, & Wilks, 2001), es decir, un procedimiento por el cual se identifiquen instancias de un corpus base, y se asocien a las clases ontológicas correspondientes. El método propuesto presenta la idea de explotar las categorías de Wikipedia. Dado un término a instanciar, extraído del texto de una noticia en el caso de *News@hand*, éste es buscado en Wikipedia. Si el término existe en esa base de datos, se obtiene una página web que contiene una descripción y una serie de categorías pre-establecidas del concepto. Mediante una heurística que enlaza esas categorías con las clases ontológicas, se determina la clase que mejor se ajusta a la instancia a crear. La heurística ofreció buenos resultados, pero puede mejorarse procesando los textos descriptivos de los conceptos, con el fin de resolver casos de ambigüedad entre clases (Cucerzan, 2007) o extraer relaciones semánticas entre instancias (Ruiz-Casado, Alfonseca, & Castells, 2006).

Una vez se han poblado las ontologías de dominio se puede proceder a la **anotación de los contenidos** (Uren, et al., 2006). La anotación consiste en la identificación de conceptos (clases e instancias) ontológicos en los contenidos de los ítems. Es un problema difícil de resolver y es ampliamente estudiado en áreas de investigación como la Recuperación de Información, el Procesamiento del Lenguaje

Natural y la Web Semántica. En esta tesis la anotación se ha abordado con la adaptación de las herramientas de procesamiento lingüístico *Wraetlic* (Alfonseca, Moreno-Sandoval, Guirao, & Ruiz-Casado, 2006). Estas herramientas procesan textos a nivel morfológico y sintáctico para extraer todos sus nombres, incluyendo nombres propios y compuestos. Con los nombres extraídos se aplica una heurística que mediante similitudes morfológicas localiza las clases o instancias afines. En esta aproximación no se realiza ningún análisis a nivel semántico. Por ello se dieron situaciones de ambigüedad en las que se eligió erróneamente el significado de los conceptos asociados. Al igual que en el poblamiento ontologías, en este caso también se debería llevar a cabo un proceso de desambiguación semántica de los conceptos identificados.

Aparte de las bases de conocimiento ontológico y de anotaciones semánticas, otro de los recursos empleado por los modelos de recomendación presentados es el de los **perfiles de usuario**. Los perfiles usados en este trabajo fueron manualmente creados por los usuarios. Para facilitar esta tarea en los experimentos realizados se proveyó a los evaluadores de herramientas de creación y edición de sus preferencias. Así, por ejemplo, *News@hand* posee un explorador de ontologías que permite al usuario visualizar la jerarquía de clases, expandiendo y contrayendo relaciones taxonómicas, listar las instancias de cada clase, y buscar conceptos con ayuda de mecanismos que “auto-completan” los términos de las consultas a medida que se van escribiendo. Los usuarios valoraron muy positivamente la herramienta anterior, pero sugirieron ciertas mejoras, entre las que destaca la incorporación de un módulo de recomendación de preferencias. Cuando se está creando el perfil, el sistema podría sugerir nuevas preferencias que estuviesen relacionadas con las ya introducidas. Las relaciones consideradas podrían proceder de similitudes semánticas o de correlaciones entre conceptos a nivel de contenidos o a nivel de perfiles de todos los usuarios (Jäschke, Marinho, Hotho, Schmidt-Thieme, & Stumme, 2007; Sigurbjörnsson & Van Zwol, 2008). Además de la ayuda proporcionada por las interfaces gráficas de las aplicaciones desarrolladas, en esta tesis se ha propuesto una estrategia que convierte automáticamente etiquetas sociales (del inglés *social tags*) a conceptos ontológicos. El usuario, de este modo, en vez de tener que buscar conceptos existentes, directamente introduce términos que describen sus gustos e intereses, y el sistema intenta encontrarlos en las ontologías. Este tipo de estrategias, que no son triviales, pues han de considerar errores gramaticales, acrónimos, sinónimos, etc., es un tema de investigación de especial interés para aplicaciones sociales y está en pleno auge actualmente (Specia & Motta, 2007; Van Damme, Hepp, & Siorpaes, 2007; Hess, Maass, & Dierick, 2008; Van der Sluijs & J, 2008). Finalmente, otra de las soluciones posibles se basa en que el usuario no declare preferencia alguna, y que sea el sistema el encargado de deducirlas o aprenderlas a través de las acciones del usuario. Por estar fuera del alcance de la tesis, esta

aproximación no fue estudiada. Sin embargo, otros investigadores ya han iniciado trabajos en este ámbito utilizando *News@hand* (Picault & Ribière, 2008).

D.2.2 Modelos de recomendación

Las evaluaciones realizadas demostraron que la **recomendación contextualizada** mejora la eficacia del modelo básico de recuperación de contenidos personalizada, al focalizar los intereses actuales del usuario. El contexto se definió como el conjunto de conceptos semánticos (ponderados) que forman parte de las anotaciones de aquellos ítems que han sido recientemente visualizados o evaluados por el usuario. Esta descripción, aún siendo útil, puede ser enriquecida con información semántica de otras fuentes externas (Chirita, Firan, & Nejdl, 2006), como por ejemplo las tareas planificadas en una agenda electrónica, los mensajes recientes de un cliente de correo electrónico, o los sitios web incluidos como favoritos en un navegador web. En la propuesta los pesos asignados a los conceptos del contexto decaen con el tiempo, asumiendo la hipótesis de que el foco de interés va desapareciendo progresivamente para dar paso a uno nuevo. Sin embargo, otras hipótesis son plausibles (White, Ruthven, Jose, & Van Rijsbergen, 2005), y darían lugar a nuevas estrategias de actualización del contexto semántico. Una vez definidos los mecanismos de creación y evolución del contexto, éste tiene que integrarse con el modelo de recomendación personalizada. Como primera aproximación se estudió la combinación lineal de ambos. No obstante, de nuevo, otras alternativas podrían tenerse en cuenta (Vallet, Castells, Fernández, Mylonas, & Avrithis, 2007).

