l'université
nantes
angers
le mans
PÔLE DE RECHERCHE ET D'ENSEIGNEMENT SUPÉRIEUR

UNIVERSITÉ DE NANTES

Thèse de Doctorat

Diego TORRES

Mémoire présenté en vue de l'obtention du
grade de Docteur de l'Université de Nantes
sous le label de l'Université de Nantes Angers Le Mans

Discipline : Informatique
Spécialité : Informatique
Laboratoire : LIFIA - UNLP

Soutenue le 3 octobre 2014

École doctorale : 503 (STIM)
Thèse n° : 000000000

Co-Evolution between Social and Semantic Web

JURY

Rapporteurs :	M$^{\text{me}}$ **Anne Boyer**, Professor, Université de Lorraine (France) M. **Hernan Astudillo**, Professor, UTFSM (Chile)
Examinateurs :	M. **Guillermo Simari**, Professor, Universidad Nacional del Sur (Argentina) M. **Luis Antonio Olsina**, Professor, Universidad Nacional del La Pampa (Argentina)
Invité :	M. **Pablo Fillotranni**, Associate Professor, Universidad Nacional del Sur (Argentina)
Directeur de thèse :	M. **Pascal Molli**, Professeur, Université de Nantes
Co-directrice de thèse :	M$^{\text{me}}$ **Alicia Diaz**, Profeseur, Directeur etranger - UNLP - LIFIA - Argentina
Co-encadrante :	M$^{\text{me}}$ **Hala Skaf-Molli**, Maître de Conferences Université de Nantes (France)

Co-evolución entre la Web Social y la Web Semántica

Diego Torres

Facultad de Informática

Universidad Nacional de La Plata

Directora (Universidad Nacional de La Plata): Dra. Alicia Díaz

Director (Universidad de Nantes): Dr. Pascal Molli

Co-Directora (Universidad de Nantes): Dra. Hala Skaf-Molli

Tesis realizada en co-tutela para obtener el grado de

Doctor en Ciencias Informáticas de la Universidad Nacional de La Plata y Docteur de l'Université de Nantes

Abril de 2014

A Mariana,
a mi familia,
a los que tuvieron que exiliarse,
a la memoria de los desaparecidos,
y a los que seguimos pidiendo:
memoria, verdad y justicia.

Resumen

La Web Social y la Web Semántica han impactado en la forma en que la creación de conocimiento se ha llevado a cabo en la Web. La Web Social promociona la participación de los usuarios para crear y editar contenido y conocimiento en la Web. La proliferación de contenido y la necesidad de tener una administración automatizada de esta información disparó la aparición de la Web Semántica. Actualmente, la Web Social y la Web Semántica conviven y comparten un mismo tema: un mejor manejo del conocimiento. Sin embargo, la mayoría de la información en la Web Social no es parte de la Web Semántica, y la información de la Web Semántica no es utilizada para mejorar a la Web Social.

Esta tesis presenta un enfoque innovador para estimular una co-evolución entre la Web Semántica y la Web Social: las fuerzas que impulsan la Web Social y las herramientas que llevan a cabo la Web Semántica trabajando en conjunto con el fin de tener beneficios mutuos. En este trabajo afirmamos que la co-evolución entre la Web Social y la Web Semántica mejorará la generación de información semántica en la Web Semántica, y mejorará la producción de conocimiento en la Web Social.

Esto invita a responder las siguientes preguntas: ¿Cómo puede incluirse la generación de datos semánticos en las actividades de los usuarios de la Web Social? ¿Como puede definirse la semántica de un recurso web en un entorno social? ¿Cómo puede inyectarse en la Web Social las nuevas piezas de información extraídas de la Web Semántica? ¿Poseen las comunidades de la Web Social convenciones generales que deban ser respetadas?

Con el fin de mejorar la Web Semántica con las fuerzas de la Web Social, en este trabajo se proponen dos enfoques de *Social Semantic Tagging*: P-Swooki que permite a usuarios de una wiki semántica gestionar anotaciones semánticas permitiendo completar el proceso de construcción de conocimiento, y Semdrops que permite a los usuarios describir en forma semántica cualquier recurso de la Web tanto en un espacio de conocimiento personal como en un espacio compartido. Además, con el fin de mejorar el contenido de la Web Social, proponemos BlueFinder: un sistema de recomendación que detecta y recomienda la mejor manera de representar en un sitio de la Web Social, información que es extraída de la Web Semántica. En particular, BlueFinder recomienda la manera de representar una propiedad semántica de DBpedia en Wikipedia, respetando las convenciones de la comunidad de usuarios de Wikipedia.

Abstract

Social and Semantic Web has impacted in the manner of knowledge building is fulfill in the Web. The Social Web promoted the participation of users to create and edit Web content and knowledge. The content proliferation and the need to have a better machine management of such information trigger the Semantic Web. Currently, the Social and the Semantic Web are living together and they share a same topic: a better management of knowledge. However, most of the Social Web information is not part of the Semantic Web, and Semantic Web information is not used to improve the Social Web.

This thesis introduced an innovative approach to stimulate a co- evolution between the Semantic and Social Web: social and machine forces work together in order to have mutual benefits. We claim that having a co-evolution between Social and Semantic Web will improve the generation of semantic data and a knowledge production improvement in the Social Web.

This invite us to answer the following questions: How can the generation of semantic data be included in the activities of Social Web users? How can the semantics of a Web resource be defined in a social environment? How can new pieces of information be injected from the Semantic Web in the Social Web? Has the Social Web community general convention to be respected?

In this work, in order to improve the Semantic Web by means of Social Web, two Social Semantic Tagging approach are proposed: P-Swooki allows P2P semantic wiki users to manage personal semantic annotations in order to complete the knowledge building process, and Semdrops allows users to semantically describe any web resources in both personal and shared knowledge spaces. In order to improve the Social Web content, we also propose BlueFinder: a recommender system that detects and recommends the best manner to represent in a Social Web site information that is extracted from the Semantic Web. In particular, BlueFinder recommends the manner to represents a DBpedia Semantic property in Wikipedia following the Wikipedia community conventions.

Résumé

Le Web social et le Web sémantique ont eu un impact sur la façon dont la création de connaissances est effectuée sur le Web. Le Web social favorise la participation des utilisateurs à créer et modifier des contenus et des connaissances sur le Web. La prolifération de contenu et la nécessité d'une gestion automatisé de l'information a déclenché l'émergence du Web sémantique. Actuellement, le Web social et le Web sémantique coexistent et partagent un thème commun : une meilleure gestion de la connaissance. Cependant, la plupart des informations sur le Web social ne fait pas partie du Web sémantique et l'information du Web sémantique n'est pas utilisée pour améliorer le Web social.

Cette thèse présente une approche innovante pour stimuler la co-évolution entre le Web sémantique et le Web social : les utilisateurs et les ordinateurs qui travaillent ensemble afin d'obtenir des avantages mutuels. Nous affirmons que la co-évolution du Web social et du Web sémantique permettra, d'une part d'améliorer la production de l'information sémantique au Web sémantique, et d'une autre part d'améliorer la production de connaissances sur le Web social.

Cela exige de répondre aux questions suivantes : Comment peut-on inclure la production de données sémantiques sur les activités des utilisateurs du Web social ? Comment peut-on définir la sémantique d'une ressource Web dans un environnement social ? Comment le Web social peut injecter de nouveaux éléments d'information extraits du Web sémantique ? Est-ce que la communauté de Web social a des conventions générales qui doivent être respectées ?

Dans ce travail, afin d'améliorer le Web sémantique en utilisant l'information disponible de Web social, deux approches sémantique sociale sont pro-

posées : P- Swooki qui permet aux utilisateurs de gérer un wiki sémantique avec des annotations sémantiques personnelle pour compléter le processus de construction de connaissances, et Semdrops qui permet aux utilisateurs de décrire sémantiquement une ressource web tout à la fois dans un espace de connaissance personnelle et un espace partagé. En outre, afin d'améliorer le contenu du Web social, nous proposons BlueFinder : un système de recommandation qui détecte et recommande la meilleure façon de représenter l'information extraite du Web sémantique dans un site Web social. En particulier, BlueFinder recommande la façon de représenter une propriété sémantique de DBpedia dans Wikipedia, en respectant les conventions de la communauté de Wikipedia.

Agradecimientos

En primer lugar quiero agradecer a mis directores del doctorado: Alicia Díaz, Hala Skaf-Molli y Pascal Molli. Antes de comenzar el doctorado, Alicia me abrió las puertas a la investigación e hizo que me sintiera capaz de seguir en esto que tanto me gusta. Alicia: gracias por ese don único que tenés de decir las palabras justas, transmitir confianza y ser una directora y una persona maravillosa. Muchas gracias a Hala y Pascal por el apoyo y la dirección, por la ayuda y consejos en la escritura de artículos, por el esfuerzo extra que ocasionan las diferencias horarias entre Argentina y Francia, y por la calidez durante mis estadías en Nancy y Nantes.

Quiero agradecer al LIFIA por darme todo y más cada vez que necesito algo. Gracias a Gustavo Rossi, Silvia Gordillo, Gabriel Baum y Alicia Diaz por hacer de LIFIA un laboratorio donde los lazos personales son importantes para construir y para crecer. A mis amigos Julián G, Matías U., Juan Ignacio y Sergio. Hago extensivo el agradecimiento al resto de mis compañeros de LIFIA. A Francisco Blanco, que me ayudó en la programación y puesta en evaluación de varios prototipos utilizados en el trabajo de tesis.

También agradezco a las organizaciones que posibilitaron la realización de los viajes y estadías durante mi doctorado. Muchas gracias a la Universidad Nacional de La Plata y a la Facultad de Informática por ayudarme en este proceso, a la UNLP por los subsidios que posibilitaron los viajes y las estadías. También agradezco a la Universidad de Nantes, al laboratorio LINA, a Pierre Cointe y al staff del grupo GDD en Nantes. A Jean-Yves Leblin que me ayudo mucho con las evaluaciones en Gally. Han sido muy amables durante todas

mis estancias en Nantes. Gracias también por las ayudas económicas durante mis estancias allí y a la Maison des Chercheurs Etrangers en Nantes. También quiero agradecer al Ministerio de Ciencia y Técnica de Argentina que financió mediante proyectos parte de las estadías y viajes en el exterior. Gracias al gobierno nacional que permitió que el desarrollo de la investigación sea posible en esta ultima década.

Durante las estadías en Nantes, el apoyo afectivo fue muy grande. Quiero agradecerles fuertemente a Khaled y Nagham (y también a la pequeña Natalie) por tanto cariño que nos han dado, por las ayudas con el idioma y las implicancias de estar en otro país. También a Luis, Raziel, Gabriela, Amir y James. A Luis Peñaranda y Flavia Felicioni por la ayuda y el apoyo durante mis estadias en Nancy.

A mis compañeros de cátedras en La Plata y a los jefes de trabajos prácticos Leandro, Arturo, Federico y Andrés por encontrar la forma de subsanar mis ausencias durante mis viajes a Francia.

A mis compañeros de TPI en la Universidad Nacional de Quilmes. A Pablo "Fidel"Martínez López por ayudarme durante mis licencias y darme apoyo. A Eduardo Bonelli y Gabriela Arévalo por las charlas y consejos. A mi gran compañero de cátedra Ariel Miglio que me cubrió las espaldas muchas veces durante mis ausencias y solo recibí de su parte comentarios positivos y de aliento. A los alumnos.

A Maria Lira por sus clases de francés. A Florencia Capria y Andrea Asenti del Rio por las correcciones en inglés.

A mis amigos, que me ayudaron mucho durante todo este proceso: Lanto, Leandro Quiroga, Diego De Sogos, Miguel Lemus, Joaquin Alem, Emiliano Schmid, Juan Enriqué, Fede Naso y Andres Fortier. Les digo gracias con un abrazo y nos vemos en las cenas de los martes. Gracias a los chicos de teatro, que me permitieron darme un respiro y relajarme durante las clases y ensayos: Ivan, Casper, Nati, René, Clara, Nahuel y Nadia. Gracias a Infonoise!

Gracias a los que se atrevieron a leer los borradores de la tesis y realizar devoluciones y correcciones: Arturo Zambrano, Diego De Sogos, Leandro Antonelli, Julián Grigera y Sergio Firmenich.

A Keko, porque discutimos apasionadamente y soñamos en grande.

Desde los más profundo de mi corazón quiero agradecer a toda mi familia. Todos demostraron siempre su apoyo, su paciencia y su cariño. Realmente gracias a todos ustedes: los Rocca, Picciola, del Mármol, Cambiaggio y Weisse. A mi hermano de la vida Emiliano Schmid y su familia: Nati, Alejo -mi ahijado - y Sofi. A mi sobrina Ana Julia. A mis padres y mi hermana, no solo por el apoyo en este último tiempo, sino por mostrarme que el amor construye, siempre.

Por último, a Mariana del Mármol. Gracias por acompañarme. Gracias por enseñarme a vivir en plena felicidad.

Aknowledgements

I want to thank my PhD directors: Alicia Diaz, Hala Skaf-Molli and Pascal Molli. Before starting the PhD, Alicia opened for me the doors of the research world and she made it possible for me to do this activity I love. Alicia: Thank you for that unique gift you have to say the right words, to convey confidence and to be a director and a wonderful person. Thank you very much Hala and Pascal for the support and direction, for your help and advice in writing articles, for the extra effort entailed by the time difference between Argentina and France, and for all your warmth during my stay in Nancy and Nantes.

I would like to thank LIFIA for giving me everything and even whenever I needed something. I want to thank Gustavo Rossi, Silvia Gordillo, Gabriel Baum and Alicia Diaz because they make LIFIA a research laboratory where personal ties are important to build and grow. To my friends Julian Grigera, Matias Urbieta, Ignacio Vidal y Sergio Firmenich. I extend gratitude to the rest of the LIFIA staff. Thanks to Francisco Blanco, who helped me a lot in the development and evaluation of several prototypes used in this thesis.

I also thank the organizations that made travel and stay during my PhD possible. Thank you very much to Universidad Nacional de La Plata (UNLP) and Facultad de Informática for helping me in this process. I am grateful to UNLP for all the subsidies that allowed me to travel and stay. I also thank the University of Nantes, LINA laboratory, Pierre Cointe and the staff of the GDD group at Nantes. To Jean-Yves Leblin who helped me with the evaluations at Gally. Everyone was so helpful during all my stays at Nantes. Thanks also for the financial support during my stay there, and to the Maison des

Chercheurs Etrangers in Nantes. Thank to the Ministry of Science and Technology of Argentina who financed through projects part of the travel and stay abroad. Thanks to the national government who made the development of research possible in this last decade.

During my stay in Nantes, emotional support was great. I want to strongly thank Khaled and Nagham (and petite Natalie) for the affection they gave us, their help with language and culture shock. Also to Luis, Raziel, Gabriela, Liza, Amir and James. To Luis Peñaranda and Flavia Felicioni for their help and support during my stays in Nancy.

To my co-chairs in Objetos I and II in La Plata, and specially to Leandro, Arturo, Federico and Andrés to find a way to make up for my absences during my trip to France.

To my fellows in TPI at the Universidad Nacional de Quilmes. Thanks Pablo "Fidel"Martínez López for helping me during my leaves and support. Thanks Eduardo Bonelli and Gabriela Arévalo for talks and advice. To my great co-chair Ariel Miglio who covered my back many times during my absences and only have gave me positive comments and encouragement. To my students.

Thanks Maria Lira for the French lessons. To Florencia Capria and Andrea Assenti del Rio for the English lessons and corrections.

To my friends who helped me throughout this process: Leandro Quiroga, Leandro Antonelli, Diego De Sogos, Miguel Lemus, Joaquin Alem, Emiliano Schmid, Federico Naso and Andres Fortier. Dear friends, see you at dinner on Tuesdays. Thanks to the guys from drama classes who allowed me to find relaxation and breaks through lessons and exercises: Ivan, Casper, Nati, René, Clara, Nahuel and Nadia. Thanks also to Infonoise!

Thanks to those who dared to read drafts of the thesis and provided feedback and corrections: Arturo Zambrano, Diego De Sogos, Leandro Antonelli, Julian Grigera and Sergio Firmenich.

To Keko because we passionately discussed and dreamed big.

From the bottom of my heart I want to thank all my family. Everyone always showed their support, patience and love. Really thank you all: Rocca, Picciola, del Mármol, Cambiaggio and Weisse. To my "brother of life", Emiliano Schmid and his family: Nati, Alejo-my godson - and Sofi. To my niece Ana Julia. To my parents and my sister, not only for the support in recent times, but for showing that love can be built, always.

Finally, to Mariana del Mármol. Thanks you for standing by me. Thanks for teaching me to live in complete happiness.