En relación a la **recomendación orientada a grupo**, se hace evidente la necesidad de una mayor experimentación. De hecho, las estrategias de modelado de grupo propuestas en esta tesis son las únicas que no se evaluaron en *News@hand*, a pesar de estar integradas en el sistema. Como mejora futura de las técnicas anteriores, se plantea la inclusión de nuevos factores en los métodos de combinación de perfiles, que podrían estar relacionados con diversas fuentes de contexto, como la localización, fecha y hora actuales, la edad y sexo de los usuarios, etc. (Ardissono, Goy, Petrone, Segnan, & Torasso, 2003). Así, por ejemplo, no es lo mismo recomendar un programa televisivo de sobremesa a una familia con niños pequeños, que sugerir una película a una pareja después de una cena romántica.

La **recomendación híbrida multi-capas** puede ser considerada como la contribución más significativa de la tesis, y de ahí que se haya probado de forma más exhaustiva, tanto con usuarios reales en diferentes escenarios, como con conjuntos de datos creados artificialmente. Sin embargo, uno de los aspectos que no se ha analizado es su rendimiento. Aunque, de forma análoga a las estrategias de filtrado colaborativo, las matrices de similitud entre usuarios e ítems pueden recalcularse con un proceso autónomo, de forma periódica y sin afectar al rendimiento del sistema, la eficiencia de los algoritmos empleados puede mejorarse considerablemente. En

concreto, la técnica de agrupamiento (del inglés *clustering*) de conceptos para generar las comunidades de interés multi-capas, emplea estrategias jerárquicas que crean grupos de conceptos a K niveles, donde K es el número de conceptos (Duda, Hart, & Stork, 2001). Se prevé la aplicación de técnicas de agrupamiento más escalables basadas en SVD y LSI (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990; Landauer, Foltz, & Laham, 1998) o *co-clustering* (George & Merugu, 2005). Además del problema de la escalabilidad, otra línea de trabajo que se plantea es la de estudiar nuevos modelos de comparación y combinación de preferencias y contenidos semánticos de forma colaborativa. Recientemente han surgido aproximaciones muy similares a la de esta tesis, cuya representación del conocimiento ontológica es compartida (incluyendo incluso la idea de expansión semántica), pero que abogan por modelos de recomendación alternativos. Por ejemplo, en (Mobasher, Jin, & Zhou, 2004), los autores presentan una estrategia de filtrado colaborativo en la que la similitud entre dos ítems (ver Sección 2.3.2) se define a partir de una medida que tiene en cuenta los conceptos semánticos comunes a ambos. En (Gauch, Chaffee, & Pretschner, 2003), por el contrario, las medidas de similitud entre ítems se basan en las distancias entre conceptos dentro de las estructuras ontológicas.

D.2.3 Plataforma de evaluación

La construcción de *News@hand* tuvo una doble motivación. Por una parte, sería utilizado como plataforma de evaluación de los modelos de recomendación. El sistema permitiría la realización de experimentos menos restringidos que los llevados a cabo con anterioridad. Los usuarios interactuarían con los modelos durante periodos de tiempo más largos, proporcionando mayor cantidad de información con la que medir más fidedignamente la eficacia de las propuestas. Por otra parte, su implementación y posterior puesta en marcha servirían para poner de manifiesto los problemas y dificultades que conlleva la implantación de una aplicación basada en tecnologías semánticas. De hecho, fueron esos los aspectos que originaron las técnicas automáticas de poblamiento de ontologías y transformación de términos a conceptos ontológicos citados anteriormente.

La experiencia y resultados empíricos obtenidos en los experimentos, y los comentarios recibidos por parte de los evaluadores serán utilizados para corregir errores encontrados en el sistema, y para realizar cambios y mejoras en la propia metodología de evaluación. Una vez que *News@hand* tenga operativas todas sus funcionalidades será hecho público en la Web. En ese momento, se espera con optimismo poder realizar **experimentos a mayor escala**, con un número significativamente grande de usuarios, y durante periodos de tiempo de varios meses (Middleton, Shadbolt, & Roure, 2004).

Por supuesto, las evaluaciones futuras no estarán limitadas a las propuestas planteadas en este trabajo. Se prevé la investigación adicional de otros temas

pendientes de resolver en el área de los sistemas de recomendación. En concreto, se presenta interesante el estudio de ***modelos de recomendación dirigidos por consulta*** (Adomavicius, Tuzhilin, & Zheng, 2005), y técnicas que faciliten la ***comprensibilidad de las recomendaciones*** obtenidas (Tintarev & Masthoff, 2007). Para el primer caso, se podrían diseñar lenguajes de definición de recomendaciones que sean extensiones de lenguajes de consulta ontológica (e.g., RDQL), o se podrían combinar modelos de recomendación con mecanismos de búsqueda semántica (Castells, Fernández, & Vallet, 2007). Por otra parte, para el segundo caso, se podrían evaluar técnicas que infieran y expliquen los conceptos y relaciones semánticas que han determinado las recomendaciones dadas al usuario.

References

- Adomavicius, G., & Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17 (6), 734-749.
- Adomavicius, G., Tuzhilin, A., & Zheng, R. (2005). RQL: A Query Language for Recommender Systems. *Stern School of Business, New York University* .
- Agosti, M., Crestani, F., Gradenigo, G., & Mattiello, P. (1990). An Approach to Conceptual Modelling of IR Auxiliary Data. *Proceedings of the 9th IEEE International Phoenix Conference on Computer and Communications*, (pp. 500-505). Scottsdale, AZ, USA.
- Agosti, M., Melucci, M., & Crestani, F. (1995). Automatic Authoring and Construction of Hypertext for Information Retrieval. 15 (1), 15-24.
- Ahn, J., Brusilovsky, P., Grady, J., He, D., & Syn, S. Y. (2007). Open User Profiles for Adaptive News Systems: Help or Harm? *Proceedings of the 16th International World Wide Web Conference (WWW 2007)*, (pp. 11-20). Banff, AB, Canada.
- Alani, H., O'Hara, K., & Shadbolt, N. R. (2002). ONTOCOPI: Methods and Tools for Identifying Communities of Practice. *Proceedings of the 17th IFIP World Computer Congress (WCC 2002)*, (pp. 225-236). Montreal, QC, Canada.
- Alfonseca, E., Moreno-Sandoval, A., Guirao, J. M., & Ruiz-Casado, M. (2006). The Wraetlic NLP Suite. *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*. Genoa, Italy.
- Ali, K., & Van Stam, W. (2004). TiVo: Making Show Recommendations using a Distributed Collaborative Filtering Architecture. *Proceedings of the 10th International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD 2004)*, (pp. 394 - 401). Seattle, WA, USA.

- Anand, S. S., & Mobasher, B. (2007). Contextual Recommendation. In B. Berendt, A. Hotho, D. Mladenice, & G. Semeraro, *From Web to Social Web: Discovering and Deploying User and Content Profiles* (Vol. 4737, pp. 142-160). Berlin, Germany: Springer-Verlag. Lecture Notes in Computer Science.
- Ardissono, L., Goy, A., Petrone, G., Segnan, M., & Torasso, P. (2003). INTRIGUE: Personalized Recommendation of Tourist Attractions for Desktop and Handset Devices. *Applied Artificial Intelligence*, 17 (8-9), 687-714.
- Baeza-Yates, R., & Ribeiro Neto, B. (1999). *Modern Information Retrieval*. Harlow, UK: Addison-Wesley.
- Balabanovic, M., & Shoham, Y. (1997). Fab: Content-Based Collaborative Recommendation. *Communications of the ACM archive*, 40 (3), 66-72.
- Basu, C., Hirsh, H., & Cohen, W. (1998). Recommendation as Classification: Using Social and Content-based Information in Recommendation. *Proceedings of the AAAI 1998 Workshop on Recommender Systems 1998*, (pp. 11-15). Chicago, IL, USA.
- Benjamins, V. R., Davies, J., Baeza-Yates, R. A., Mika, P., Zaragoza, H., Greaves, M., et al. (2008). Near-Term Prospects for Semantic Technologies. *IEEE Intelligent Systems*, 23 (1), 76-88.
- Berners-Lee, T. (2000). *Weaving the Web*. New York, NY, USA: Harper Collins.
- Berners-Lee, T., Hendler, J., & Lassila, O. (2001). *The Semantic Web: A New Form of the Web Content that is Meaningful to Computers Will Unleash a Revolution of New Possibilities*. Retrieved from Scientific American (May 2001).
- Billsus, D. & Pazzani, M. J. (1998). Learning Collaborative Information Filters. *Proceedings of the 15th International Conference on Machine Learning (ICML 1998)*, (pp. 46-54). Madison, WI, USA.
- Billsus, D., & Pazzani, M. J. (1999). A Personal News Agent that Talks Learns and Explains. *Proceedings of the 3rd International Conference on Autonomous Agents (Agents 1999)*, (pp. 268-275). Seattle, WA, USA.
- Billsus, D., & Pazzani, M. J. (2000). User Modeling for Adaptive News Access. *User Modeling and User-Adapted Interaction*, 10 (2-3), 147-180.
- Borda, J. C. (1781). *Mémoire sur les élections au Scrutin*. Histoire de l'Académie Royale des Sciences.

- Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering. *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI 1998)*, (pp. 43-52). Madison, WI, USA.
- Breitman, K. K., Casanova, M. A., & Truszkowski, W. (2007). *Semantic Web: Concepts, Technologies and Applications*. London, UK: Springer-Verlag.
- Brewster, C., Ciravegna, F., & Wilks, Y. (2001). Knowledge Acquisition for Knowledge Management: Position Paper. *Proceedings of the IJCAI 2001 Workshop on Ontology Learning*. Seattle, WA, USA.
- Burger, J., Cardie, C., Chaudhri, V., Gaizauskas, R., Harabagiu, S., Israel, D., et al. (2001). *Issues, Tasks and Program Structures to Roadmap Research in Question and Answering*. Retrieved from National Institute of Standards and Technology.
- Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, 12 (4), 331-370.
- Castells, P., Fernández, M., Vallet, D., Mylonas, P., & Avrithis, Y. (2005). Self-Tuning Personalized Information Retrieval in an Ontology-Based Framework. *Proceedings of the 1st IFIP WG 2.12 & WG 12.4 International Workshop on Web Semantics (SWWS 2005)*. Agia Napa, Cyprus.
- Castells, P., Fernández, M., & Vallet, D. (2007). An Adaptation of the Vector-Space Model for Ontology-based Information Retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 19 (2), 261-272.
- Castells, P., Foncillas, B., Lara, R., Rico, M., & Alonso, J. L. (2004). Semantic Web Technologies for Economic and Financial Information Management. *Proceedings of the 1st European Semantic Web Symposium (ESWS 2004)*, (pp. 473-487). Heraklion, Greece.
- Catells, P. (2003). La Web Semántica. In C. Bravo, & M. A. Redondo, *Sistemas Interactivos y Colaborativos en la Web* (pp. 195-212). Ediciones de la Universidad de Castilla.
- Cattuto, C., Loreto, V., & Pietronero, L. (2007). Collaborative Tagging and Semiotic Dynamics. *Proceedings of the National Academy of Sciences*, 104, 1461-1469.
- Chen, H., & Lynch, K. J. (1992). Automatic Construction of Networks of Concepts Characterizing Document Databases. *IEEE Transactions on System, Man and Cybernetics*, 22 (5), 885-902.