Contents

I Introduction 9

1 Introduction 11
- 1.1 Co-Evolution Problem . . . 13
- 1.2 Topic and Hypothesis . . . 14
- 1.3 Contributions of this Thesis . . . 15
- 1.4 Publications . . . 16
  - 1.4.1 From Social Web to Semantic Web . . . 16
  - 1.4.2 From Semantic Web to Social Web . . . 16
- 1.5 Outline of this Thesis . . . 16

II From Social Web to Semantic Web 19

2 Introduction 21
- 2.1 Outline of the Second Part . . . 22

3 Background 25
- 3.1 Collaborative Knowledge Building . . . 25
  - 3.1.1 Tools for Collaborative Knowledge Building Systems for Semantic Data . . . 27
- 3.2 Semantic Wikis . . . 27
- 3.3 Social Tagging . . . 30
  - 3.3.1 Gruber's Model . . . 31
- 3.4 Efforts in Semantic Data Generation . . . 32

4 P-Swooki 35
- 4.1 P2P Collaborative Knowledge Building Approach . . . 36
  - 4.1.1 P2P Collaborative Knowledge Building Process . . . 37
  - 4.1.2 Personal Semantic Annotations: Individuals and Categories . . . 38
- 4.2 P-Swooki: P2P Collaborative Knowledge Building System . . . 39
  - 4.2.1 Shared Semantic Annotation Management . . . 39
  - 4.2.2 Personal Semantic Annotations Management . . . 39
  - 4.2.3 P-Swooki Data Model . . . 40
  - 4.2.4 Personal Semantic Annotation Storage Model . . . 40
  - 4.2.5 Editing Operations . . . 41
- 4.3 P-Swooki Implementation . . . 41

5 Semdrops 43
5.1 Semdrops Approach . . . 44
5.1.1 Semantic Social Tagging Model . . . 44
5.1.2 Using Semantic Mediawiki as Semantic Support . . . 45
5.1.3 How to Use the Model ? . . . 46
5.2 Implementation . . . 47

6 P-Swooki and Semdrops Evaluation 51
6.1 P-Swooki Evaluation . . . 51
6.1.1 Method . . . 52
6.1.2 Results . . . 52
6.1.3 Discussion and Learned Lessons . . . 53
6.2 Semdrops Evaluation . . . 55
6.2.1 Method . . . 55
6.2.2 Results . . . 56
6.2.3 Usability Evaluation . . . 58
6.2.4 Discussion and Learned Lessons . . . 58
6.3 Summary of this Part . . . 59

III From Semantic Web to Social Web 61

7 Introduction 63
7.1 Outline of the Third Part . . . 65

8 Background 67
8.1 Social Web: Wikipedia . . . 68
8.1.1 Articles . . . 68
8.1.2 Categories . . . 69
8.1.3 Wikipedia Conventions . . . 72
8.2 Semantic Web: DBpedia . . . 73
8.2.1 Data extraction . . . 73
8.2.2 Data Query . . . 75
8.3 Collaborative Recommender Systems . . . 75
8.3.1 Implicit and Explicit Ratings . . . 77
8.3.2 Similarity Functions . . . 77
8.3.3 Number of Neighbors . . . 78
8.4 Improving Social Web with Semantic Web . . . 78
8.4.1 General Approaches . . . 78
8.4.2 Adding missing links in Wikipedia . . . 80

9 A Semantic Information Gap Between DBpedia and Wikipedia 83

10 BlueFinder 87
10.1 Formal Definitions . . . 88
10.1.1 DBpedia Model . . . 88
10.1.2 Wikipedia Model . . . 89
10.2 Wikipedia Pairs Connection . . . 90
10.3 Problem Statement . . . 91
10.4 BlueFinder Recommender System . . . 91

10.4.1 Indexing . . . 92
10.4.2 Recommendation . . . 92
10.4.3 BlueFinder Algorithm . . . 93

11 BlueFinder Evaluation 99
11.1 BlueFinder General Experimentation . . . 99
11.1.1 Evaluation metrics . . . 100
11.1.2 Results and discussion . . . 103
11.2 Adapting BlueFinder to French and Spanish Wikipedia . . . 113
11.2.1 Experimentation . . . 114
11.3 Summary of this Part . . . 117

IV Conclusions and Perspective 119

12 Conclusions 121

13 Perspectives 127
13.0.1 Semdrops Improvements . . . 127
13.0.2 BlueFinder improvements . . . 128
13.0.3 Cross Language Combinations . . . 128
13.0.4 Evolution of the Conventions . . . 129
13.0.5 Semdrops and BlueFinder Mash-up . . . 130

A Appendix 131
A.1 Social Evaluation . . . 131
A.1.1 Method . . . 131
A.1.2 Results and Discussion . . . 132
A.2 Tables . . . 133

List of Figures

1.1 Co-evolution between Social and Semantic Web Forces 14

3.1 Stahl's Collaborative Knowledge Building Process 26
3.2 Co-Protégé: The screenshot shows the personal and the shared ontology edition spaces 28
3.3 OntoEdit 29
3.4 SMW mark-up enhanced for semantic annotations. 30
3.5 SMW inline query result. 31
3.6 Delicious.com tags folksonomy. 32

4.1 Adding Personal Semantic Annotation in Semantic Wiki 37
4.2 P2P Collaborative Knowledge Building Process 38
4.3 Personal Semantic Annotation Data Model 40
4.4 P-Swooki Architecture 42
4.5 P-Swooki Interface 42

5.1 Tagging Flow 46
5.2 Semantic Tags for Wikipedia's Paris page: (a) At the top, the Wikipedia's Paris page without the Social Semantic Tags. (b) At the bottom, *Semdrops* in action for the same Wikipedia page. 48
5.3 Semdrops Architecture 49

6.1 Quantities of semantic Tags 56
6.2 Quality of Semantic Tags 57
6.3 Proportion of Semantic Tags 57

8.1 Main structure of a Wikipedia article: Lionel Messi 69
8.2 Infobox detail 70
8.3 Category in Wikipedia: Rosario, Santa Fe 71
8.4 Example of Wikipedia Graph. Part of the links between Rosario, Santa Fe and Lionel Messi 72
8.5 Example of Wikipedia Infobox Template and its rendered output. 73

9.1 Gap proportion for the twenty DBpedia properties of Table 9.1 . 85
9.2 Information flow between Social Web and Semantic Web 86

10.1 BlueFinder data flow 94

11.1 Evaluation method 102
11.2 Presicion, Recall, F1, and Hit-rate mean of all properties 104
11.3 Precision all properties 105
11.4 Recall for all properties 107

11.5 F1 for all properties 108
11.6 Hit-rate for all properties 109
11.7 Confidence and hit-rate ratio according to prediction order in the recommendation set 110
11.8 BlueFinder evaluation in the Spanish Wikipedia. 116
11.9 BlueFinder evaluation in the French Wikipedia. 117

List of Tables

5.1 Semdrops Mapping Rules 46

6.1 Personal Semantic Annotation for *Semantic Wiki* Page 52
6.2 Personal Semantic Annotation for *Semantic Web* Page 53
6.3 Personal and Shared Semantic Annotation for Semantic Wiki Page 54

9.1 Results of 20 SPARQL queries for 20 properties and different resources types. 84

11.1 Confidence and hit-rate ratio for High Confidence predictions . . 110
11.2 Gini index of the properties 112
11.3 Gini index of the properties $prop_9$: `debutstadium` and $prop_{11}$: `training` with $maxRecom$=5 113
11.4 Path Index bipartite graphs used in the experimentation. Columns show the number of path queries, the number of connected pairs, the number of edges and the number of elements in the Data set 115
11.5 Detail of BlueFinder evaluation in the Spanish Wikipedia. . . . 115
11.6 Details of BlueFinder Evaluation in the French Wikipedia. . . . 116

A.1 Community Evaluation results. 13 of these contributions were made by Wikipedia users who respect the most representative path queries for Q_2 131
A.2 Precision with maxRecom=1 134
A.3 Precision with maxRecom=3 135
A.4 Precision with maxRecom=5 136
A.5 Precision with maxRecom=unlimited 137
A.6 Recall with maxRecom=1 138
A.7 Recall with maxRecom=3 139
A.8 Recall with maxRecom=5 140
A.9 Recall with maxRecom=unlimited 141
A.10 F1 with maxRecom=1 142
A.11 F1 with maxRecom=3 143
A.12 F1 with maxRecom=5 144
A.13 F1 with maxRecom=unlimited 145
A.14 Hit-rate with maxRecom=1 146
A.15 F1 with maxRecom=unlimited 147
A.16 Hit-rate with maxRecom=5 148
A.17 Hit-rate with maxRecom=unlimited 149

I

Introduction

1

Introduction

Contents

1.1 Co-Evolution Problem 13

1.2 Topic and Hypothesis 14

1.3 Contributions of this Thesis 15

1.4 Publications 16

1.4.1 From Social Web to Semantic Web 16

1.4.2 From Semantic Web to Social Web 16

1.5 Outline of this Thesis 16

Social and Semantic Web has impacted in the manner of knowledge building is fulfill in the Web. In the first place, the Social Web promoted the participation of users to create and edit Web content and knowledge. According to Gruber, the Social Web *"is represented by a class of web sites and applications in which user participation is the primary driver of value"* [32]. Furthermore, content creation is enhanced by means of enabling social support as this generates a proliferation of content on the Web in a collaborative context [58]. Flickr[1], Facebook[2], Wikipedia[3], Bibsonomy[4], Blogger[5], Wordpress[6] and Tumblr[7] are examples of Web sites which enable users to create new pieces of knowledge and information. In the end, the Semantic Web challenge is the generation of documents that are able to be processed by computers. The content proliferation and the need to have a better management of such information triggers the Semantic Web. According to Tim Berners-Lee, *"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work*

[1] http://www.flickr.com

[2] http://www.facebook.com

[3] http://www.wikipedia.org

[4] http://www.bibsonomy.org

[5] http://www.blogger.com

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

[7] http://www.tumblr.com

in cooperation" [8]. Using ontologies, the backbone of the Semantic Web, is the manner to have well-defined the meaning. The potential of the Semantic Web allows computers to derive information from collections of documents. This enables users to have, for example, better results in document search and interoperability among systems. Freebase [13], DBpedia [12], and Yago [85] are examples of Semantic Web projects.

Currently, the Social and the Semantic Web are living together and they share a same topic: a better management of knowledge. The amount of documents which are generated by Social Web users needs computer tools in order to have a better processing of the information. Besides, the Semantic Web has to integrate the Social Web information in order to improve the search and query capabilities over socially generated content. Indeed, several elements from Social Web are promising to be integrated with the Semantic Web, and the same happens in the other direction. Both Social and Semantic Web could have an improvement by a complementary interaction. This interaction is described by Engelbart as *co-evolution*: *"The elements involved in augmenting communities of knowledge workers include the development of both 'tool systems' and 'human systems',.., the co-evolution occurs between the tools and the people using them."*[25]. Additionally, Engelbart includes the idea of bootstrapping methods, pieces of knowledge, and tools in order to have a capacity to generate further tools which could be used to enrich the human activities. These ideas could be mapped to the challenging Social and Semantic Web coevolution, where the human systems are represented by the social forces from Social Web; and the tool systems (ontology definitions, semantic reasoners and the systems interoperability) are the elements of the Semantic Web.

In a context of coevolution between Social and Semantic Web, bootstrapping the Social Web knowledge building process and the semantic-web tools is a challenge. Social Web forces generate new pieces of knowledge, and most of them are not part of the Semantic Web. Hence, the new knowledge could be used to *"augment"* the Semantic Web knowledge bases. The augmentation process requires adjusting the knowledge representation and method used in the Social Web to the machine-enabled representation in the Semantic Web. Consequently, the reasoning tools are applied to the *augmented* Semantic Web in order to derive new pieces of knowledge. Semantic Web query systems allow users to derive information that is complex to obtain by only using Social Web tools. However, the new pieces of knowledge are not only important to the Semantic Web: injecting the new pieces of knowledge into the Social Web will trigger and reinforce the social activities to create other knowledge elements. The injection process requires to bootstrap, again, the knowledge representation and methods from the Social Web in order to translate the new Semantic Web information pieces into a social-enabled language that respect the social conventions.

Unfortunately, several current works in Semantic Web and Social Web are only coexisting instead of *co-evolving*. Most of the Social Web information is not part of the Semantic Web, and Semantic Web information is not injected in terms of social conventions into the Social Web. Indeed, tools from the Semantic Web are not bootstrapped in order to have a better work of human in the Social Web. Current works are in line with the Social Semantic Web concept where *"socially created and shared knowledge leads to the creation of explicit* *and semantically-rich knowledge representations"* [88]. In this direction, using the Semantic Web technologies to model social data (e.g FOAF which models a network of friends), and using the wisdom of crowds to generate semantic data are the main topics. Although they are related to the idea of co-evolution, most of these works are either focused in the generation of semantic data in a particular field by applying rigid ontology definitions (e.g. [101]), or are limited for a specific domain like bookmarks (e.g. [40, 62, 101]), or are mapping of semi-structured information to a semantic repository (e.g. DBpedia). Better approaches are Semantic Wikis which combines a social activity with semantic technologies in a whole system. However, they do not allow users to combine personal conceptualizations with the shared ones, or to semantically describe elements that are out of the wiki system. Finally, the use of Semantic Web to improve the Social Web is not as well developed as from the social activities to the Semantic Web. The most commons approaches combine the search facilities from the Semantic Web with a social network in order to make recommendations to a user based on the network of friends e.g. recommending movies with a Facebook profile.

1.1 Co-Evolution Problem

Having this context in mind, a design of a real co-evolution between Social and Semantic Web is a challenge. As we have seen, there are many factors involved in the relation among the social and semantic forces. Figure 1.1 shows a schema of the Social and Semantic Web that are related by two arrows. The elements to bootstrap the human, methods and tools from both sides of the Figure 1.1 could be placed where the arrows are. Although there are many approaches to develop a co-evolution between Social and Semantic Web, two scenarios will be exploited in this dissertation. Whereas the co-evolution involves a whole study of the relations between Semantic and Social Web, in order to have an easy analysis of the problem we split it into two parts: (1) the use of the Social Web to improve the Semantic Web, and (2) the use of the Semantic Web to improve the Social Web.

For the first case (1), we improve the Semantic Web by means of social forces. In this sense, we have to take into account the methods, the knowledge pieces and the manner to share information. The generation of knowledge in the social context involves a complex collaborative knowledge building process [84]. Personal and shared understanding are articulated in this process. People incorporate into their personal beliefs new knowledge pieces from their social context, and also people externalize personal beliefs to a social discussion. This articulation process requires methods to incorporate and externalize pieces of knowledge and could generate either contradictions, consensus, and new pieces of knowledge. In the co-evolution scenario, these mechanisms have to be bootstrapped in specific tools. Therefore, any tool that supports the collaborative knowledge building process must provide artifacts to articulate both personal and shared knowledge. Additionally, in order to publish the new knowledge pieces in the Semantic Web, an ontology definition have to be provided. The ontology has to be flexible in order to allow users to describe an underlying ontology definition. In conclusion, the social activities and methods will be bootstrapped in order to generate and manipulate Semantic Web enabled content.

On the other side (2), the Semantic Web includes the tools and pieces of knowledge to improve the Social Web. At the first glance the idea sounds quite simple, the Semantic Web provides information interoperability and query facilities to derive new pieces of information that are complex to be performed in the Social Web context i.e a complex query that combines several sources of information. However, the challenge of this part includes to understand if the new pieces are relevant for the community on the Social Web, and to adequate the new pieces of knowledge into the social process. Hence, the mapping of semantic relationships from semantic repositories with elements from the social documents is another important issue. For example, a semantic property could be related to a hyperlink between two web documents, or even it could be more complex as a string of links among several web documents. One strategy could be learning from the Social Web the representation of the pieces of knowledge that also appears in the Semantic Web, and after that discover the manner to represent the new pieces of knowledge in terms of social conventions. There are several variables which are involved in the definition of social conventions such as the number of users, the specificity of the subject or the relevance of the created document.

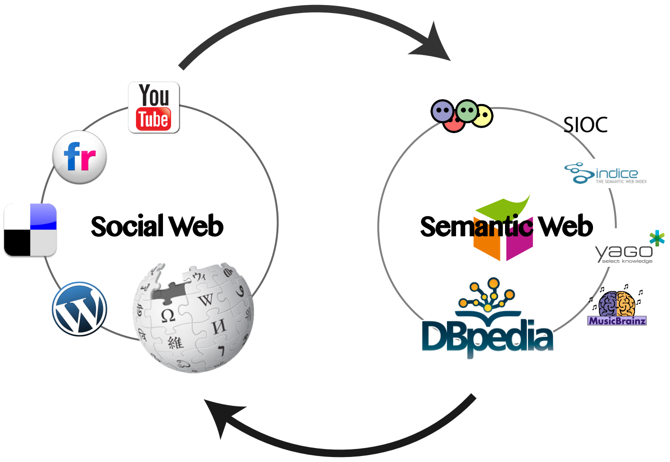


Figure 1.1: Co-evolution between Social and Semantic Web Forces

1.2 Topic and Hypothesis

In this thesis, we are interested in stimulate the virtuous co-evolution cycle of information between the Social and the Semantic Web. We claim that having a co-evolution between Social and Semantic Web will improve the generation of semantic data and a knowledge production improvement in the Social Web. This hypothesis could be divided into two more specific: (1) Bootstrapping the collaborative knowledge production and a semantic tagging activity will improve the semantic description of web content, (2) Bootstrapping the machine support to obtain new pieces of knowledge from the Semantic Web, social methods and conventions will improve the information in the Social Web.

To answer the first hypothesis, we focus on the generation of specific ontologies in order to generate the semantic data which is necessary to realize the

Semantic Web. This is exemplified by the arrow from Social Web to Semantic Web in Figure 1.1. To generate this semantic data we need to answer the following questions: *How can the generation of semantic data be included in the activities of Social Web users? How can the semantics of a Web resource be defined in a social environment? Do Social Web regular users have enough skills to describe Semantic Web enabled content?*. This issues are developed in Part II of this dissertation.

On the other hand, the Social Web could be improved by information obtained from the Semantic Web. Semantic Web technologies are mainly used to have better navigation and search results. Unfortunately, the Semantic Web is not used to insert new pieces of knowledge in a Social Web manner. The injection of new pieces of knowledge will close the co-evolution virtuous cycle. However, the Social Web involves social activities and conventions which have to be respected. Therefore, *How can new pieces of information be injected from the Semantic Web in the Social Web? Has the Social Web community general conventions to be respected? How Social Web conventions can be detected in an automatic manner?* . We focus on the information flow from the Semantic Web to the Social Web with the cases of DBpedia and Wikipedia. This issues are developed in Part III of this document.

1.3 Contributions of this Thesis

The contributions of this thesis are:

A Model to allow users to add semantic information to any web resource by means of social forces: Users describe the semantic data of any web resource in a collaborative knowledge building process. The process integrates both shared and personal knowledge spaces with the use of Semdrops. Semdrops allows users to generate a semantic definition of the web resources, specially those resources that are not currently part of the Semantic Web. In this thesis we introduce two complementary approaches: P-Swooki and Semdrops. P-Swooki allows P2P semantic wiki users to manage personal semantic annotations in order to complete the knowledge building process, and Semdrops allows users to semantically describe any web resources in both personal and shared knowledge spaces.