- Chirita, P. A., Costache, S., Handschuh, S., & Nejdl, W. (2007). PTAG - Large Scale Automatic Generation of Personalized Annotation Tags for the Web. *Proceedings of the 16th International World Wide Web Conference (WWW 2007)*, (pp. 845-854). Banff, AB, Canada.
- Chirita, P., Firan, C. S., & Nejdl, W. (2006). Summarizing Local Context to Personalize Global Web Search. *Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM 2006)*, (pp. 287-296). Arlington, VA, USA.
- Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., & Sartin, M. (1999). Combining Content-Based and Collaborative Filters in an Online Newspaper. *Proceedings of the SIGIR 1999 Workshop on Recommender Systems: Algorithms and Evaluation*. Berkeley, CA, USA.
- Coates, A. B. (2001). The Role of XML in Finance. *Proceedings of the XML Conference and Exposition 2001*. Orlando, FL, USA.
- Cohen, P. R., & Kjeldsen, R. (1987). Information Retrieval by Constrained Spreading Activation in Semantic Networks. *Information Processing and Management*, 23 (4), 255-268.
- Contreras, J., Benjamins, V. R., Blázquez, M., Losada, S., Salla, R., Sevilla, J., et al. (2004). A Semantic Portal for the International Affairs Sector. *Proceedings of the 14th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2004)*, (pp. 203-215). Northamptonshire, UK.
- Copeland, A. H. (1951). *A Reasonable Social Welfare Function*. Seminar on Applications of Mathematics to the Social Sciences, University of Michigan.
- Crestani, F. (1997). Application of Spreading Activation Techniques in Information Retrieval. *Artificial Intelligence Review*, 11(6), 453-482.
- Crestani, F., & Lee, P. L. (2000). Searching the Web by Constrained Spreading Activation. *Information Processing and Management*, 36 (4), 585-605.
- Croft, W. B. (1986). User-Specified Domain Knowledge for Document Retrieval. *Proceedings of the 9th ACM Conference on Research and Development in Information Retrieval (SIGIR 1986)*, (pp. 201-206). Pisa, Italy.
- Crouch, C. J. (1990). An Approach to the Automatic Construction of Global Thesauri. *Information Processing and Management*, 26 (5), 629-640.

- Cucerzan, S. (2007). Large-Scale Named Entity Disambiguation Based on Wikipedia Data. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, (pp. 708-716). Prague, Czech Republic.
- Das, A., Datar, M., Garg, A., & Rajaram, S. (2007). Google News Personalisation: Scalable Online Collaborative Filtering. *Proceedings of the 16th International World Wide Web Conference (WWW 2007)*, (pp. 271-280). Banff, AB, Canada.
- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the Society for Information Science*, 41 (6), 391-407.
- Delgado, J., & Ishii, N. (1999). Memory-Based Weighted-Majority Prediction for Recommender Systems. *Proceedings of the SIGIR 1999 Workshop on Recommender Systems: Algorithms and Evaluation*. Berkeley, CA, USA.
- Deshpande, M., & Karypis, G. (2004). Item-based Top-N Recommendation Algorithms. *ACM Transactions on Information Systems*, 22 (1), 143-177.
- Duda, R. O., Hart, P., & Stork, D. G. (2001). *Pattern Classification*. New York, NY, USA: John Wiley.
- Dudek, J. (2001). XML in Health Care. *Proceedings of XML Europe 2001 Conference*. Berlin, Alemania.
- Dumais, S. T. (1994). Latent Semantic Indexing (LSI) and TREC-2. *Proceedings of the 2nd Text Retrieval Conference (TREC2)*, (pp. 105-116).
- Euzenat, J., & Shvaiko, P. (2007). *Ontology Matching*. Heidelberg, Germany: Springer-Verlag.
- Farquhar, A., Fikes, R., & Rice, J. (1996). The Ontolingua Server: A Tool for Collaborative Ontology Construction. *International Journal of Human-Computer Studies*, 46 (6), 707-727.
- Gauch, S., Chaffee, J., & Pretschner, A. (2003). Ontology-based Personalized Search and Browsing. *Web Intelligence and Agent Systems*, 1 (3-4), 219-234.
- Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., & Schneider, L. (2002). Sweetening Ontologies with DOLCE. *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2002)*, (pp. 166-181). Sigüenza, Spain.

- Gennari, J., Musen, M. A., Fergerson, R. W., Grosso, W. E., Crubezy, M., Eriksson, H., et al. (2003). The Evolution of Protege: An Environment for Knowledge-Based Systems Development. *International Journal of Human-Computer Studies*, 58 (1), 89-123.
- George, T., & Merugu, S. (2005). A Scalable Collaborative Filtering Framework based on Co-Clustering. *Proceedings of the 5th IEEE Conference on Data Mining (ICDM 2005)*, (pp. 625-628). Houston, TX, USA.
- Golbeck, J., & Hendler, J. (2006). FilmTrust: Movie Recommendations Using Trust in Web-based Social Networks. *Proceedings of the IEEE Consumer Communications and Networking Conference*. Las Vegas, NV, USA.
- Golbeck, J., & Mannes, A. (2006). Using Trust and Provenance for Content Filtering on the Semantic Web. *Proceedings of the 1st Workshop on Models of Trust on the Web (MTW 2006)*. Edinburgh, UK.
- Goldman, S. A., & Warmuth, M. K. (1995). Learning Binary Relations Using Weighted Majority Voting. *Machine Learning*, 20 (3), 245-271.
- Gómez-Pérez, A., Fernández-López, M., & Corcho, O. (2003). *Ontological Engineering*. London, UK: Springer-Verlag.
- Good, N., Schafer, J. B., Konstan, J. A., Borchers, A., Sarwar, B., Herlocker, J., et al. (1999). Combining Collaborative Filtering with Personal Agents for Better Recommendations. *Proceedings of the 16th National Conference of the American Association of Artificial intelligence (AAAI 1999)*, (pp. 439-446). Orlando, FL, USA.
- Gruber, T. (1993). A Translation Approach to Portable Ontologies. *Knowledge Acquisition*, 5 (2), 199-220.
- Gruber, T. (2008). Collective Knowledge Systems: Where the Social Web meets the Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web*, to appear.
- Guarino, N. (1998). Formal Ontology and Information Systems. *Proceedings of the 1st International Conference on Formal Ontologies in Information Systems (FOIS 1998)*, (pp. 3-15). Trento, Italy.
- Guha, R. V., McCool, R., & Miller, E. (2003). Semantic Search. *Proceedings of the 12th International World Wide Web Conference (WWW 2003)*, (pp. 700-709). Budapest, Hungary.