The BlueFinder algorithm that detects and recommends the best Wikipedia convention to represent a DBpedia semantic property in Wikipedia. BlueFinder uses the Semantic Web query capabilities to detect information that is not part of Wikipedia. BlueFinder learns from Wikipedia community the conventions that are used to represent relations among articles. After that, BlueFinder uses the learned information to represent a semantic property following the Wikipedia conventions. Then, if a pair of Wikipedia articles are related in DBpedia with a semantic property but they are not in Wikipedia, BlueFinder recommends the best manner to relate the couple of articles in Wikipedia by navigational paths. These new navigational paths improve Wikipedia content by means of Semantic Web information and wikipedia community method detection.

1.4 Publications

This thesis is based on the following publications:

1.4.1 From Social Web to Semantic Web

- Diego Torres, Hala Skaf-Molli, Alicia Diaz And Pascal Molli. Personal and Shared Knowledge Building in P2P Semantic Wikis. *In 6th European Semantic Web Conference (ESWC 2009) (Poster)*, Heraklion, Greece, May 2009.

- Diego Torres, Hala Skaf-Molli, Alicia Diaz And Pascal Molli. Personal Navigation in Semantic Wikis. In *International Workshop on Adaptation and Personalization for Web 2.0 in connection with UMAP'09* , Trento, Italy , June 2009

- Diego Torres, Hala Skaf-Molli, Alicia Diaz And Pascal Molli. Supporting Personal Semantic Annotations in P2P Semantic Wikis. In *DEXA'09:20th International Conference on Database and Expert Systems Applications* , Linz, Austria , August 2009.

- Diego Torres, Alicia Diaz, Hala Skaf-Molli and Pascal Molli. Semdrops: A Social Semantic Tagging Approach for Emerging Semantic Data. *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference* on. Volume 1. pp 340-347. 2011.

1.4.2 From Semantic Web to Social Web

- Diego Torres, Pascal Molli, Hala Skaf-Molli, and Alicia Diaz. From dbpedia to wikipedia: Filling the gap by discovering wikipedia conventions. In *2012 IEEE/WIC/ACM International Conference on Web Intelligence (WI'12)*, 2012.

- Diego Torres, Pascal Molli, Hala Skaf-Molli and Alicia Diaz. Improving wikipedia with DBpedia. *Proceedings of the 21st international conference companion on World Wide Web*. ACM, 1109-1112, Lyon, France, 2012.

- Diego Torres, Hala Skaf-Molli, Pascal Molli and Alicia Diaz. BlueFinder: Recommending Wikipedia Links Using DBpedia Properties. *ACM Web Science Conference 2013 (WebSci '13)*. Paris, France. May 2013.

- Diego Torres, Alicia Diaz, Hala Skaf-Molli and Pascal Molli. Recommending non-English Wikipedia Links Using DBpedia Properties. *SWCS2013 at ESWC2013 Workshop on Semantic Web Collaborative Spaces (SWCS2013)*. Montpellier, France. May, 2013.

1.5 Outline of this Thesis

This thesis is organized in the following three parts. Part II focus on the study of using the forces from the Social Web in order to improve the content of the Semantic Web. The first hypothesis is developed in this Part. We exploit the collaborative knowledge building process introduced by Sthal [84] in Social Web systems like Wikis and Semantic Wikis. This Part begins with the inclusion of personal knowledge spaces in a P2P semantic wiki which enables users to manage personal navigation and personal semantic definitions. After that contribution, we introduce a more general approach called Semdrops that defines personal and shared spaces to semantically annotate any web resource. Semdrops uses social forces to generate semantic data. This Part includes an evaluation that supports the results of these approaches.

Part III develops the work to improve the Social Web with the information derived from the Semantic Web. This part describes the study of using semantic information from DBpedia to improve Wikipedia. DBpedia and Wikipedia are the most relevant examples of Semantic and Social Web systems respectively. In this Part, we introduce BlueFinder, a collaborative filtering system that detects the best representation of a DBpedia semantic property in Wikipedia. BlueFinder improves Wikipedia with information from DBpedia. However, the main challenge of BlueFinder is to detect conventions that are in line with the Wikipedia community. At the end, an exhaustive evaluation of BlueFinder over twenty semantic properties is introduced. The evaluation showed that BlueFinder is accurate to make the recommendations and is a promising tool to improve Wikipedia content. Finally, we introduced an early experimentation of the use of BlueFinder to detect conventions among non-English versions of Wikipedia.

Finally, Part IV details the conclusions of the whole dissertation, the research lines that can be followed from the work presented in this thesis.

II

From Social Web to Semantic Web

2

Introduction

This Part of the thesis is focus in the first specific hypothesis: *The bootstrapping of the collaborative building knowledge process with a semantic tagging activity will improve the semantic description of web content.* In this Part we are going to introduce a model of articulation among elements from Social and Semantic Web in order to improve the Semantic Web content with new pieces of knowledge that are generated in the Social Web.

In order to have a co-evolution scenario we bootstrapped social knowledge building activities, an Semantic Web ontology representation, and a social tagging activity. The generation of knowledge in the Social Web is immerse into a collaborative knowledge building process, in this work we study the one that was introduced by Stahl [84]. It is basically a spiral process where knowledge first emerges at individual contexts and then it is socialized [64, 84]. Firstly, in co-evolution terms, the Stahl process is detected as the general social context that defines the interaction methods: people interacts by means of exchanging pieces of knowledge between their own personal knowledge space and the shared one. Secondly, a semantic ontology definition is developed to represent the data in the knowledge base information. Because the combination and definition of them are easier than large and sophisticated ones [36, 82], we base our approach on the use of lightweight ontologies. In consequence, all the knowledge that is included in personal or shared knowledge space is Semantic Web enabled content. Finally, a social tagging activity is incorporated in order to articulate the social knowledge process with the Semantic Web information description. The tagging activity is the nexus among the ontology, the personal and shared knowledge spaces, people, and the Semantic Web.

In this part of the thesis we present two complementary approaches in order to improve the Semantic Web content by means of social forces: *P-Swooki* and *Semdrops*. *P-Swooki* is P2P semantic Wiki extension that enables the management of personal semantic annotations in order to complete the knowledge building process. In addition, we define Semdrops, a simple semantic social tagging model. This model is integrated into a system to provide an easy interaction modality that enables users to attach tags to any web resource without modifying the resource itself. *Semdrops* defines a conceptual model which is an extension of the Gruber tag model [31], where the tag concept is extended to semantic tag. We implement *Semdrops* as a Firefox add-on tool that turns the web browser into a collaborative semantic data editor *in situ* for any web resource.

The main contribution of this part of the dissertation is *to allow users to add semantic information to any web resource by means of social forces.* In detail, the contributions are:

1. **A model to manage personal and shared knowledge in P2P semantic wikis**: The model defines personal semantic annotations in the context of a Semantic Wiki. Categories and Individuals can be defined in the personal user space. The personal space complements the shared one. This model was incorporated into a P2P Semantic Wiki system called P-Swooki.

2. **A Semantic Social Tag model to *semantify* any web resource**: We define the Semdrops conceptual model as an extension of the Gruber model. Semdrops extends the Tag element with a Semantic Tag that is refined into three more specific elements: Category tags, Property tags and Attribute tags. Additionally, we developed the Semdrops Firefox-add on to support the complete knowledge building process in order to create semantic information to any web resource. With Semdrops it is possible to add semantic tags to any web resource (semantic or not). The social semantic tag generates in a collaborative process a lightweight semantic definition of the web resources, especially those resources which are not currently part of the Semantic Web.

This part is based on our previous work published on four occasions:

- In [91] and [94] we introduced the preliminaries of the Personal Semantic Space in Semantic Wikis and personal navigation in semantic wikis. Finally, the article "Supporting Personal Semantic Annotations in P2P Semantic Wikis" [95] consolidates previous work.

- The article "Semdrops: A Social Semantic Tagging Approach for Emerging Semantic Data" [92] where the Social Semantic Tagging model is introduced and developed in the Semdrops add-on.

2.1 Outline of the Second Part

In order to present a reader guide, the second part of this dissertation is organized in the following chapters:

- Chapter 3 describes related work and background information for this part of the thesis. Section 3.1 describes the conceptual Collaborative Knowledge Building Process introduced by Stahl, and the four basic interaction types in the process: Externalization, Publication, Internalization and Reaction. Then, Section 3.2 presents Semantic Wikis and their main characteristics in knowledge definition such as mark-up language, ontology representation and queries. Finally, the chapter describes the bases of Social Tagging and the Gruber's Tag model in Section 3.3.

- Chapter 4 is dedicated to P-Swooki. P-Swooki supports the Personal Knowledge Management in Semantic Wikis, particularly with the case of P2P semantic wikis. At the beginning of the chapter, Section 4.1 describes Collaborative Knowledge Building Approach in the context of P2P semantic wikis. In this work, we extend P2P semantic wikis by supporting personal understanding building. Particularly, we extend the concepts of Externalization and Publication. Both of them are detailed in Section 4.1.1. The extension is conceptualized in a Personal Semantic Annotations Model which is described in Section 4.1.2. Finally, Section 4.1.2 presents the P-Swooki approach that incorporates in a P2P semantic wiki personal semantic annotation model.

- Chapter 5 is dedicated to describing the Semdrops approach. Semdrops continues the line of providing the complete collaborative knowledge building system but being exploited to any Web resource. The aim of Semdrops is to *semantify* the Web by adding semantic tags to any Web resource. First, we describe the Social Semantic Tag conceptual model in Section 5.1.1. This model enables users to add semantic information to a web resource in terms of Categories, Properties or Attributes. After that, in Section 5.1.2, we describe the use of Semantic Mediawiki as support for our approach; and we also provide a workflow which describes how to use the tagging model in Section 5.1.3. Finally, in Section 5.2, we introduce the Semdrops Firefox add-on which enable it possible to perform the semantic annotation to any web resource by using a Web browser.

- Chapter 6 is dedicated to the enrichment of the Semantic Web information thanks to the social forces. The enrichment is made by the use of two tools: P-Swooki and Semdrops. In this chapter we introduce the evaluations of P-Swooki and Semdrops approaches. The evaluation part begins with the study of usability of P-Swooki approach. In this evaluation we analyze the production of personal and shared annotations by a group of users. After the interaction with P-Swooki, the users were asked how comfortable the knowledge building process was. On the other hand, in Section 6.2, we evaluated the usability and data generation quality of Semdrops. In this evaluation we conducted a SUT survey and we compared the semantic annotation made by Semdrops with those that are included in DBpedia.

3

Background

Contents

3.1 Collaborative Knowledge Building 25

3.1.1 Tools for Collaborative Knowledge Building Systems for Semantic Data 27

3.2 Semantic Wikis 27

3.3 Social Tagging 30

3.3.1 Gruber's Model 31

3.4 Efforts in Semantic Data Generation 32

In this chapter, we provide the necessary background on research that motivates this Part of the thesis. We briefly discuss the collaborative knowledge building process on the field of producing semantic data, the use of personal information in semantic wikis, and social tagging approaches.

3.1 Collaborative Knowledge Building

Collaborative knowledge building focuses on understanding as a learning process where personal understanding can not be built internally without social interaction. People need to participate in a social process and create new knowledge collaboratively. Gerry Stahl proposes a conceptual collaborative knowledge building model which shows the *"mutual constitution of the individual and the social knowledge building as a learning process"* [84] , as depicted in the figure 3.1.

Stahl's process starts with the description of the *personal understanding* by specifying personal beliefs, which are tacit. Then, personal beliefs can be articulated in a *"language"* and they enter into a social process of interaction with other people and their shared understanding. Later, the shared knowledge enters into the personal understanding and provokes a change in personal beliefs, motivations and concerns. When this happens, these modifications become a new tacit understanding and will be the new starting point for future

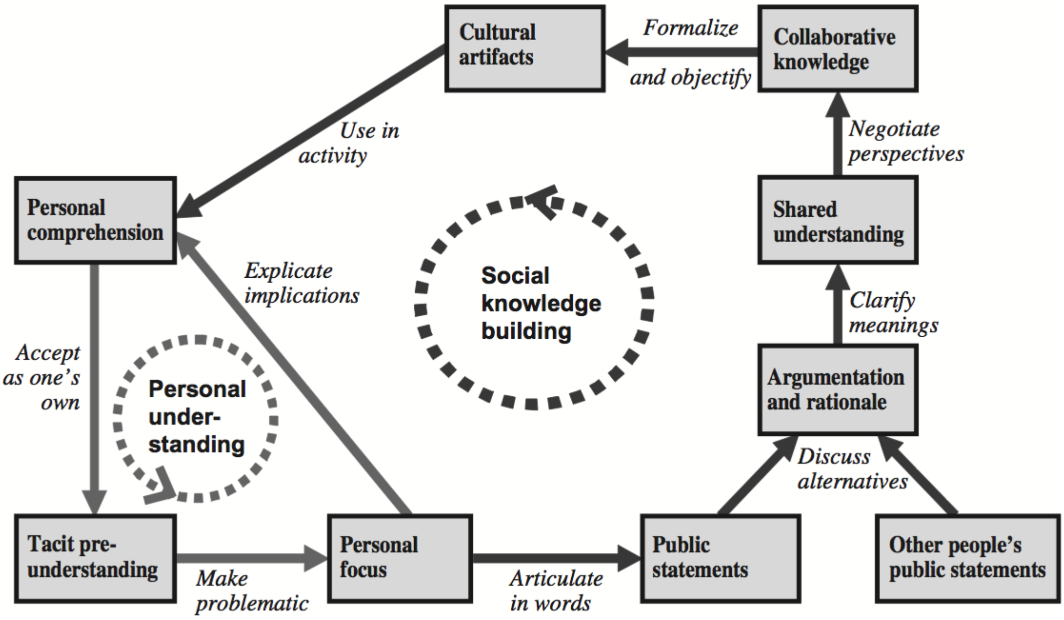


Figure 3.1: Stahl's Collaborative Knowledge Building Process

understanding and further learning. Diaz et al. [24] reinterpreted this process and proposed a four-step-spiral process for centralized knowledge sharing.

- **Externalization**: It is the act of making explicit some knowledge that is tacit. This process is done in the personal individual context. In this process the person re-organizes their own believes and understanding to include a new piece of information. The new explicit knowledge has to be expressed in a language tool that could be formal or informal. For example, an email or a written note is informal, and the use of an ontology definition is a formal language.

- **Publication**: It is the action to share the personal explicit knowledge with other people. At this moment, the explicit knowledge has to be articulated in a community common language. This action produces new-shared knowledge.

- **Internalization**: This is done when knowledge goes from the shared space to the personal space and from explicit to tacit. The individual incorporates, in its personal space, some pieces of knowledge from the community. In this process, new pieces of knowledge could generate some contradictions that the person could externalize in a Reaction.

- **Reaction**: It is the act of opening a discussion and argumentation linked to previous shared contributions to achieve a consensus. A reaction always involves an externalization and an eventual publication.

The majority of work on knowledge management focus on *organizational knowledge management* [15, 49]. Many of them follow the traditional Knowledge Management (KM) approach [64] to create large centralized knowledge repositories, in which corporate knowledge is collected, represented and organized, according to a single - shared - conceptual schema [55]. *"This centralized approach -and its underlying objectivist epistemology- is one of the reasons why so many KM systems are deserted by users"*[14]. In [14, 15], the authors propose a P2P organizational knowledge management in order to make organizational memory more flexible. However, this approach is more suitable to *knowledge discovery and propagation* rather than collaborative and personal knowledge building.

3.1.1 Tools for Collaborative Knowledge Building Systems for Semantic Data

Several tools foster collaborative knowledge building systems to support semantic data generation. Co-Protégé [23] is a collaborative knowledge tool that extends Protégé [30]. Protégé is an extensible platform for knowledge-based systems development and research mainly used for knowledge engineers. Consequently, Co-Protégé focused on the ontology development and in the management of conflictive situations. Co-protégé makes the occurrence of conflicts and its resolution an explicit process. In order to build an ontology in a collaborative manner, Co-Protégé introduced an adaptation of the knowledge sharing process which takes into account the knowledge sharing activity that occurs when a group of people collaboratively develops an ontology.

The activity is described in a knowledge-sharing-process where the knowledge moves from personal spaces to shared spaces in a complete cycle. The interactions of the knowledge building cycle are the ones described in the previous section such as internalization, publication, externalization and reaction. Figure 3.2 shows a screenshot with the private and shared spaces.

In addition to the field of Collaborative Knowledge Building Systems, OntoEdit [87] is a system used to develop ontologies. OntoEdit defines a workflow consisting in three stages:

- **Requirements Specification**: According to the authors, this task is performed by experts in the domain and in the modeling. This stage is supported by two tools: OntoKick and Mind2Onto. OntoKick is focus on the creation of the requirements specification documents, and extraction of the relevant structures for the building of a semi-formal ontology description.

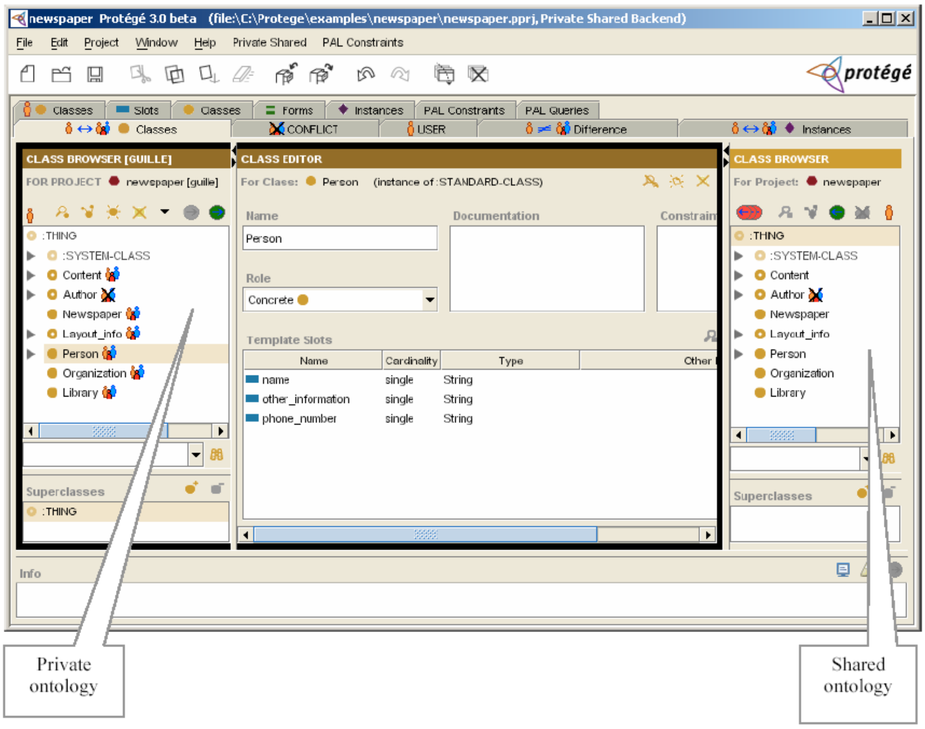
- **Refinement**: The goal of this step is to refine the ontology definition into a more mature ontology.

- **Evaluation**: This phase targets at the evaluation of the formal ontology definition. At the end of this phase the ontology is published to be used in a productive environment.

A screen capture could be seen at Figure 3.3.

3.2 Semantic Wikis

Semantic Wikis combine the wiki properties such as easy content elaboration, community knowledge building features, with Semantic technologies like struc-


newspaper Protégé 3.0 beta (file:\C:\Protege\examples\newspaper\newspaper.pprj, Private Shared Backend)
File Edit Project Window Help Private Shared PAL Constraints
protégé
Classes Slots Classes Forms Instances PAL Constraints PAL Queries
Classes CONFLICT USER Difference Instances
CLASS BROWSER [GUILLE]
FOR PROJECT newspaper [guille]
:THING
:SYSTEM-CLASS
Content
Author
Newspaper
Layout_info
Person
Organization
Library
Superclasses
:THING
CLASS EDITOR
For Class: Person (instance of :STANDARD-CLASS)
Name
Documentation
Constraints
Role
Template Slots
CLASS BROWSER
For Project: newspaper
:THING
:SYSTEM-CLASS
Content
Author
Newspaper
Layout_info
Person
Organization
Library
Superclasses
Info
Private ontology
Shared ontology

Name: Person

Role: Concrete

Name	Cardinality	Type	Other
name	single	String	
other_information	single	String	
phone_number	single	String	

Figure 3.2: Co-Protégé: The screenshot shows the personal and the shared ontology edition spaces

tured content, a machine-readable language and ontologies[20, 43]. According to Shaffert, *"Semantic wikis connect social and artificial intelligence, supporting users in ways that aren't available in normal wikis."*[79].

Regular wikis enable users to collaboratively edit web pages. The most important properties in a Wiki are that each user is open to participate in the content, to create new pages, to categorize articles and to discuss with other users without managing technical knowledge.

A Semantic Wiki extends the regular wiki facilities with semantic technologies which allow users to include extra information to structure wiki data. For example, it includes flexible annotations to add semantic labels in the body of the articles by means of extending the wiki mark-up language, for instance by adding a type in the links between articles. The semantic mark-ups used in typing links **generate an underlining ontology definition** that is created in the same collaborative process that generates the wiki content. In general, most of the Semantic Wikis share some basic features: properties, data types, OWL/RDF mapping and embed semantic queries.

- **Properties.** Are named semantic relations between two articles. The way to declare or create a property is by means of adding a type to a direct link between the articles. The type of the direct link is the property. For example, Semantic Mediawiki (SMW) extended the Mediawiki mark-up to link one article to another by adding the property name. The property name describes the semantic relations that link one article with the another. For example, at top of Figure 3.4 the edition page of London article is shown . It is easy to understand the mark-ups, for

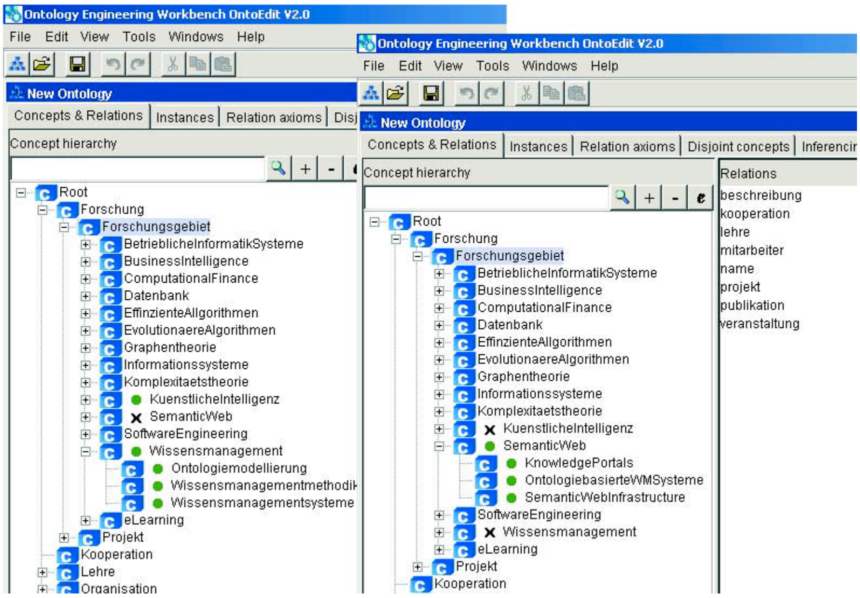


Figure 3.3: OntoEdit

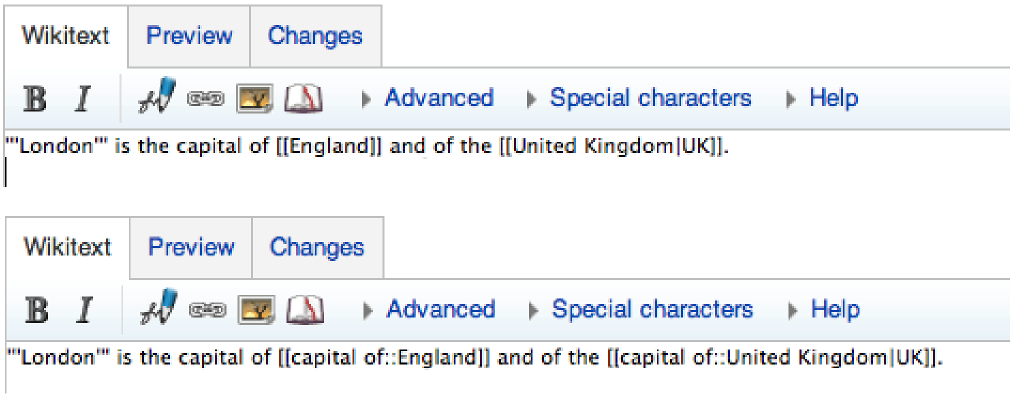
example `'''...'''` is used to boldface the text in between and `[[England]]` is a link to England article. To enhance the London article with semantic information, the link to England could be written as `[[capital of::England]]`. In this way, `capital of` is the type for the link and, as a consequence, the London article contains a property `capital of` with value England.

- **Data types** Properties can use several data types. Some of the basic data types provided are Strings, Date, Geographic coordinate, and Page. A data type has a definition of the possible values, conversion rules (for example from kilometers to miles), and rendering alternatives. In Semantic Mediawiki it is possible to extend the data types definition.

- **OWL Mapping and Semantic Features** The different semantic wiki implementations map the semantic annotations into a specific OWL or RDF definition. A lightweight ontology appears on the basis of the semantic annotations written by the wiki community. Most of the Semantic Wikis implementations have a direct mapping to an ontology definition, for example in SMW normal pages correspond to abstract individuals, properties correspond to OWL properties, categories correspond to OWL classes, and property values can be abstract individuals or typed literals. Most of the annotations are mapped to RDF triplets.

Semantic Mediawiki includes an important feature based on the query system. It is possible to insert the results of a semantic query in the content of a page with the use of *inline queries*. Inline queries allow editors to add dynamically created lists or tables to a page, thus making up-to-date query results available to readers who are not even aware of semantic queries. For example, Listing 3.1 details the inline query from Germany[1] page in `semanticweb.org`.

[1] `http://semanticweb.org/wiki/Germany`



'''London''' is the capital of [[England]] and of the [[United Kingdom|UK]].

'''London''' is the capital of [[capital of::England]] and of the [[capital of::United Kingdom|UK]].

Figure 3.4: SMW mark-up enhanced for semantic annotations.

The query lists the name and population of all articles which belong to category `City` and are related to `Germany` article with the semantic property `located in`. The results of this query are shown in Figure 3.5.

```
{{#ask: [[Category:City]] [[located in::Germany]] | ?Population}}
```

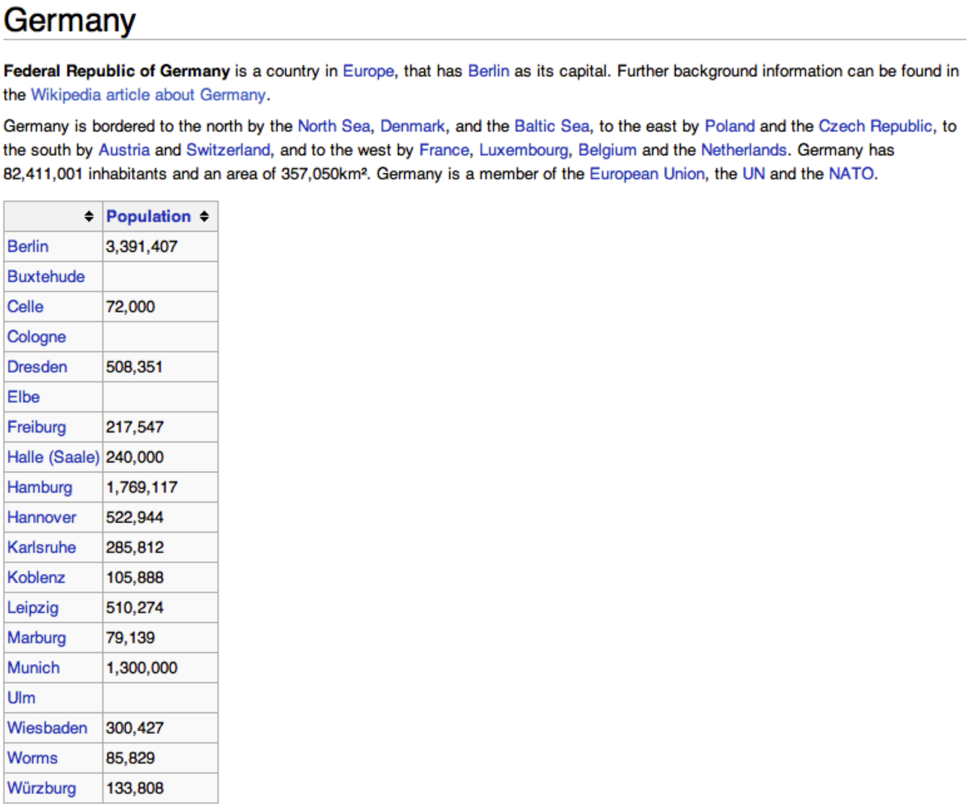
Listing 3.1: Semantic Mediawiki inline query example.

There is a wide diversity of Semantic Wikis. Most of them enhance the regular social interaction with a wiki by adding semantic capabilities like typing the links or define OWL classes with the use of categories, such as Semantic Mediawiki [44] and IkeWiki [78]. SweetWiki [19] also defines an ontology to describe the wiki itself called *"The Wiki Object Model"* and it supports social tagging. The *Wiki Object Model* allows semantic engineers to make SPARQL queries over the semantic wiki concept in addition to the wiki articles that are part of the wiki. Other example is Swooki [83], Swooki combines advantages of P2P wikis with Semantic wikis. Swooki ensures consistency on replicated pairs by means of Woot [68] algorithm. The architectural organization of Swooki connects in a P2P network several Semantic Wikis that synchronize the articles and semantic information contents.

All of these semantic wikis are more appropriate to support collaborative knowledge emerging, in contrast, they do not provide functionalities to manage combined personal and shared understandings. However, personal semantic wikis like SemperWiki [66, 67] only support personal knowledge building. Currently, there are no semantic wikis that help people to combine and manage in a usable way both personal and shared knowledge.

3.3 Social Tagging

Social Tagging is the activity in which users cooperate to put a label to describe a Web resource within the Social Web [34, 45]. As we have seen in the previous section, the properties in Semantic Wikis could be seen as a kind of tagging that requires some formalism (i.e. additionally to the type name, the property is a semantic relation between two resources) and SweetWiki evidences a difference by including both social tagging and traditional semantic wiki features. The less structured activity of free tagging was defined as *folksonomies*.



Germany

Federal Republic of Germany is a country in Europe, that has Berlin as its capital. Further background information can be found in the Wikipedia article about Germany.

Germany is bordered to the north by the North Sea, Denmark, and the Baltic Sea, to the east by Poland and the Czech Republic, to the south by Austria and Switzerland, and to the west by France, Luxembourg, Belgium and the Netherlands. Germany has 82,411,001 inhabitants and an area of 357,050km². Germany is a member of the European Union, the UN and the NATO.

⬍	Population ⬍
Berlin	3,391,407
Buxtehude	
Celle	72,000
Cologne	
Dresden	508,351
Elbe	
Freiburg	217,547
Halle (Saale)	240,000
Hamburg	1,769,117
Hannover	522,944
Karlsruhe	285,812
Koblenz	105,888
Leipzig	510,274
Marburg	79,139
Munich	1,300,000
Ulm	
Wiesbaden	300,427
Worms	85,829
Würzburg	133,808

Figure 3.5: SMW inline query result.

Wander Val called *folksonomies* the activity of freely tagging items on the Web. According to Wander Val, folksonomies are created when people tag items online for their own later information retrieval purposes. One of the benefits related to folksonomies are the management of personal information by using users own words instead of using keywords defined by the system. Another benefit is the social exchange and reuse of tags done by other users, and finally the use of tags to make searches and discover related elements by using the same user tag language [96]. The usage of user's tags with similar interests had a tendency to cover a shared knowledge[39].

Some Web portals include the management of tags in different resource types, such as Flickr[2] for photo tagging, Bibsonomy[3] for bibliographic reference tagging or Delicious[4] for bookmark tagging. Figure 3.6 shows an example of *Delicious.com* tags folksonomy.

Tom Gruber introduced a semantic conceptualization of social tagging.

3.3.1 Gruber's Model

Gruber defines the core concept of Tagging for a full social environment where users tag several resources with different labels. However, if two applications share the same tagging model between them and we have to compare tags from

[2] http://www.flikr.com

[3] http://www.bibsonomy.org

[4] http://delicious.com

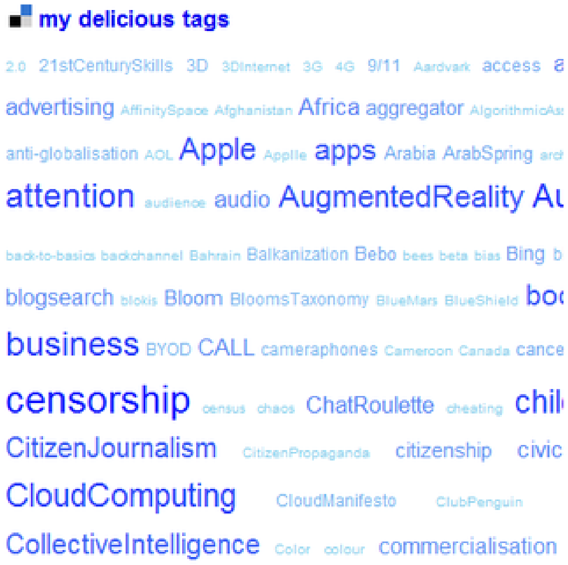

my delicious tags
2.0 21stCenturySkills 3D 3DInternet 3G 4G 9/11 Aardvark access
advertising AffinitySpace Afghanistan Africa aggregator
anti-globalisation AOL Apple Apple apps Arabia ArabSpring
attention audience audio AugmentedReality
back-to-basics backchannel Bahrain Balkanization Bebo bees beta bias Bing
blogsearch blokis Bloom BloomsTaxonomy BlueMars BlueShield
business BYOD CALL cameraphones Cameroon Canada
censorship census chaos ChatRoulette cheating
CitizenJournalism CitizenPropaganda citizenship civic
CloudComputing CloudManifesto ClubPenguin
CollectiveIntelligence Color colour commercialisation

Figure 3.6: Delicious.com tags folksonomy.

different systems, the tagging model needs to include the system that manages the particular tags. In conclusion, the tagging model is defined as:

```
Tagging(<resource>,<tag>,<tagger>,<source/context>)
```

Where <resource> is the tagged element, <tag> is the label selected by the <tagger> and <source/context> is the application where the tagging action was done. With this model it is possible to represent the following scenario:

```
Tagging(Object1, tag1, tagger1, source1)
Tagging(Object1, tag2, tagger1, source1)
Tagging(Object1, tag1, tagger2, source1)
Tagging(Object1, tag3, tagger3, source2)
Tagging(Object2, tag1, tagger4, source2)
```

The importance of the Gruber's tagging model recalls in the semantic definition of the tag activity. This model could be extended to bootstrap the human methods in the web and to describe any web resources in a Semantic Web compatible manner.

3.4 Efforts in Semantic Data Generation

Over the last years, many efforts have focused on the *semantification* of the web. One of the most relevant is the Linking Open Data (LOD) [5] initiative. The objective of LOD is to connect related data that was not previously connected using URI and RDF. LOD generates a large amount of RDF statements, most of them are generated automatically, only less than one percent of semantic contents are a user-generated [10]. DBPedia [12] is a major application of LOD. DBpedia is the Linked Data version of Wikipedia, it is based on an automatic approach to generate RDF statements, by converting structured information in Wikipedia pages into RDF statements. In *Semdrops* approach, the semantic data emerge by the social effort.