- Harbourt, A. M., Syed, E. J., Hole, W. T., & Kingsland, L. C. (1993). The Ranking Algorithm of the Coach Browser for the UMLS Metathesaurus. *Proceedings of the 17th Annual Symposium on Computer Applications in Medical Care*, (pp. 720-724). Washington, DC, USA.
- Hendler, J. A. (2001). Agents and the Semantic Web. *IEEE Intelligent Systems*, 16 (2), 30-37.
- Herlocker, J., Konstan, J. A., & Riedl, J. (2000). Explaining Collaborative Filtering Recommendations. *Proceedings of the ACM 2000 Conference on Computer Supported Cooperative Work (CSCW 2000)*, (pp. 241-250). Philadelphia, PA, USA.
- Herlocker, J., Konstan, J. A., Borchers, A., & Riedl, J. (1999). An Algorithmic Framework for Performing Collaborative Filtering. *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR 1999)*, (pp. 230-237). Berkeley, CA, USA.
- Herlocker, J. Konstan, J. A., Terveen, L., & Riedl, J. (2004). Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems*, 22 (1), 5-53.
- Hersh, W. R., Hickam, D. H., & Leone, T. J. (1992). Words, Concepts, or Both: Optimal Indexing Units for Automated Information Retrieval. *Proceedings of the 16th Annual Symposium on Computer Applications in Medical Care*, (pp. 644-648). Baltimore, MD, USA.
- Hersh, W. R., & Greenes, R. A. (1990). SAPHIRE - An Information Retrieval System Featuring Concept Matching, Automatic Indexing, Probabilistic Retrieval, and Hierarchical Relationships. *Computers and Biomedical Research*, 23, 410-425.
- Hess, A., Maass, C., & Dierick, F. (2008). From Web 2.0 to Semantic Web: A Semi-Automated Approach. *Proceedings of the 1st International Workshop on Collective Semantics: Collective Intelligence and the Semantic Web (CISWeb 2008)*, (pp. 20-34). Tenerife, Spain.
- Hill, Stead, L., Rosenstein, M., & Furnas, G. (1995). Recommending and Evaluating Choices in a Virtual Community of Use. *Proceedings of the 13th International Conference on Human Factors in Computing Systems (CHI 1995)*, (pp. 194-201). Denver, CO, USA.
- Hotho, A., Jäschke, R., Schmitz, C., & Stumme, G. (2006). Information Retrieval in Folksonomies: Search and Ranking. *Proceedings of the 3rd European Semantic Web Conference (ESWC 2006)*, (pp. 411-426). Budva, Montenegro.

- Jameson, A., Baldes, S., & Kleinbauer, T. (2003). Enhancing Mutual Awareness in Group Recommender Systems. *Proceedings of the 1st Workshop on Intelligent Techniques for Web Personalization (ITWP 2003)*. Acapulco, Mexico.
- Järvelin, K., Kekäläinen, J., & Niemi, T. (2001). ExpansionTool: Concept-Based Query Expansion and Construction. *Information Retrieval*, 4 (3-4), 231-255.
- Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., & Stumme, G. (2007). Tag Recommendations in Folksonomies. *Knowledge Discovery in Databases (PKDD) 2007*, 506-514.
- Jin, X., & Mobasher, B. (2003). Using Semantic Similarity to Enhance Item-Based Collaborative Filtering. *Proceedings of the 2nd LASTED International Conference on Information and Knowledge Sharing*. Scottsdale, AZ, USA.
- Jones, S. (1993). A Thesaurus Data Model for an Intelligent Retrieval System. *Journal of Information Science*, 19, 167-178.
- Jones, G. J., Quested, D. J., & Thomson, K. E. (2000). Personalised Delivery of News Articles from Multiple Sources. *Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2000)*, (pp. 340-343). Lisbon, Portugal.
- Karvounarakis, G., Alexaki, S., Christophides, V., Plexousakis, D., & Scholl, M. (2002). RQL: A Declarative Query Language for RDF. *Proceedings of the 11th International World Wide Web Conference (WWW 2002)*, (pp. 592-603). Honolulu, HI, USA.
- Karypis, K. (2001). Evaluation of Item-Based Top-N Recommendation Algorithms. *Proceedings of the 10th International Conference on Information and Knowledge Management (ACM CIKM 2001)*, (pp. 247-254). Atlanta, GA, USA.
- Kiryakov, A., Popov, B., Terziev, I., Manov, D., & Ognyanoff, D. (2004). Semantic Annotation, Indexing, and Retrieval. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2 (1), 49-79.
- Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., & Riedl, J. (1997). GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM*, 40 (3), 77-87.
- Kruk, S. R., & Decker, S. (2005). Semantic Social Collaborative Filtering with FOAFRealm. *Proceedings of the 1st Semantic Desktop Workshop*. Galway, Ireland.

- Krulwich, B., & Burkey, C. (1997). The Infofinder Agent: Learning User Interests Through Heuristic Phrase Extraction. *IEEE Intelligent Systems and Their Applications*, 12 (5), 22-27.
- Landauer, T. K., Foltz, P. W., & Laham, D. (1998). Introduction to Latent Semantic Analysis. *Discourse Processes*, 259-284.
- Lang, K. (1995). NewsWeeder: Learning to Filter Netnews. *Proceedings of the 12th International Conference on Machine Learning (ML 1995)*, (pp. 331-339). Tahoe City, CA, USA.
- Ledsche, T. A., & Berry, M. W. (1997). Large-Scale Information Retrieval with Latent Semantic Indexing. *Information Sciences*, 100 (1-4), 105-137.
- Lee, W. S. (2001). Collaborative Learning for Recommender Systems. *Proceedings of the 18th International Conference on Machine Learning*, (pp. 314-321). Williamstown, MA, USA.
- Levenshtein, V. I. (1966). Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics - Doklady*, 10, 707-710.
- Lewis, D. D., & Gale, W. A. (1994). A Sequential Algorithm for Training Text Classifiers. *Proceedings of the 17th ACM Conference on Research and Development in Information Retrieval (SIGIR 1994)*, (pp. 3-12). Dublin, Ireland.
- Linden, G., Smith, B., & York, J. (2003). Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Computing*, 7 (1), 76-80.
- Liu, H., Maes, P., & Davenport, G. (2006). Unravelling the Taste Fabric of Social Networks. *International Journal on Semantic Web and Information Systems*, 2 (1), 42-71.
- Luke, S., Spector, L., & Rager, D. (1996). Ontology-based Knowledge Discovery on the World-Wide Web. In A. Franz, & H. Kitano, *Internet-based Information Systems: Papers from the AAAI Workshop* (pp. 96-102). Menlo Park, CA, USA.
- Madala, R., Takenobu, T., & Hozumi, T. (1998). The Use of WordNet in Information Retrieval. *Proceedings of the Conference on the Use of WordNet in Natural Language Processing Systems*, (pp. 31-37). Montreal, QC, Canada.
- Maedche, A., & Staab, S. (2001). Learning Ontologies for the Semantic Web. *IEEE Intelligent Systems and Their Applications*, 16 (2), 72-79.