[5] http://linkeddata.org

Other approaches to increase the semantic data on the web are based on the enrichment of social tags with semantic information:

- Annotea [40] defines a simple ontology to generate semantic bookmarks.
- Mukherjee et al. introduce the idea that semantic bookmarks are generated automatically to structure hypermedia documents [62]. This approach is limited to specific web resources i.e. the bookmarks.
- SOBOLEO [6] [101] uses SKOS taxonomy to annotate bookmarks. Although, SOBOLEO allows the set of tags to web resources and organizes them into hierarchies, it does not support a rich semantic model.
- In Limpens´s Ph.D. thesis[51] and Monin et al. [61] the *niceTag* ontology which sub-classifies the *isRelatedTo* tag into a set of sub-properties to conceptualize the different possible uses of tags presented by Golder & Huberman is introduced. Unfortunately, the rigid structure in the use of pre-defined properties does not allow users to add new properties definitions.
- Huynh-Kim-Ban et al. [7] introduced an extension of the tag model presented in del.icio.us which includes both *synonymy* and *inclusion* relationships among tags, but they do not include the possibility to express, for example, attributes.

OntoGame [81] combines Web 2.0 and semantic definition to determine ontology construction, ontology alignment, and ontology population. Authors propose five activities for the players: collecting name entities, matching name entities according to a defined ontology, introducing hierarchical relations like subclasses, modularization and lexical enrichment. For example, they propose games for semantic annotations. In these games it is necessary to include a resource and an ontology definition. The game shows the user a resource and a suitable domain ontology, then the players have to select the appropriate ontology element to annotate the resource. The output will be a semantic annotation.

Some other approaches propose semantic enrichment of social tagging [31, 41] by defining a tag ontology for this purpose. For instance, TagOntology [63] proposed a tripartite system (User, Resource, Tag); MOAT [73] allows the use of a URI to define the meaning of a tag, and SCOT [42] enforces the social context involved in folksonomies. All of these approaches focus on giving semantic to the tags.

[6] Social Bookmarking and Lightweight Engineering of Ontologies

4

P-Swooki

Contents

4.1 P2P Collaborative Knowledge Building Approach 36

4.1.1 P2P Collaborative Knowledge Building Process . . . 37

4.1.2 Personal Semantic Annotations: Individuals and Categories 38

4.2 P-Swooki: P2P Collaborative Knowledge Building System 39

4.2.1 Shared Semantic Annotation Management 39

4.2.2 Personal Semantic Annotations Management 39

4.2.3 P-Swooki Data Model 40

4.2.4 Personal Semantic Annotation Storage Model 40

4.2.5 Editing Operations 41

4.3 P-Swooki Implementation 41

Immersed in the co-evolution context, we are interested in the bootstrapping of the social methods to create pieces of knowledge, and the Semantic Web tools to represent them. An initial approach should provide tools that support the collaborative knowledge building process. Particularly, the articulation between personal and social knowledge spaces. Additionally, the pieces of knowledge that are created in the social process require to be represented in a semantic description. In this part of the dissertation, the support to complete the knowledge building process is done by extending a P2P Semantic Wiki. The extension is centered in the management of personal and shared knowledge spaces. Furthermore, semantic wikis provide a semantic representation of their content. In our approach, the semantic representation is adapted to the management of personal and shared spaces. In consequence, an augmented tool with better navigability, structuring and knowledge representation allows the emerging of Semantic Web enabled knowledge in both personal and shared spaces.

More specific, the goal of this work is to propose an innovative semantic wiki approach that supports both personal and shared knowledge building. In this approach, the shared knowledge is unique and accessible to everyone, while the personal knowledge is only accessible by its owner and represents the user private view (perspective) of the shared one. Personal knowledge can differ from the shared one, but it can also have overlapped parts.

For the emerging of shared knowledge, we follow the same approach as SMW where *shared semantic annotations* are embedded in the wiki text by using a suitable syntax. For the personal knowledge, we propose *Personal Semantic Annotations* to externalize personal understanding. *Personal semantic annotations* are associated to the wiki page and they are only accessed by the owner user. For the end-user, the *personal semantic annotations* look like *tags*, however they are semantically richer: they support categories and individuals.

In our experience, the addition of *Personal Semantic Annotations* to semantic wikis enables users:

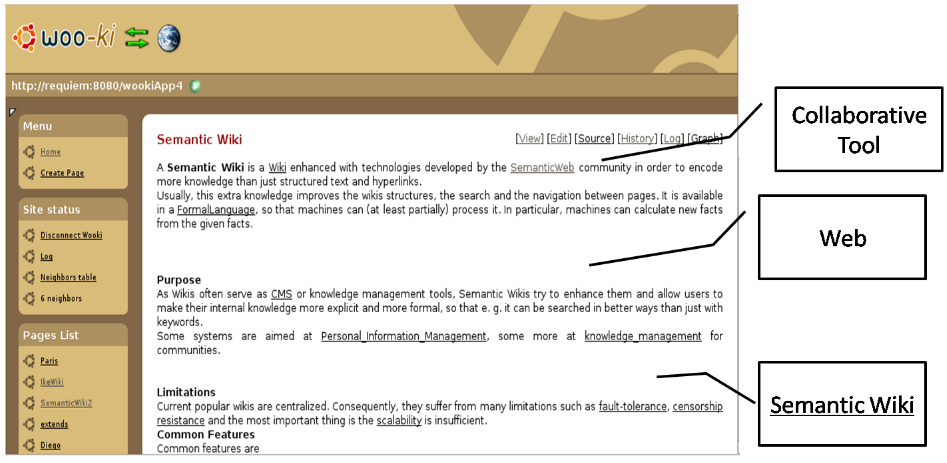
- To support the individual understanding in the collaborative knowledge building process [84]
- To provide personalized knowledge retrieval, structuring and navigation.
- To enable a combined personal and shared knowledge retrieval.
- To enrich the shared semantic annotations and to augment, therefore, the shared knowledge base.

Moreover, adding personal semantic annotations and shared ones involves complementary activities. Whereas adding a shared semantic annotation seems to be suitable during the edition, adding a personal one seems to be more suitable during the browsing activity. In order to validate this hypothesis, we have conducted an evaluation study that is detailed in Section 6.1.

In this chapter of the dissertation, we introduce a peer to peer semantic wiki called *P-Swooki* that supports both personal and shared knowledge building. *P-Swooki* extends a peer-to-peer semantic wiki Swooki [83] by adding personal knowledge building. To validate our approach we choose a peer to peer semantic wiki because in a P2P architecture information dissemination is easily controlled *i.e.* shared annotations are broadcast and integrated by all peers while personal semantic annotations remain local.

4.1 P2P Collaborative Knowledge Building Approach

In this work, we extend P2P semantic wikis by supporting personal understanding building. In addition to sharing semantic annotations embedded in the wiki text, users can also associate *personal semantic annotations* to semantic wiki pages. These private annotations express personal understanding of the users. For example, if a user was navigating the semantic wiki page "Semantic Wiki" as it is shown in the figure 4.1, eventually, she would like to annotate this page as "Collaborative Tool", "Web" and "Semantic Wiki". If these annotations only express personal understanding, they should be private. Other annotations as "Semantic Web" or "Wiki" are shared, they could be defined by the same user or by other users. We can notice that users manage simultaneously both shared and personal semantic annotations.



Semantic Wiki

A **Semantic Wiki** is a Wiki enhanced with technologies developed by the SemanticWeb community in order to encode more knowledge than just structured text and hyperlinks.
Usually, this extra knowledge improves the wikis structures, the search and the navigation between pages. It is available in a FormalLanguage, so that machines can (at least partially) process it. In particular, machines can calculate new facts from the given facts.

Purpose
As Wikis often serve as CMS or knowledge management tools, Semantic Wikis try to enhance them and allow users to make their internal knowledge more explicit and more formal, so that e. g. it can be searched in better ways than just with keywords.
Some systems are aimed at Personal_Information_Management, some more at knowledge_management for communities.

Limitations
Current popular wikis are centralized. Consequently, they suffer from many limitations such as fault-tolerance, censorship resistance and the most important thing is the scalability is insufficient.
Common Features
Common features are

Figure 4.1: Adding Personal Semantic Annotation in Semantic Wiki

We adapt the collaborative knowledge building process to P2P settings as it is detailed in the next section.

4.1.1 P2P Collaborative Knowledge Building Process

A P2P collaborative knowledge building process is a continuous spiral process which involves *externalization*, *publication*, *internalization* and *reaction* as we have explained in Section 3.1. Externalization and publication steps have to be redefined to support P2P settings. Internalization and Reactions are not modified.

Besides, users manage in a well-differentiated way both personal and shared understandings. In the semantic wiki, every user needs to manage in separate spaces the personal and shared annotations. In this line, we define two repositories: the *personal understanding repository* and the *shared understanding repository* respectively. In a P2P setting, we consider that every user works in one peer and has both repositories. The *shared understanding repositories* will eventually be identical for all users due to the synchronization algorithms [68].

Our P2P collaborative knowledge building process redefines the externalization and publication steps as:

- *Externalization*, where personal knowledge goes from tacit to explicit. Users use personal semantic annotations to externalize their own knowledge. This is an individual activity, this knowledge remains private in the context of the *personal understanding space*.

- *Publication*, where knowledge goes from the individual context to the shared one. As a result, a personal semantic annotation becomes a shared one. In other words, this involves moving a personal semantic annotation from a given user's personal understanding repository to the shared one. This step involves replicating the annotation to every user as a shared annotation.

For example, in the figure 4.2 the "user1" externalizes "Collaborative Tool" personal annotation on her personal repository. Then, when she performs a publication this personal annotation should be disseminated to every user, even to herself. After the publication, the semantic annotation "Collaborative

Tool" should appear in every shared understanding repository as a shared annotation.

Consequently, personal understanding building is achieved by supporting the separation of both knowledge repositories (personal and shared) and the externalization step.

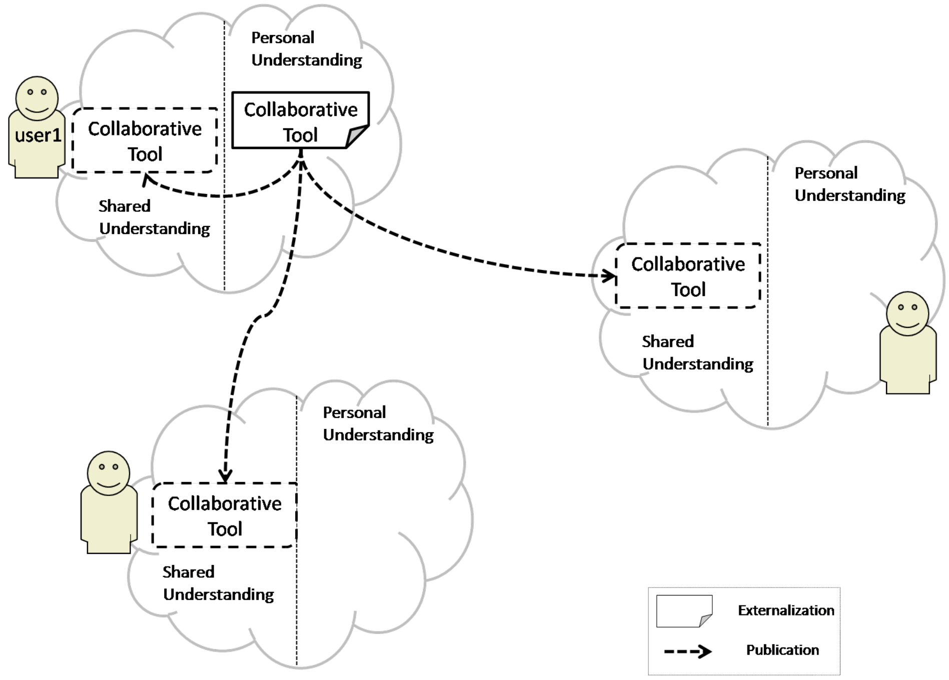


Figure 4.2: P2P Collaborative Knowledge Building Process

This P2P collaborative knowledge building approach has several advantages:

- **Personal navigation**: the system allows the users to simultaneously have personal and shared navigation on the the same content. Shared navigation is the traditional navigation supported by any semantic wiki. Personal navigation is a new kind of navigation, it is personal and it is the consequence of the personal semantic annotations. The user has instant gratification after adding personal semantic annotations.
- **Enrichment of shared knowledge** : the user can make public her personal semantic annotations. Consequently, shared knowledge is enriched.
- **Improve system usability**: adding shared semantic annotations and personal semantic ones involves complementary activities. Whereas adding shared semantic annotations seems to be suitable during the editing activity, adding personal annotations seems to be more suitable during reading activity.

In order to allow users to annotate the P2P Wiki page in a personal understanding space, we incorporate the Personal Semantic Annotations.

4.1.2 Personal Semantic Annotations: Individuals and Categories

Every semantic wiki page could be tagged with several personal semantic annotations as it was shown above. A personal semantic annotation can be a *category* or an *individual*.

Categories define a family of elements. For example, in the previous example (figure 4.1), the annotation "Semantic Wiki" was underlined in order to indicate that this wiki page is a *Semantic Wiki category definition*.

Individuals denote elements that fall at least in one category. *Semantic Mediawiki* is an individual that fall in the category Semantic Wiki. An Individual can belong to many categories.

A semantic wiki page can be annotated with many annotations. For example, a user would personally like to annotate the wiki page "Swooki" as a "Semantic Wiki" and as "P2P application".

The annotation model is simple, it only considers categories and individuals. In the following Chapter, we describe Semdrops that extends the tagging model and supports relationships and attributes.

4.2 P-Swooki: P2P Collaborative Knowledge Building System

We have developed *P-Swooki*, a P2P collaborative knowledge building system that extends the P2P semantic wiki *Swooki* with personal semantic annotations.

Shared semantic annotations are already supported by *Swooki* as detailed in the section 4.2.1. Therefore, we only had to add personal annotations functionalities to Swooki [83]. In sections 4.2.2, 4.2.3 and 4.2.5, we detail the personal annotations management, the data model and its associated operations.

4.2.1 Shared Semantic Annotation Management

The P2P semantic wiki architecture and basic features that are elementary to introduce our approach were provided by Swooki. In Swooki every peer hosts a copy of all wiki pages and the *shared understanding repository*. When a peer updates its local copy of data, it generates a corresponding operation. This operation is processed in four steps:

1. It is executed immediately against the local replica of the peer,
2. It is broadcasted through the P2P network to all other peers,
3. It is received by the other peers,
4. It is integrated to their local replica. If needed, the integration process merges this modification with concurrent ones, generated either locally or received from a remote server.

To synchronize data, Swooki implements a modified version of the P2P synchronization algorithm detailed in [68]. Swooki synchronization algorithm ensures the convergence on the wiki text and the *shared understanding repository* i.e. when the system is idle, all copies are identical.

4.2.2 Personal Semantic Annotations Management

In *P-Swooki*, personal semantic annotations are hosted locally. When a user updates her personal semantic annotations, she generates a corresponding operation. The operation is executed locally against the user *personal understanding repository*. This operation is *not* broadcast to other peers.

As it was explained above, the process to annotate a wiki page is simple. The system enables users to annotate a wiki page as a new *category* or as an *individual* of an existing category.

In order to handle personal semantic annotations, we extended the Swooki data model and defined new editing operations.

4.2.3 P-Swooki Data Model

The data model is an extension of Swooki [83, 100] data model. Therefore, each semantic wiki peer has assigned a global unique identifier named $NodeID$.

As in any wiki system, the basic element is a wiki page, therefore every wiki page has assigned a unique identifier $PageID$, which is the name of the page. The name is set when the page is created. If several servers concurrently create pages under the same name, their content will be directly merged by the synchronization algorithm. Notice that a URI can be used to unambiguously identify the concept described in the page. The URI must be global and location independent in order to ensure load balancing. For the sake of simplicity, we use a string as page identifier.

The figure 4.3 describes the personal semantic annotations data model. This data model is described by the Ontology Definition Meta-model (ODM) [29].

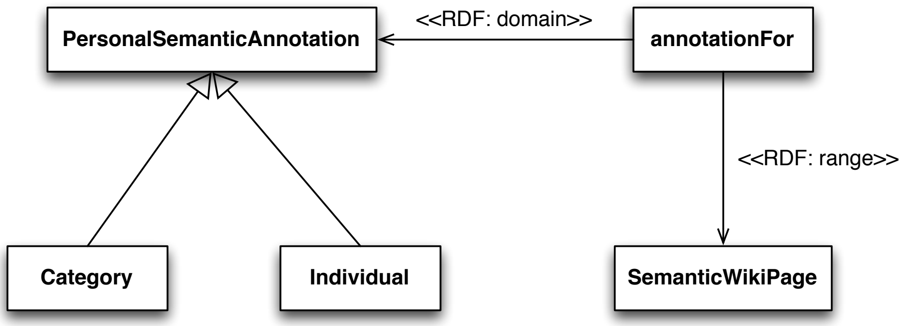


Figure 4.3: Personal Semantic Annotation Data Model

4.2.4 Personal Semantic Annotation Storage Model

RDF is the standard data model for encoding semantic data. In P-Swooki, every peer has two local RDF repositories : *Personal Statements* and *Shared Statements*. They implement the *personal understanding repository* and the *shared understanding repository* respectively.

- The *Shared Statements* contain a set of RDF statements which were extracted from the wikis pages. A statement is defined as a triple (Subject, Predicate, Object) where the subject is the name of the page, the predicates (or properties) and the objects are related to the concept involved in the page.
- The *Personal Statements* contain personal semantic annotations which are represented as personal RDF statements. A personal RDF statement is

defined as a triple (Subject, Predicate, Object) where the subject is the wiki page and the predicate annotates the page as a personal semantic annotation type as described in the next section.

We define two operations on the RDF repositories:

- insertRDF(R,t): adds a statement *t* to the *Personal Statements* or *Shared Statements* repository *R*.
- deleteRDF(R,t): deletes a statement *t* from the *Personal Statements* or *Shared Statements* repository *R*.

These operations are not manipulated directly by the end-user, they are called implicitly by the editing operations as it is shown in the following section.

4.2.5 Editing Operations

There are four editing operations for editing personal semantic annotations: *addIndividual*, *addCategory*, *delIndividual* and *delCategory*. An update is considered as a delete of an old value followed by an insert of a new value.

1. ***addCategory(**PageID**,** CategoryName**)*** : where *PageID* is the identifier of the semantic wiki page. *CategoryName* is the name of the new category.

This operation sets the wiki page *PageId* as a category in the user personal repository. This operation calls the *insertRDF(Personal Statements,(PageId, RDF.Type, CategoryName))* function to add a new triplet into the personal RDF repository.

2. ***addIndividual(**PageID**,** CategoryName**)*** : sets the wiki page *PageID* as a member of the category *CategoryName*. If *CategoryName* does not exist, it is added automatically to the *Personal Statements* repository by calling the operation *addCategory* and then the operation automatically annotates the *PageId* as member of the *CategoryName*.

During this operation an RDF statement is added to the personal repository by calling *insertRDF(Personal Statements, (PageId, belongsTo, CategoryName))* where *belongsTo* is a predicate to associate an individual to a category.

3. ***delIndividual(**PageID**,** CategoryName**)***: eliminates the *PageID* as member of the category *CategoryName* from the personal RDF repository by calling *DeleteRDF(Personal Statements, (PageId, RDF.Type, CategoryName))*.

4. ***delCategory(**PageID**,**CategoryName**)*** : first, it calls the *delIndividual* operation for each member of the category *CategoryName*, and then deletes the category *CategoryName* from the personal RDF repository by calling the *DeleteRDF* operation.

4.3 P-Swooki Implementation

P-Swooki is implemented as an extension of Swooki. Swooki is a P2P semantic wiki which is implemented in Java as servlets in a Tomcat Server and uses Sesame 2.0 as RDF repository.

P-Swooki is developed over a Swooki architecture using one peer per user. A P-Swooki peer is made up by the following components (see figure 4.4). The grey boxes are Swooki components whereas the white ones are the P-Swooki components.

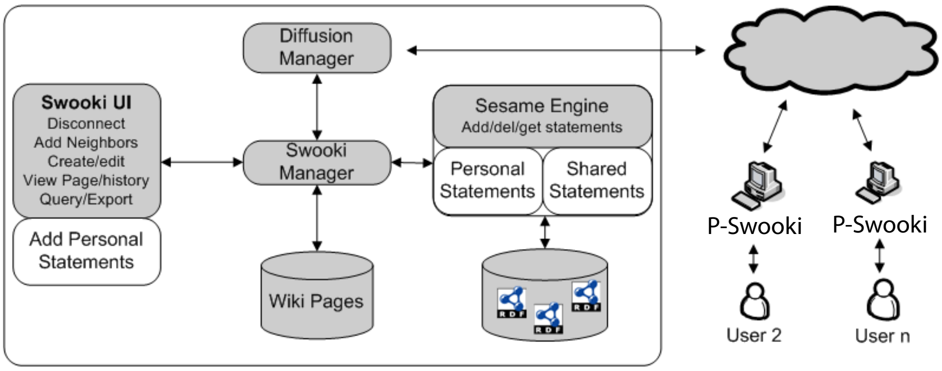


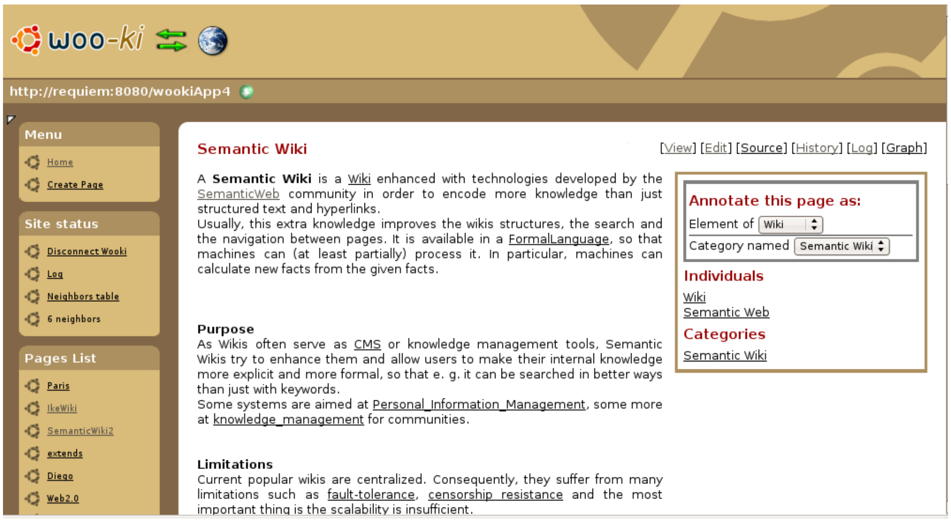
Figure 4.4: P-Swooki Architecture

User Interface. The P-Swooki UI component is composed by the Swooki wiki editor and it incorporates the functionalities to make personal annotations. This basically divides the wiki page into two areas: the shared and private annotation spaces. The shared space is defined by a regular wiki editor supported by Swooki functionality. The private annotation one includes a box to add personal semantic annotations and to visualize them (see figure 4.5).

Swooki Manager. The Swooki manager implements the synchronizing algorithm.

Sesame Engine. We use a multi-set [83] extension of Sesame 2.0 [17] as RDF repository. Sesame is controlled by the Swooki manager for storing and retrieving RDF statements. P-Swooki stores the private annotations using a different name space. This allows administrators to reuse the storing and retrieving facilities already implemented by Swooki.

Diffusion Manager. The diffusion manager is in charge of maintaining the membership of the unstructured network and of implementing a reliable broadcast for the shared repositories.



Semantic Wiki

A **Semantic Wiki** is a Wiki enhanced with technologies developed by the SemanticWeb community in order to encode more knowledge than just structured text and hyperlinks.
Usually, this extra knowledge improves the wikis structures, the search and the navigation between pages. It is available in a FormalLanguage, so that machines can (at least partially) process it. In particular, machines can calculate new facts from the given facts.

Purpose
As Wikis often serve as CMS or knowledge management tools, Semantic Wikis try to enhance them and allow users to make their internal knowledge more explicit and more formal, so that e. g. it can be searched in better ways than just with keywords.
Some systems are aimed at Personal_Information_Management, some more at knowledge_management for communities.

Limitations
Current popular wikis are centralized. Consequently, they suffer from many limitations such as fault-tolerance, censorship resistance and the most important thing is the scalability is insufficient.

Annotate this page as:
Element of Wiki
Category named Semantic Wiki

Individuals
Wiki
Semantic Web

Categories
Semantic Wiki

Figure 4.5: P-Swooki Interface

5

Semdrops

Contents

5.1 Semdrops Approach 44
5.1.1 Semantic Social Tagging Model 44
5.1.2 Using Semantic Mediawiki as Semantic Support . . . 45
5.1.3 How to Use the Model ? 46
5.2 Implementation 47

This chapter introduce a complementary approach to achieve the first hypothesis of this thesis. In this case, we extend the model proposed with P-Swooki in order to add semantic data to any web resource. In this chapter, the knowledge building process and a social semantic tagging activity are bootstrapped in order to generate new semantic data for any web resource.

We propose an approach that allows the emergence of semantic data by means of identifying and modeling the tagging activity. This is a step to the *semantification* of the web based on social effort. The proposed approach has:

- to promote semantic data proliferation.
- to promote the semantic data quality
- to be applicable to any web resources.
- to be used by a wide community
- to provide an easy interaction mode.

In order to achieve all of these requirements, the challenge now is to define a simple semantic social tagging model whose integration into the system provides an easy interaction modality that enables the attaching of a web resource without modifying the resource itself. In this work, we propose a *social semantic tagging* approach called *Semdrops*. *Semdrops* defines a conceptual model which is an extension of the Gruber tag model [31], where the tag concept is extended to semantic tag. We implement *Semdrops* as a Firefox add-on tool that turns the web browser into a collaborative semantic data editor *in situ* for any web resource.

To validate *Semdrops*'s approaches, we conducted an evaluation and usability studies and we compared the results with automatic generation methods of semantic data such as DBpedia. The studies demonstrated that *Semdrops* is an effective and complementary approach to produce adequate semantic data on the Web (The details of the evaluation are in the Section 6.2 of Chapter 6).

5.1 Semdrops Approach

Semdrops uses the force of mass collaboration to facilitate the emergence of semantic data. This data can be used to emerge lightweight ontology. It is based on a social semantic tagging strategy [42] that adopts the characteristics of social tagging enhanced with semantic meta-data. In order to make the semantic annotation activity easy to the user, Semdrops proposes a simple semantic representational model to conceptualize the social semantic tagging activity. This model captures the semantic annotations performed by the users. This means that the model allows users to represent the attachment of social semantic tags to a web resource in an easy manner. The model also enables users to tag a resource without modifying the resource itself. Section 5.1.3 presents a detailed example.

Although Semdrops social semantic tag model was designed to make the tagging activity easy by regular users, it has to be mapped to existing semantic representation models like RDF, OWL in order to make it compatible with the Semantic Web technologies. In this work, we particularly use Semantic MediaWiki to make this mapping as is detailed in the section 5.1.2.

5.1.1 Semantic Social Tagging Model

According to the literature, the most elementary building blocks of a tagging model is a tripartite model [63] made up of the taggers, the tags and the resources being tagged. As we detailed in Section 3.3.1, Gruber [31] introduced into the model one more element: the source of the tags, and named the model as the "four-places relation" model. *Semdrops* conceptual model is an extension of the four-places relation one, where the tag concept is extended to *semantic tag*. *Semdrops* defines four main concepts: *Web resource*, *tagger*, *semantic tag* and *semantic support*.

Web resource: A *Web resource* is any Resource in the web which could be identified by an URI.

Tagger: The *tagger* is the responsible of adding semantic tags to the web resource. A *tagger* can be an individual user or a community.

Semantic Tags: According to mc sharaefel et al.[56], *"a semantic tag is one where the tag itself is backed up by an RDF. Semantic tags are semantic annotations used to add a semantic description over a Web resource in an unsophisticated manner. "*. In our approach, we extend the definition of mc sharaefel et al.; particularly, we describe three kinds of semantic tags: Category Tag (CT), Property Tag (PT) and Attribute Tag (AT).

- **Category Tag (CT):** This semantic tag represents the belonging of a Web Resource to a Category. Tagging a resource by means of a CT defines the name of the tag as identifier of the category and the Web Resource as individuals which belong to that category. Also, it is possible to organise categories specifying detailed levels by means of hierarchies using *subsumption*. A sub-category is defined as a subset of individuals of the super-category.

- **Property Tag (PT):** This semantic tag describes a relationship between two Web Resources. In this case, the name of the property types the relationship between the two web resources, even when there is not a navigational link between them. Having PT, the navigation is improved by a semantic relationship.

- **Attribute Tag (AT):** This semantic tag describes a structural attribute for a web resource, for example the height of a mountain. It uses the label as a name for the attribute and it is related to a simple data type as a number or a literal.

Semantic Support. We use this support to store semantic tags. The underlying semantic support has an impact on how *Semdrops* maps semantic tags to an existing semantic support data model. This is done by specifying a set of mapping rules. This semantic support makes Semdrops model compatible with other Semantic Web technologies. It can be any RDF repository e.g. Sesame [17], Jena [57] or even Semantic Mediawiki [43].

5.1.2 Using Semantic Mediawiki as Semantic Support

In this work, we particularly use Semantic MediaWiki (SMW) as semantic support. We have chosen SMW because we have already used semantic wikis for knowledge personalization as we described in the previous Chapter, and this allows the use of the powerful features of SMW such as query semantic tags and export semantic data.

Mapping Rules

We define a set of rules to transform Semdrops's semantic tags into suitable SMW semantic annotations in order to store them in the semantic support of SMW. In order to carry out this mapping, a web resource is mapped to a wiki article in terms of SMW.

- **Category Semantic Tag.** Let X be the URL identifying the Web Resource, and CatName the label for a category.

  $CategoryTag(X, CatName) \equiv$
  $SemWiki(X) \leftarrow ([[Category : CatName]])$.

  Where $SemWiki(page) \leftarrow (code)$ means that the wiki page titled *page* has the string *code* included in its content.

- **Property Semantic Tag.** Let X, Y be the URLs identifying two different web resources and let PropName be the label of a property.

Semdrops	**SMW**	**OWL-like**
Category(X,TName)	[[Category: TName]	X rdf:type TName
Property(X,PName,OW)	[[PName::OWContent]]	X PName OW
Attribute(X,AName,Value)	[[AName:= Value]]	X AName Value

Table 5.1: Semdrops Mapping Rules

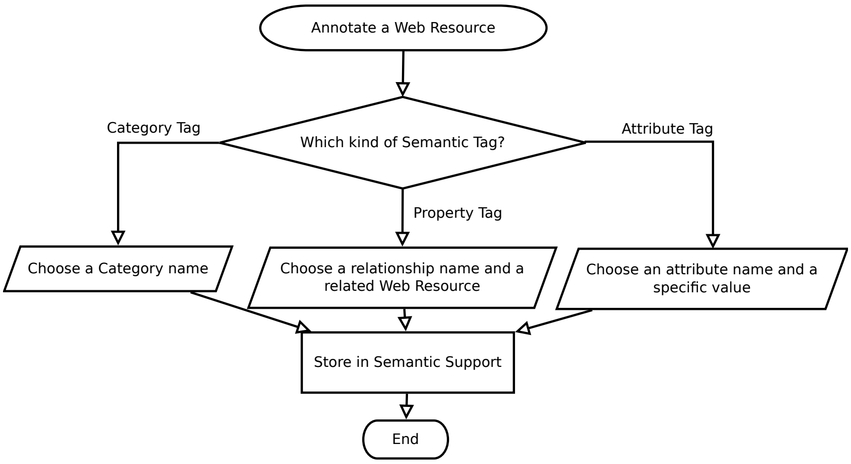

Annotate a Web Resource
Which kind of Semantic Tag?
Category Tag
Property Tag
Attribute Tag
Choose a Category name
Choose a relationship name and a related Web Resource
Choose an attribute name and a specific value
Store in Semantic Support
End

Figure 5.1: Tagging Flow

$PropertyTag(X, PropName, OWebContent) \equiv$
$SemWiki(X) \leftarrow ([[PropName :: OWebContent]])$

- **Attribute Semantic Tag.** Let X be the URL identifying a web resource and let AtrName be the label of an attribute and AValue can be a literal or numerical value. $AttributeTag(X, AtrName, AValue) \equiv$ $SemWiki(X) \leftarrow ([[AtrName := AValue)]]$

The table 5.1 summarizes the mapping rules. The left hand is the *Semdrops* semantic tags, in the middle the equivalent SMW and the right the equivalent in owl-like.

5.1.3 How to Use the Model ?

In this example, we consider a user who is reading the *Paris* city page[1] on Wikipedia. She would like to tag that Paris is a City and also that it is the location of the movie Amelie.

Following the workflow in figure 5.1, first she has to decide which kind of semantic tags she wants to use. She uses a *Category Tag* to describe *Paris* as a *City*. For that, the literal *City* is used as the name of the Category Tag. Next, she decides to use *Property Tag* to describe the relationship between *Paris* and the film *Amelie*. A *Property Tag* requires a label to type the relationship and also a URI for two involved resources: the movie and the city. Therefore the string *was movie location of* types the relationships and the

[1] http://en.wikipedia.org/wiki/Paris

Amelie Wikipedia's page represents the movie. Figure 5.2 (a) shows the initial wikipedia page and the figure 5.2 (b) shows the tagged page. The left side of the screen shows the semantic tags for the current page. The semantic tags tree has three main labels: Category, Links and Attributes. The user can visualize the semantic tags grouped by these three types and can fold and unfold the list of each group of tags.

During the semantic tagging activity, *Semdrops* maps the semantic tags into the semantic support model of SMW by using the mapping rules detailed in the previous section. All the new meta-data is generated without modifying the original resource.

In this scenario, a new semantic relation is created between *Paris* city and the *Amelie* movie, there was no prior relationship between these two resources nor a navigational link.

Semdrops as Mashups Tool

The previous scenario shows how semantic tags can be added easily to a web resource. For instance, the semantic tags describing the web page of Paris in Wikipedia could be associated with those defined in the Paris wiki page at *SemanticWeb.org* [2]. In order to combine different sources, *Semdrops* includes the ability to access linked semantic data in cross semantic repositories. It takes special attention in owl:sameAs tags in the RDF definition of the web resource. For example, if in the RDF description of a web resource the tag *owl:sameAs:: targetResource* appears, the semantic tags of the targetResource will be added as semantic tags of the original resource.

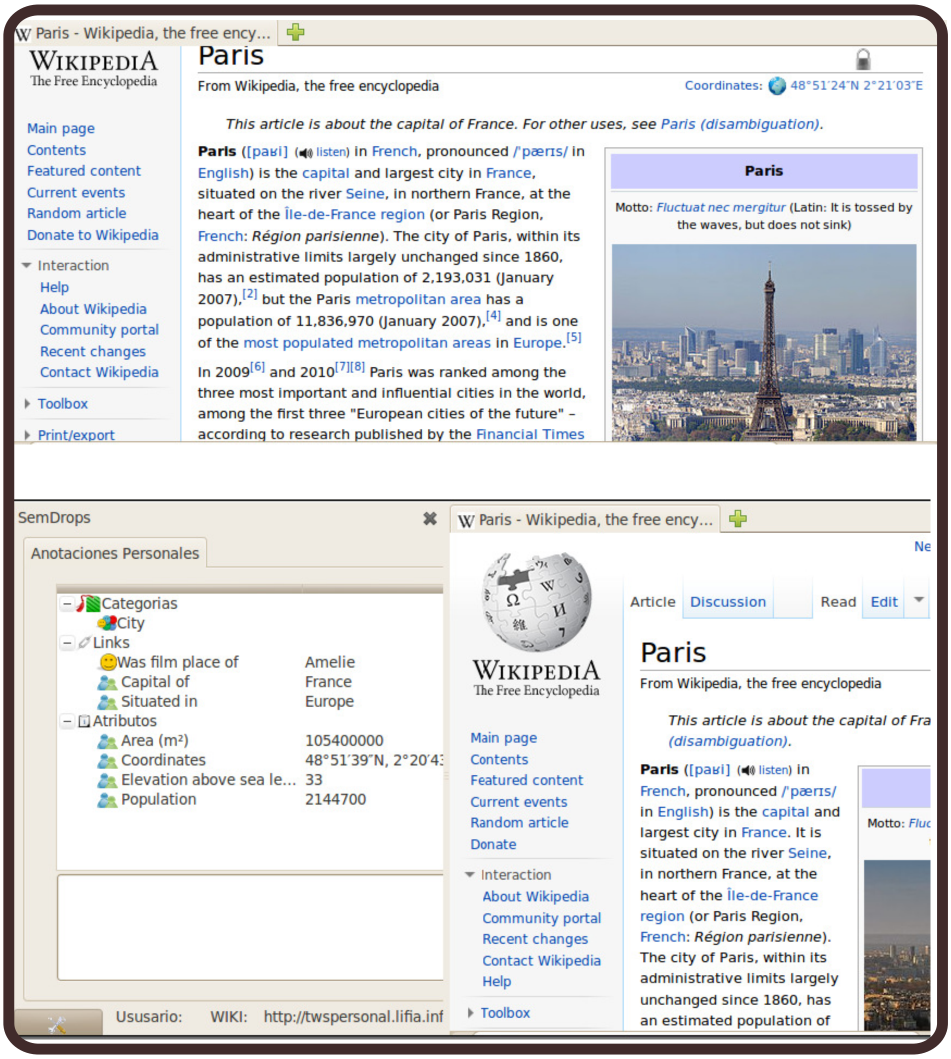
Figure 5.2 (b) shows the use of *Semdrops* with the semantic tags of the example in the section 5.1.3 with a cross reference to the Paris page on *SemanticWeb.org*. In this example, the semantic tags with the green icon are from the cross reference, and the category *City* has a special icon because it was produced in both semantic repositories: SMW repository and the *SemanticWeb.org* repository.