- Maedche, A., Staab, S., Stojanovic, N., Studer, R., & Sure, Y. (2003). SEMantic portAL: The SEAL Approach. In D. Fensel, J. A. Hendler, H. Lieberman, & W. Wahlster, *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential* (pp. 317-359).
- Maes, P. (1994). Agents that Reduce Work and Information Overload. *Communications of the ACM*, 37 (7), 31-40.
- Masthoff, J. (2004). Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers. *User Modeling and User-Adapted Interaction*, 14 (1), 37-85.
- Masthoff, J. (2005). The Pursuit of Satisfaction: Affective State in Group Recommender Systems. *Proceedings of the 10th International Conference on User Modeling (UM 2005)*, (pp. 297-306). Edinburgh, Scotland, UK.
- Mayfield, J., & Finin, T. (2003). Information Retrieval on the Semantic Web: Integrating Inference and Retrieval. *Proceedings of the SIGIR 2003 Workshop on the Semantic Web*. Toronto, ON, Canada.
- McCarthy, J. F., & Anagnost, T. D. (1998). MusicFX: An Arbiter of Group Preferences for Computer Supported Collaborative Workouts. *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work (CSCW 1998)*, (pp. 363-372). Seattle, WA, USA.
- McCarthy, K., Salamo, M., McGinty, L., & Smyth, B. (2006). CATS: A Synchronous Approach to Collaborative Group Recommendation. *Proceedings of the 19th International Florida Artificial Intelligence Research Society Conference (FLAIRS 2006)*. Melbourne Beach, FL, USA.
- McGuinness, D. L. (2003). Ontologies Come of Age. In D. Fensel, J. Hendler, H. Lieberman, & W. Wahlster, *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential* (pp. 171-195). Cambridge, MA, USA: MIT Press.
- Melville, P., Mooney, R. J., & Nagarajan, R. (2002). Content-Boosted Collaborative Filtering for Improved Recommendations. *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI 2002)*, (pp. 187-192). Edmonton, AB, Canada.
- Middleton, S. E., Roure, D. D., & Shadbolt, N. R. (2004). Ontology-based Recommender Systems. In S. Staab, & R. Studer, *Handbook on Ontologies* (pp. 477-498). Springer-Verlag, Series on Handbooks in Information Systems.
- Middleton, S. E., Shadbolt, N. R., & Roure, D. D. (2004). Ontological User Profiling in Recommender Systems. *ACM Transactions on Information Systems*, 22 (1), 54-88.

- Mika, P. (2005). Flink: Semantic Web Technology for the Extraction and Analysis of Social Networks. *Journal of Web Semantics*, 3 (2-3), 211-223.
- Mika, P. (2005). Ontologies are Us: A Unified Model of Social Networks and Semantics. *Proceedings of the 4th International Semantic Web Conference (ISWC 2005)*, (pp. 522-536). Galway, Ireland.
- Mika, P. (2005). Social Networks and the Semantic Web: The Next Challenge. *IEEE Intelligent Systems*, 20 (1), 80-93.
- Miller, G. A. (1995). WordNet: A Lexical Database for English. New Horizons in Commercial and Industrial Artificial Intelligence. *Communications of the ACM*, 38 (11), 39-41.
- Mobasher, B., Jin, X., & Zhou, Y. (2004). Semantically Enhanced Collaborative Filtering on the Web. In B. Berendt, A. Hotho, D. Mladenic, M. Van Someren, & M. Spiliopoulou, *Web Mining: From Web to Semantic Web* (pp. 57-76).
- Mooney, R. J., & Roy, L. (2000). Content-based Book Recommending Using Learning for Text Categorization. *Proceedings of the 5th ACM Conference on Digital Libraries*, (pp. 195-240). San Antonio, TX, USA.
- Mooney, R. J., Bennett, P. N., & Roy, L. (1998). Book Recommending Using Text Categorization with Extracted Information. *Proceedings of the AAAI 1998 Workshop on Recommender Systems*, (pp. 70-74). Madison, WI, USA.
- Nadjarbashi-Noghani, M., Zhang, J., Sadat, H., & Ghorbani, A. A. (2005). PENS: A Personalised Electronic News System. *Proceedings of the 3rd Annual Communication Networks and Services Research Conference (CNSR 2005)*, (pp. 31-38). Halifax, NS, Canada.
- Nakamura, A., & Abe, N. (1998). Collaborative Filtering Using Weighted Majority Prediction Algorithms. *Proceedings of the 15th International Conference on Machine Learning (ICML 1998)*, (pp. 395-403). Madison, WI, USA.
- Niwa, S., Doi, T., & Honiden, S. (2006). Web Page Recommender System based on Folksonomy Mining for ITNG'06 Submissions. *Proceedings of the 3rd International Conference on Information Technology (ITNG 2006)*, (pp. 388-393). Las Vegas, NV, USA.
- O'Connor, M., Cosley, D., Konstan, J. A., & Riedl, J. (2001). PolyLens: A Recommender System for Groups of Users. *Proceedings of the 7th European Conference on Computer Supported Cooperative Work (ECSCW 2001)*, (pp. 199-218). Bonn, Germany.