5.2 Implementation

Semdrops uses a Model-View-Controller architecture as it is detailed in the figure 5.3. It includes a semantic MediaWiki as a semantic support for storing and retrieving the semantic tags.

Semdrops is implemented as a Firefox sidebar add-on, it is available as an open source and can be downloaded from `http://sourceforge.net/projects/semdrops/`. It implements the semantic social tagging model. The semantic tags tree has three main labels: Category, Links and Attributes. The user can visualize the semantic tags grouped by these three types and can fold and unfold the list of each group of tags as shown in the figure 5.2. *Semdrops* implements drag & drop functionality, this improves user interaction and facilitates semantic tags generation. For instance, in the example of figure 5.2, a user can drag the *France* link and drop it on the Links label. This action

[2] SemanticWeb.org is a Semantic Mediawiki used mainly for content related with Semantic Web



Paris

From Wikipedia, the free encyclopedia

Coordinates: 48°51′24″N 2°21′03″E

This article is about the capital of France. For other uses, see Paris (disambiguation).

Paris ([paʁi] (listen) in French, pronounced /ˈpærɪs/ in English) is the capital and largest city in France, situated on the river Seine, in northern France, at the heart of the Île-de-France region (or Paris Region, French: *Région parisienne*). The city of Paris, within its administrative limits largely unchanged since 1860, has an estimated population of 2,193,031 (January 2007),[2] but the Paris metropolitan area has a population of 11,836,970 (January 2007),[4] and is one of the most populated metropolitan areas in Europe.[5]

In 2009[6] and 2010[7][8] Paris was ranked among the three most important and influential cities in the world, among the first three "European cities of the future" – according to research published by the Financial Times

Paris

Motto: *Fluctuat nec mergitur* (Latin: It is tossed by the waves, but does not sink)

SemDrops

Categorias	
City	
Links	
Was film place of	Amelie
Capital of	France
Situated in	Europe
Atributos	
Area (m²)	105400000
Coordinates	48°51′39″N, 2°20′4
Elevation above sea le...	33
Population	2144700

Ususario: WIKI: http://twspersonal.lifia.inf

Paris

From Wikipedia, the free encyclopedia

This article is about the capital of Fra (disambiguation).

Paris ([paʁi] (listen) in French, pronounced /ˈpærɪs/ in English) is the capital and largest city in France. It is situated on the river Seine, in northern France, at the heart of the Île-de-France region (or Paris Region, French: *Région parisienne*). The city of Paris, within its administrative limits largely unchanged since 1860, has an estimated population of

Motto: *Fluc*

Figure 5.2: Semantic Tags for Wikipedia's Paris page: (a) At the top, the Wikipedia's Paris page without the Social Semantic Tags. (b) At the bottom, *Semdrops* in action for the same Wikipedia page.

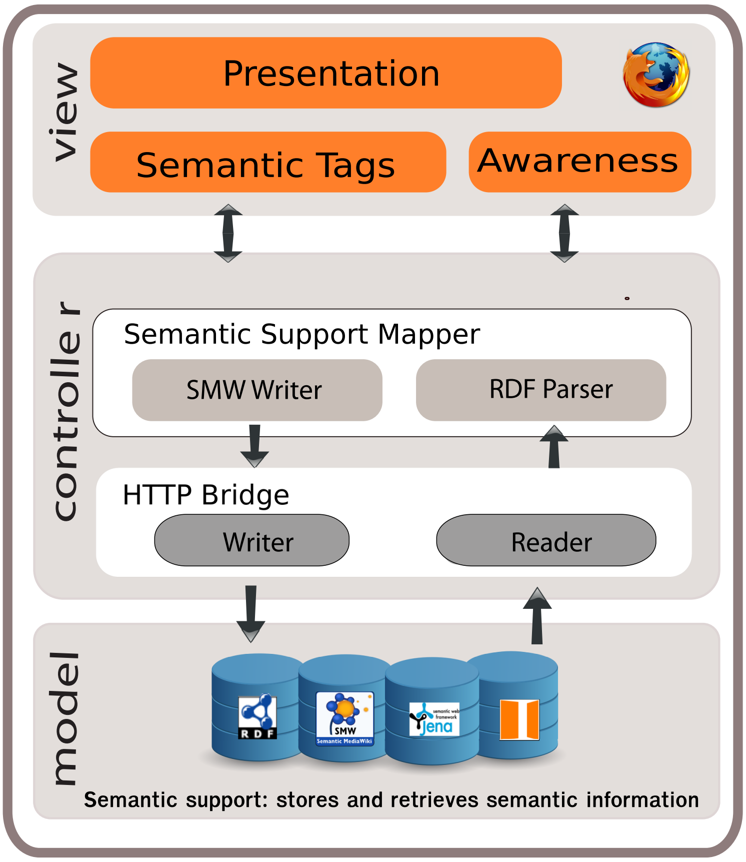


Figure 5.3: Semdrops Architecture

opens the new link dialog box filled with the France page link. Drag & drop functionality is available for links and texts in the Web resource.

6

P-Swooki and Semdrops Evaluation

Contents

6.1 P-Swooki Evaluation 51

6.1.1 Method 52

6.1.2 Results 52

6.1.3 Discussion and Learned Lessons 53

6.2 Semdrops Evaluation 55

6.2.1 Method 55

6.2.2 Results 56

6.2.3 Usability Evaluation 58

6.2.4 Discussion and Learned Lessons 58

6.3 Summary of this Part 59

6.1 P-Swooki Evaluation

In this section, we present the evaluation of the P-Swooki approach. In order to analyze the usefulness of the personal semantic annotations we run an evaluation study. In this study we wanted to answer the following questions:

- Are personal semantic annotations useful for any users?
- Do personal annotations augment the shared knowledge?
- Do participants found both kind of annotations could help to make better knowledge retrieving?

6.1.1 Method

We have conducted two separate experiments, one in France and another one in Argentina. The total number of participants were 15 people. The participants ranged in age from 25 to 45. All participants were involved in computer science, all were familiar with wikis and 5 of them were familiar with semantic wikis and have some experience in ontology building. The participants were in different rooms and they were not allowed to communicate to each other during the experience.

We started the first experience in France with a short explanation about semantic wikis, shared knowledge and personal knowledge. We asked participants to develop a semantic wiki by using both kinds of annotations. They started with a non empty wiki. In fact, two semantic wikis pages were created in advance; one about *Semantic Wiki* and another one about *Semantic Web*. We also suggested participants should use special syntax in order to control vocabulary explosion as it occurs in folksonomies [80].

In order to consolidate the first experience, we repeated the same experience in Argentina which confirmed the results previously obtained in France. These experiences show a preliminary evidence of the contribution of our approach regarding the usability of personal semantic annotations and their complementarity with the shared knowledge. In the following of this section we show the results of these experiences. As both experiences showed the same outcome, we will only present the results from France.

6.1.2 Results

The tables 6.1 and 6.2 show the type (individual or category) and the amount of personal semantic annotations that each participant has added to the *Semantic Wiki* and *Semantic Web* wiki pages respectively.

Table 6.1: Personal Semantic Annotation for *Semantic Wiki* Page

User	Individual	Category	Total
1	1	1	2
2	1	0	1
3	2	5	7
4	6	5	11
5	2	1	3
6	4	3	7
7	1	0	1

The *Semantic Wiki* page was annotated by all the participants. They annotated this page as individual 17 times and as category 15 times. The most active participant added 11 personal semantic annotations to this page. The average number of annotations per participant was 4.5. The average without the most active participant was 3.5.

The *Semantic Web* page was annotated by all the participants. They annotated this page as individual 8 times and as category 9 times. The most

Table 6.2: Personal Semantic Annotation for *Semantic Web* Page

User	Individual	Category	Total
1	0	1	1
2	1	0	1
3	0	2	2
4	6	5	11
5	1	1	2
6	1	0	1
7	1	0	1

active participant added 11 personal semantic annotations to this page. The average number of annotations per participant was 2.5. The average without the most active participant was nearly 1.

6.1.3 Discussion and Learned Lessons

The results above confirm our initial hypothesis about the usefulness of the personal semantic annotations because all the participants have added personal annotations.

The Table 6.3 shows the shared and personal semantic annotations used by the participants for the *Semantic Wiki* page.

The column *Shared annotations* regroups the shared semantic annotations. At the beginning of the experience, it was empty. This page could be annotated as a category such as *Category: PersonalInformationManagement* or as an object property such as *limitation : scalability*.

The second and third columns regroup all the personal semantic annotations. Notice that in some cases many users used the same semantic annotation. For instance, four users used *SemanticWeb* annotations and two users used *Wiki* annotations.

We can observe that the total number of semantic annotations is increased. Therefore, personal semantic annotations could be useful to augment the shared knowledge.

With this evaluation we learn the following lessons:

- Every participant used both personal and shared semantic annotations;
- Most participants said that it is easy to use personal semantic annotations, because it is not necessary to embed them into the text.
- Some participants had difficulties distinguishing between a category and an individual.
- All participants manifested the importance having a good user-interface to facilitate personal navigation.
- For some participants personal annotations were useful to structure their own navigational map according to their personal taxonomy.
- Personal annotations were easier for people not familiar with semantic wikis whereas someone familiar with semantic wikis did not see exactly the added value of personal semantic annotations.

Table 6.3: Personal and Shared Semantic Annotation for Semantic Wiki Page

Shared annotations	Individual	Category
Category: PersonalInformationManagement	ResearchTopic	SemanticWiki (4)
Category: KnowledgeManagement	SemanticWeb (4)	Wiki(2)
Category: SemanticWeb	CollaborativeTool(2)	SemanticWeb
Category:FormalLanguage	Web	Web (2)
KindOf: Wiki	NoDelete	CollaborativeTool
has: FactBox	Semantics	WebOfData
limitation : fault-tolerance	KnowledgeWeb	NoUndoTag
limitation : scalability	Something	WWW
limitation : censorship	Wiki (3)	CSCW
	Web	Semantic

- One participant did not understand the difference between personal and shared semantic annotations.
- For most participants, it was easier to add personal annotations when they were browsing and to add shared annotations when they were editing the wiki pages.
- Most participants found that combining both kind of annotations could help them to make better knowledge retrieving.

Although, it may be premature, the average of personal annotations shows a tendency: people feel comfortable using personal annotations and adding personal annotations is a complementary activity in semantic wikis.

These first results encourage us to continue in this direction, however, we need to conduct larger scale experiences to consolidate these results.

6.2 Semdrops Evaluation

To demonstrate the capability of *Semdrops* to produce adequate semantic data, we conducted an evaluation study. In this study, we asked a group of participants to use *Semdrops* to semantically tag a set of predefined web resources; more precisely, a set of Wikipedia pages. We also conducted a usability study to measure the satisfaction of participants regarding *Semdrops*.

In the following sections, we detail the evaluation methodology, the qualitative and quantitative analysis and the usability study.

6.2.1 Method

We have asked 20 people to participate in the evaluation, 15 males and 5 females. They are between 23 and 35 years old and fans of the *Lost* TV series [1], most of them are in the domain of computer sciences. In order to make the participants familiar with *Semdrops*, we provided them with a textual user guide and a screen-cast showing a using example[2].

Every participant was able to use *Semdrops* to add semantic tags to two specific web resources: the Wikipedia pages *John Locke (Lost)* [3] and *Richard Alpert (Lost)* [4], at any time and over four days. The participants were in different places and did not communicate with each other during the evaluation. Additionally, they only knew that other people were involved in the evaluation but they did not know who they were because we did not want the participants to communicate outside the system.

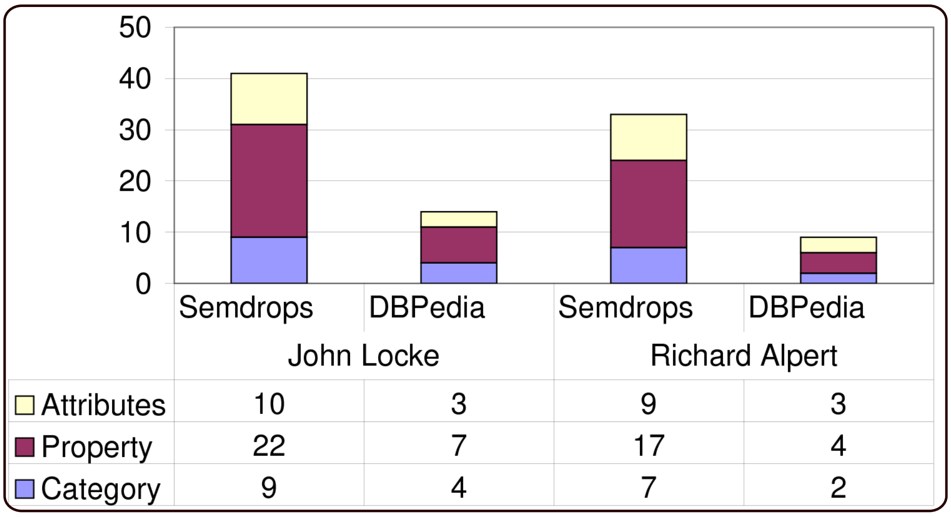
The participants were asked to read and analyze the wiki pages and after that to add the semantic tags if they considered them appropriate to semantically describe these wiki pages. They could also either delete or revise the semantic tags which they considered inappropriate. English language was used to name the tags in order to compare the results later with DBpedia.

[1] http://abc.go.com/shows/lost

[2] The screen cast is available at http://www.youtube.com/watch?v=N3Lr32iA9Vw

[3] http://en.wikipedia.org/wiki/John_Locke_(Lost)

[4] http://en.wikipedia.org/wiki/Richard_Alpert_(Lost)



	Semdrops	DBPedia	Semdrops	DBPedia
	John Locke		Richard Alpert	
□ Attributes	10	3	9	3
■ Property	22	7	17	4
■ Category	9	4	7	2

Figure 6.1: Quantities of semantic Tags

6.2.2 Results

Quantitative Analysis - Number of Coincident Tags with DBPedia

We manually compare *Semdrops* semantic tags with semantic information coming from the DBpedia database for the same web resources. Figure 6.1 represents a comparison between the set of the semantic tags added by *Semdrops* and that semantic information coming from the DBpedia database. Numbers in the figure represent the amount of semantic annotations. In order to make the manual comparison, we preprocessed the semantic information from DBpedia to fix it to the *Semdrops* conceptual model, for example, *skos:subject category: Fictional_Hunter* in DBpedia match to a *Semdrops Category Tag* called *Fictional Hunter*.

For both considered Wikipedia pages (*Jhon Locke* and *Richard Alpert*), we observed that:

- In general, the number of semantic tags added by the social effort using *Semdrops* was the double of the number of semantic tags that they automatically generated by DBpedia project.

- The most significant augmentation is the number of Property Tags. It was three times more in both Wikipedia pages.

Qualitative Analysis

This analysis is in charge of finding out whether the participants have used the *Semdrops* conceptual model properly and made the right conceptualisation. This reveals the right understanding of the meaning of the category, property and attributes primitives by the participants. To do this, we manually inspected the *Semdrops* log file and observed that:

- All categories were well defined. This means that participants did not have any problem with the use of categories. They use the category tag according to the definition given in the section 5.1.1.

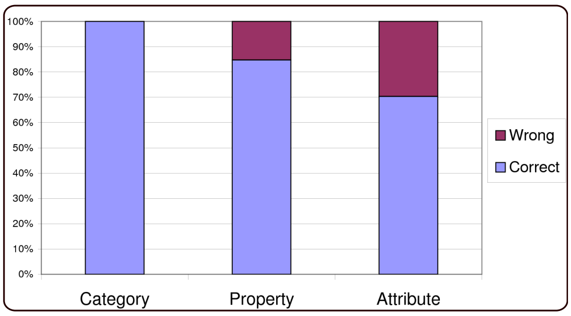


Figure 6.2: Quality of Semantic Tags

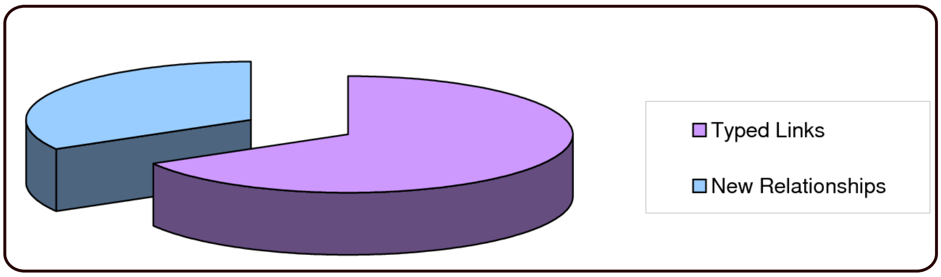


Figure 6.3: Proportion of Semantic Tags

- Sometimes participants confuse the use of property and attribute tag. For example, some participants use *birthday* with value *on May 30, 1956* as a Property tag instead of an Attribute tag.