- Paice, C. D. (1991). A Thesaural Model of Information Retrieval. *Information Processing and Management*, 27, 433-447.
- Passin, T. B. (2004). *Explorer's Guide to the Semantic Web*. New York, NY, USA: Manning Publications.
- Pattanaik, P. K. (2001). *Voting and Collective Choice*. London, UK: Cambridge University Press.
- Pazzani, M. J. (1999). A Framework for Collaborative, Content-based, and Demographic Filtering. *Artificial Intelligence Review*, 13 (5-6), 393-408.
- Pazzani, M. J., & Billsus, D. (1997). Learning and Revising User Profiles: The Identification of Interesting Websites. *Machine Learning*, 27 (3), 313-331.
- Pennock, D., Horvitz, E., Lawrence, S., & Giles, C. L. (2000). Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-Based Approach. *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI 2000)*, (pp. 473-480). Stanford, CA, USA.
- Picault, J., & Ribi re, M. (2008). An Empirical User Profile Adaptation Mechanism that Reacts to Shifts of Interests. *Submitted to the 18th European Conference on Artificial Intelligence (ECAI 2008)*. Patras, Greece.
- Popov, B., Kiryakov, A., Ognyanoff, D., Manov, D., & Kirilov, A. (2004). KIM - A Semantic Platform for Information Extraction and Retrieval. *Journal of Natural Language Engineering*, 10 (3-4), 375-392.
- R ck, C., Arbanowski, S., & Steglich, S. (2006). Context-aware, Ontology-based Recommendations. *Proceedings of the International Symposium on Applications and the Internet Workshops (SAINTW 2006)*, (pp. 98-104). Phoenix, AZ, USA.
- Rau, L. R. (1987). Knowledge Organization and Access in a Conceptual Information System. *Artificial Intelligence and Information Retrieval*, 23 (4), 269-283.
- Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. *Proceedings of the ACM 1994 Conference on Computer Supported Cooperative Work (CSCW 1994)*, (pp. 175-186). Chapel Hill, NC, USA.
- Rocha, C., Schwabe, D., & de Arag o, M. P. (2004). A Hybrid Approach for Searching in the Semantic Web. *Proceedings of the 13th International World Wide Web Conference (WWW 2004)*, (pp. 374-383). New York, NY, USA.

- Ruiz-Casado, M., Alfonseca, E., & Castells, P. (2006). From Wikipedia to Semantic Relationships: A Semi-automated Annotation Approach. *Proceedings of the 1st Workshop on Semantic Wikis: From Wiki to Semantics*. Budva, Montenegro.
- Salton, G., & McGill, M. J. (1986). *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill.
- Salton, G., & Lesk, M. E. (1971). Information Analysis and Dictionary Construction. In G. Salton, *The SMART Retrieval System* (pp. 115-142). Englewood Cliffs, N. J., USA: Prentice-Hall.
- Sarwar, B. M., Konstan, J. A., Borchers, A., Herlocker, J., Miller, B., & Riedl, J. (1998). Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System. *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, (pp. 345-354). Seattle, WA, USA.
- Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. (2000). Analysis of Recommendation Algorithms for E-Commerce. *Proceedings of the 2nd Annual ACM Conference on Electronic Commerce (EC 2000)*, (pp. 158-167). Minneapolis, MN, USA.
- Sarwar, B. M., Karypis, G., Konstan, J., & Riedl, J. (2000). Application of Dimensionality Reduction in Recommender Systems - A Case Study. *Proceedings of the WebKDD Workshop*. Boston, MA, USA.
- Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. (2001). Item-Based Collaborative Filtering Recommendation Algorithms. *Proceedings of the 10th International World Wide Web Conference (WWW 2001)*, (pp. 285-295). Hong Kong, China.
- Schein, A. I., Popescul, A., & Ungar, L. H. (2001). Generative Models for Cold-Start Recommendations. *Proceedings of the SIGIR 2001 Workshop on Recommender Systems*. New Orleans, LA, USA.
- Shamsfard, M., & Barforoush, A. (2003). The State of the Art in Ontology Learning: A Framework for Comparison. *The Knowledge Engineering Review*, 18 (4), 293-316.
- Shardanand, U., & Maes, P. (1995). Social Information Filtering: Algorithms for Automating 'Word of Mouth'. *Proceedings of the Conference on Human Factors in Computing Systems (CHI 1995)*, (pp. 210-217). San Francisco, CA, USA.
- Shoval, P., Maidel, V., & Shapira, B. (2008). An Ontology- Content-based Filtering Method. *International Journal of Information Theories and Applications*, 15, 303-318.

- Shoval, P. (1981). Expert/Consultation System for a Retrieval Data-base with Semantic Network of Concepts. *Proceedings of the 4th ACM Conference on Information Storage and Retrieval (SIGIR 1981)*, (pp. 145-149). Oakland, CA, USA.
- Sieg, A., Mobasher, B., & Burke, R. (2007). Ontological User Profiles for Personalized Web Search. *Proceedings of AAAI 2007 Workshop on Intelligent Techniques for Web Personalization*, (pp. 84-91). Vancouver, BC, Canada.
- Sigurbjörnsson, B., & Van Zwol, R. (2008). Flickr Tag Recommendation based on Collective Knowledge. *Proceeding of the 17th International World Wide Web Conference (WWW 2008)*, (pp. 327-336). Beijing, China.
- Smith, R. B., Hixon, R., & Horan, B. (1998). Supporting Flexible Roles in a Shared Space. *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, (pp. 197-206). Seattle, WA, USA.
- Spärck Jones, K. (1964). *Synonymy and Semantic Classification*. Cambridge, UK: PhD thesis, University of Cambridge.
- Specia, L., & Motta, E. (2007). Integrating Folksonomies with the Semantic Web. *Proceedings of the 4th European Web Semantic Conference (ESWC 2007)*, (pp. 1611-3349). Innsbruck, Austria.
- Stojanovic, N., Studer, R., & Stojanovic, L. (2003). An Approach for the Ranking of Query Results in the Semantic Web. *Proceedings of the 2nd International Semantic Web Conference (ISWC 2003)*, (pp. 500-516). Sanibel Island, FL, USA.
- Sujiyama, K., Hatano, K., & Yoshikawa, M. (2004). Adaptive Web Search Based On User Profile Constructed Without Any Effort From Users. *Proceedings of the 13th World Wide Web Conference (WWW 2004)*, (pp. 675-684). New York, NY, USA.
- Swets, J. (1988). Measuring the Accuracy of Diagnostic Systems. *Science*, 240, 1285-1293.
- Symeonidis, P., Nanopoulos, A., & Manolopoulos, Y. (2007). Feature-weighted User Model for Recommender Systems. *Proceedings of the 11th International Conference on User Modelling (UM 2007)*, (pp. 97-106). Corfu, Greece.
- Szomszor, M., Cattuto, C., Alani, H., O'Hara, K., Baldassarri, A., Loreto, V., et al. (2007). Folksonomies, the Semantic Web, and Movie Recommendation. *Proceedings of the ESWC 2007 Workshop Bridging the Gap between Semantic Web and Web 2.0*. Innsbruck, Austria.

- Taylor, A. D. (1995). *Mathematics and Politics: Strategy, Voting, Power and Proof*. New York, NY, USA: Springer-Verlag.
- Taylor, P. (2007). New tools to vie with Google. *Financial Times* (22nd March 2007) .
- Terveen, L., & Hill, W. (2001). Beyond Recommender Systems: Helping People Help Each Other. In J. M. Carroll, *Human-Computer Interaction in the New Millennium* (pp. 487-509). New York: Addison-Wesley.
- Tintarev, N., & Masthoff, J. (2007). A Survey of Explanations in Recommender Systems. *Proceedings of the 3rd International Workshop on Web Personalisation, Recommender Systems and Intelligent User Interfaces (WPRSIUI 2007)*, (pp. 801-810). Istanbul, Turkey.
- Tran, T., & Cohen, R. (2000). Hybrid Recommender Systems for Electronic Commerce. *Proceedings of the AAAI 2000 Workshop on Knowledge-Based Electronic Markets*, (pp. 78-84). Austin, TX, USA.
- Ungar, L. H., & Foster, D. P. (1998). Clustering Methods for Collaborative Filtering. *Proceedings of the AAAI 1998 Workshop on Recommendation Systems*. Madison, WI, USA.
- Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., et al. (2006). Semantic Annotation for Knowledge Management: Requirements and a Survey of the State of the Art. *Journal of Web Semantics*, 4 (1), 14-28.
- Uschold, M., & Grüninger, M. (1996). Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review*, 11 (2), 93-136.
- Vallet, D., Castells, P., Fernández, M., Mylonas, P., & Avrithis, Y. (2007). Personalized Content Retrieval in Context Using Ontological Knowledge. *IEEE Transactions on Circuits and Systems for Video Technology*, 17 (3), 336-346.
- Vallet, D., Fernández, M., & Castells, P. (2005). An Ontology-based Information Retrieval Model. *Proceedings of the 2nd European Semantic Web Conference (ESWC 2005)*. Heraklion, Greece: Springer Verlag Lecture Notes in Computer Science, 3532, pp. 455-470.
- Van Damme, C., Hepp, M., & Siorpaes, K. (2007). FolksOntology: An Integrated Approach for Turning Folksonomies into Ontologies. *Proceedings of the ESWC 2007 Workshop Bridging the Gap between Semantic Web and Web 2.0*, (pp. 57-70). Innsbruck, Austria.

- Van der Sluijs, K., & J, H. G. (2008). Relating User Tags to Ontological Information. *Proceedings of the 5th International Workshop on Ubiquitous User Modeling (UbiqUM 2008)*. Gran Canaria, Spain.
- Van Rijsbergen, C. J. (1979). *Information Retrieval*. London: Butterworth.
- Vorhees, E. (2004). Query Expansion using Lexical Semantic Relations. *Proceedings of the 17th ACM Conference on Research and Development in Information Retrieval (SIGIR 2004)*, (pp. 61-69). Dublin, Ireland.
- Vorhees, E. (2001). The TREC Question Answering Track. *Natural Language Engineering*, 7 (4), 361-378.
- Wasserman, S., & Faust, K. (1994). *Social Network Analysis: Methods and Applications*. Cambridge, UK: Cambridge University Press.
- Wenger, E. (1998). *Communities of Practice: Learning, Meaning and Identity*. Cambridge, UK: Cambridge University Press.
- Wenger, E. (2000). Communities of Practice: The Key to Knowledge Strategy. In E. L. Lesser, M. A. Fontaine, & J. A. Slusher, *Knowledge and Communities* (pp. 3-20). Boston, MA, USA: Butterworth-Heinemann.
- White, R. W., Ruthven, I., Jose, J. M., & Van Rijsbergen, C. J. (2005). Evaluating Implicit Feedback Models Using Searcher Simulations. *ACM Transactions on Information Systems*, 23 (3), 325-361.
- Yang, Y., & Chute, C. G. (1993). Words or Concepts: The Features of Indexing Units and their Optimal Use in Information Retrieval. *Proceedings of the 17th Annual Symposium on Computer Applications in Social Care*, (pp. 685-689). Washington, DC, USA.
- Yu, Z., Zhou, X., Hao, Y., & Gu, J. (2006). TV Program Recommendation for Multiple Viewers Based on user Profile Merging. *User Modeling and User-Adapted Interaction archive*, 16 (1), 63-82.
- Yu, Z., Zhou, X., Hao, Y., & Gu, J. (2004). User Profile Merging Based on Total Distance Minimization. *Proceedings of the 2nd International Conference on Smart Homes and Health Telematic (ICOST 2004)*, pp. 25-32. Singapore.
- Zhang, Y., Callan, J., & Minka, T. (2002). Novelty and Redundancy Detection in Adaptive Filtering. *Proceedings of the 25th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002)*, (pp. 81-88). Tampere, Finland.