- There were other cases where the right conceptualization might be controversial due to the fact that the information given in the wiki page was not enough (either because the abstraction level was not evident or the related concept was difficult to be expressed in a Wiki). Consequently, some attributes were not clear to be either an attribute or a property tag, we considered these cases also as not well defined.

To summarize, the rate of wrong property tags is near to 15 %, while it increases to 30% in the case of attributes as shown in the figure 6.2.

Added Value Analysis

By analyzing the kind of contribution of participants, we observed that:

- Most participants were worried about adding meaning to the navigational links, 67 % of the property tags were defined over existing navigational links. For example, the navigational link between the *John Locke* and the *Anthony Cooper* Wikipedia's pages was typed with *son of* property, representing that John Locke is the son of Anthony Cooper.

- In addition, new relationships among other resources were added. The 33 % of the added property tags establish new links with other Wikipedia pages. This result is illustrated in the figure 6.3.

It is important to notice that the enrichment in the existing content, in terms of both navigability and meaning, was possible to achieve by human interactions.

- At a first glance, the number of Categories defined with *Semdrops* seems to be higher than those defined in the initial Wikipedia pages. However, we later discovered that most of *Semdrops* categories already existed in Wikipedia. This was due to the fact that in Wikipedia the categories are organized in a hierarchy structure and only the most specific ones are shown in the wiki pages.

- Nearly 90 % of Categories defined with *Semdrops* were related to each other and could be organized in a hierarchy. Although, the subsumption property is part of the proposed semantic tag model, the current early version of Semdrops prototype does not support it. However, this result encourages us to extend *Semdrops* implementation to incorporate it.

To summarize, DBpedia has a good description of Category hierarchies and structural information of the page from Wikipedia´s infoboxes. On the other hand, *Semdrops* is better to make emerge semantic data that is not easy to obtain automatically. Both kinds of semantic data are highly complementary.

6.2.3 Usability Evaluation

At the end of the evaluation, we made a set of questions to measure the satisfaction of the participants with the system. We use the System Usability Scale (SUS) of Brooke [18]. SUS is a Likert scale simple questionnaire giving a subjective global value of usability. The obtained values indicate that the average of SUS usability was 71.5 over 100 with a standard deviation of 8.6. These results show that the participants had acceptable satisfaction of the system [6].

In addition, we asked users the level of comfort for making category, property and attribute tags. Most of the users had felt comfortable with the creation of categories and properties but less comfortable with the use of attributes.

6.2.4 Discussion and Learned Lessons

The conducted evaluation shows that semantic data obtained by using *Semdrops* are complementary with those of DBpedia. DBpedia has semantic data related with the structural content of the wikipages i.e. most properties reached in Wikipedia are related with navigational structure, for example *is dbprop:redirect of* or *is dbprop:disambiguates of* or infoboxes. While, *Semdrops* promotes semantic data proliferation related to the nature of the wiki article based on users background.

In addition, during the evaluation, we notice that users created properties following three main patterns :

- semantic enrichment of existing navigational links by adding semantic types.

- definition of new properties by using information in wikipage that were not related to a navigational link.
- emergence of new properties that relate pages that do not have any reference in the content of the wikipage.

Another interesting point was the generation of attributes. Attribute tags had the highest rate of mistakes. In most cases this happened because for the tagger the difference between an attribute and a property tag was not clear.

Finally, by the manual examination of the log file, we notice that some semantic tags were rectified by users. For example, the spelling of the word *bald* to describe *Jhon Locke* character was rectified twice by different users each time. There is a community regulation similar to that present in wiki community practices. In this evaluation, we made manual analysis of the log file. An automatic one could help to consolidate the generated information.

6.3 Summary of this Part

This part introduced the first hypothesis of this thesis: *Bootstrapping the collaborative knowledge production and a semantic tagging activity will improve the semantic description of web content.* To be more specific, the tools to complete the collaborative knowledge production and the definition of a formal semantic description where articulated by a social semantic tagging model.

The two complementary approaches that have been introduced in this part articulate human, methods and semantic tools to generate new Semantic Web enabled data. From the human side, *externalization*, *publication*, *internalization* and *reaction* define the general methods of the collaborative knowledge building process. In a first instance, these activities are integrated with P-Swooki and then, in a more general approach, with the social tagging action. Additionally, Semdrops provide a general ontology definition that allows users to describe web resources in a friendly manner by means of category, attribute or property tags. However, the semantic representation of the tags are by means of using a Semantic Web description. Notably, the fact of having human readable representation and, at the same time, a Semantic Web definition without extra efforts achieve the co-evolution of both words: Social and Semantic Web. This is potentiated within a process of knowledge production.

In conclusion, Semdrops improves the Semantic Web with lightweight ontologies by means of the social forces. These ontologies describe elements from the web that were not part of the Semantic Web. In the other direction, the use of Semdrops also improves the human activities. In this case, the use of personal spaces brings to the users a better navigation and structure of the web resources information.

III

From Semantic Web to Social Web

7

Introduction

This part of the thesis is dedicated to the develop of the second specific hypothesis: *Bootstrapping the machine support to obtain new pieces of knowledge from the Semantic Web, social methods and conventions will improve the information in the Social Web.* In this Part we present the use of Semantic Web to derive new pieces of knowledge that are not currently in the Social Web, and after that we can augment the Social Web by means of injecting the new pieces of knowledge.

The Semantic Web brings the ability to have better search and navigability on the Web. It is mainly built from meta-data extracted from the Social Web. DBpedia [12] is the best example. DBpedia knowledge base is built from data extracted from Wikipedia[1] infoboxes and categories. Despite DBpedia data are retrieved from Wikipedia, the semantic capacities of DBpedia enable SPARQL [75] queries to retrieve information that are not present in Wikipedia [93]. A SPARQL query that retrieves people born in a place[2] could include more people than those obtained by navigating from the place article in Wikipedia.

The case of DBpedia and Wikipedia is a good scenario to show the elements and tools from the Semantic Web that could be combined with other tools in order to improve the Social Web. With a SPARQL query it is possible to derive pieces of knowledge that are not part of the Social Web. In order to improve the Social Web it is desirable to add the derived elements. Similarly to the model from the first Part of this thesis, we need to translate the elements from the Semantic Web into a Social Web representation. Indeed, the concepts that are related by semantic properties in the Semantic Web have to be translated into hypermedia documents that are related by navigational links.

The first issue in this translation is that the meaning level of the hypermedia links are weaker than the meaning level of a semantic property: a link only allows users to navigate from one page to another page, and in the other hand a semantic property describes a meaningful relation between two well described concepts. The second issue is related to understand the social conventions that

[1] www.wikipedia.org

[2] A place could be a Country, Province, City or State.

the group of users follows to represent a semantic relation between a pair of pages.

In the co-evolution context, we combine the Semantic Web tools that allow us to obtain new pieces of knowledge that are not included in the Social Web, the tools from the Social Web used to represent the information (such as links, wikipedia articles and wikipedia categories), and the social conventions defined and followed by a community of users. In this case, the nexus among this elements is a recommender system that learn from a community of users the best manner to represent a semantic property. The recommendations are expressed in term of navigational paths which are sequences of Social Web documents related by hypermedia links. This approach was developed in order to improve Wikipedia with DBpedia.

In this part of the thesis, we present the BlueFinder algorithm that recommends the Wikipedia convention that better describes a DBpedia semantic property that relates a pair of Wikipedia articles. BlueFinder learns from other similar pairs how the Wikipedia community connects them, and then makes a set of recommendations to connect the former pair of articles. The connections are in terms of adding missing links in Wikipedia. BlueFinder follows a collaborative filtering approach [38].

This part of the thesis makes the following contributions:

1. **A measurement of the information gap between DBpedia and Wikipedia:** We computed the information gap between DBpedia and Wikipedia for a set of twenty representative DBpedia semantic properties.

2. **The BlueFinder algorithm:** We introduced the definition of BlueFinder algorithm as a collaborative filtering recommender system to predict the navigational path that best represents in Wikipedia a DBpedia semantic property between two Wikipedia articles. The navigational path respects the Wikipedia conventions. We also defined the **Semantic Pair Similarity (SPS) function** to measure the similarity between pairs of related articles based on the DBpedia types description of the pair members. The SPS function allows us to select those connected pairs that best represent the Wikipedia conventions for an unconnected pair of articles.

3. **Evaluation of the approach:** We introduced an empirical evaluation of our approach to measure precision, recall, f1, and hit-rate over 20 DBpedia semantic properties.

This part builds upon our previous work presented at three occasions:

- In [93] and [89], we introduced the first approach to discovering conventions in Wikipedia to represent a DBpedia semantic property. In these articles we made a social evaluation with Wikipedia community.

- In [90] we introduced a preliminary approach to BlueFinder as a recommending system.

7.1 Outline of the Third Part

This part of the thesis is organized in the following chapters:

- Chapter 8 introduces the background information that gives support to the work described in this dissertation. Section 8.1 and Section 8.2 describe the context of the problematic by introducing Wikipedia and DBpedia. Particularly, Section 8.1 describes Wikipedia as a Social Web site, and the section also introduces the structure of Wikipedia, articles, categories, and the Wikipedia conventions. Section 8.2 introduces an overview of DBpedia and the details of mapping and query services. After that, Collaborative Recommender Systems are introduced in Section 8.3. Finally, relevant related work in the field of improving Social Web with Semantic Web is detailed in Section 8.4.

- Chapter 9 describes the information semantic gap between DBpedia and Wikipedia.The chapter includes the analysis of twenty DBpedia semantic properties where information derived from DBpedia is not included in Wikipedia.

- Chapter 10 includes the definition of the BlueFinder approach to detect the Wikipedia conventions that best represent a DBpedia semantic property. This chapter begins with a description of the formal model of Wikipedia and DBpedia in Section 10.1. Then, Section 10.2 defines the Wikipedia pair connection rules that are used in this work. After that, in Section 10.3, the problem statement is introduced. Finally, Section 10.4 contains the details of the algorithm and the manner that it learns from Wikipedia the common conventions and then how it selects the specific one for a particular case of disconnected articles.

- Chapter 11 describes different evaluations to study the BlueFinder algorithm behavior to make recommendations over 20 different scenarios. The experimentation wants to answer the following questions: *What is the best setup combination to get the best accuracy from BlueFinder? Does BlueFinder retrieve path queries that can fix missing relations in Wikipedia? Does BlueFinder have a correlation between the confidence level and the accuracy of the predictions? Are there different conventions to represent in Wikipedia a DBpedia semantic property?*

8

Background

Contents

8.1 Social Web: Wikipedia 68

8.1.1 Articles 68

8.1.2 Categories 69

8.1.3 Wikipedia Conventions 72

8.2 Semantic Web: DBpedia 73

8.2.1 Data extraction 73

8.2.2 Data Query 75

8.3 Collaborative Recommender Systems 75

8.3.1 Implicit and Explicit Ratings 77

8.3.2 Similarity Functions 77

8.3.3 Number of Neighbors 78

8.4 Improving Social Web with Semantic Web 78

8.4.1 General Approaches 78

8.4.2 Adding missing links in Wikipedia 80

In this chapter, we provide the necessary context and background on research that motivates this part of this thesis. The approach of this part of the thesis presents a combination of a Collaborative Recommender System with DBpedia and Wikipedia. We briefly introduce Wikipedia and DBpedia as the main artifacts that represent the Social and Semantic Web respectively. After that, we describe the most important elements related to Recommender Systems and Collaborative Recommending Systems. Finally, we present a review of research on improving the Semantic Web with Social Web and the field related to adding missing links to Wikipedia.

8.1 Social Web: Wikipedia

Wikipedia is an open multilingual encyclopedia that is built by regular users on the Web. According to its own definition, Wikipedia *"is a collaboratively edited, multilingual, free Internet encyclopedia supported by the non-profit Wikimedia Foundation. Wikipedia's 30 million articles in 286 languages, including over 4.3 million in the English Wikipedia, are written collaboratively by volunteers around the world."*[1]. In Wikipedia any reader can freely edit any of the articles that appear in the encyclopedia without restrictions[2]. Wikipedia is a type of web sites called Wiki [50, 78]. A Wiki is easy to edit, write, and add links among the pages as we have mentioned in this dissertation. In Wikipedia any person is ready to participate in the edition.

The information of Wikipedia is mainly organized in Articles and Categories. An article is a page that has encyclopedic information on it. That means that in Wikipedia only relevant information is contained in it and it does not include other kind of information like dictionary definitions. The quality of Wikipedia articles varies in a range from low quality candidates for speedy deletion to high quality featured articles. The quality value is determined by elements like writing style, external academic references and structure of the information[3]. On the other hand, a Category is a group of articles that shares the category title topic.

8.1.1 Articles

An Article in Wikipedia is structured by a basic template: title, body, an infobox, and a list of categories that the article belongs to. Although only the title is mandatory, it is desirable to have the other parts with the most information possible. Figure 8.1 shows part of the *Lionel Messi* Wikipedia article.

- **Title:** It is mandatory, for any Wikipedia article and it has to be different from any other article in Wikipedia. The title also defines the article URI, for example Lionel Messi article has the `http://en.wikipedia.org/wiki/Lionel_Messi` URI. Indeed, any article in Wikipedia has a different URI to be accessed.

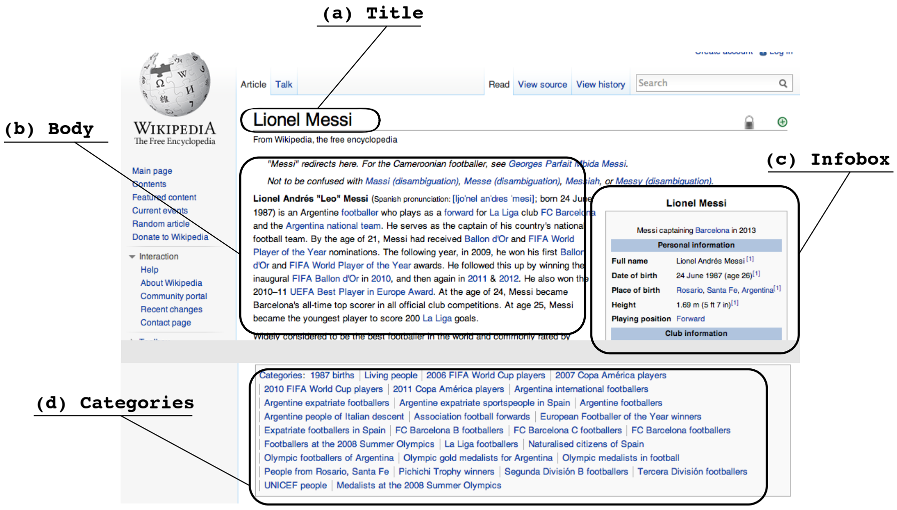
- **Body:** It is the content of the article. The body is a text document that is structured in sections and includes hypertext links to other Wikipedia articles or external resources. To write a link from a Wikipedia article *A* to another Wikipedia article *B*, a Wikipedia user has to add the wiki mark-up `[B]` in the body of the article A. If the article B exists in Wikipedia the link to B will be blue otherwise if B does not exist the link will be red colored.

- **Infobox:** *An infobox is a fixed-format table designed to be added to the top right-hand corner of articles to consistently present a summary of*

[1] `http://en.wikipedia.org/wiki/Wikipedia` on August, 9th 3013

[2] Only some sensible articles are protected by administrators or advanced users.

[3] `http://en.wikipedia.org/wiki/Wikipedia:What_is_an_article%3F` on August, 14th 2013



Lionel Messi

From Wikipedia, the free encyclopedia

"Messi" redirects here. For the Cameroonian footballer, see Georges Parfait Mbida Messi.

Not to be confused with Massi (disambiguation), Messe (disambiguation), Messiah, or Messy (disambiguation).

Lionel Andrés "Leo" Messi (Spanish pronunciation: [ljo'nel an'dres 'mesi]; born 24 June 1987) is an Argentine footballer who plays as a forward for La Liga club FC Barcelona and the Argentina national team. He serves as the captain of his country's national football team. By the age of 21, Messi had received Ballon d'Or and FIFA World Player of the Year nominations. The following year, in 2009, he won his first Ballon d'Or and FIFA World Player of the Year awards. He followed this up by winning the inaugural FIFA Ballon d'Or in 2010, and then again in 2011 & 2012. He also won the 2010–11 UEFA Best Player in Europe Award. At the age of 24, Messi became Barcelona's all-time top scorer in all official club competitions. At age 25, Messi became the youngest player to score 200 La Liga goals.

Widely considered to be the best footballer in the world and commonly rated by

Lionel Messi

Messi captaining Barcelona in 2013

Personal information	
Full name	Lionel Andrés Messi [1]
Date of birth	24 June 1987 (age 26)[1]
Place of birth	Rosario, Santa Fe, Argentina[1]
Height	1.69 m (5 ft 7 in)[1]
Playing position	Forward
Club information	

Categories: 1987 births | Living people | 2006 FIFA World Cup players | 2007 Copa América players | 2010 FIFA World Cup players | 2011 Copa América players | Argentina international footballers | Argentine expatriate footballers | Argentine expatriate sportspeople in Spain | Argentine footballers | Argentine people of Italian descent | Association football forwards | European Footballer of the Year winners | Expatriate footballers in Spain | FC Barcelona B footballers | FC Barcelona C footballers | FC Barcelona footballers | Footballers at the 2008 Summer Olympics | La Liga footballers | Naturalised citizens of Spain | Olympic footballers of Argentina | Olympic gold medalists for Argentina | Olympic medalists in football | People from Rosario, Santa Fe | Pichichi Trophy winners | Segunda División B footballers | Tercera División footballers | UNICEF people | Medalists at the 2008 Summer Olympics

Figure 8.1: Main structure of a Wikipedia article: Lionel Messi

some unifying aspect that the articles share and sometimes to improve navigation to other interrelated articles[4]. Wikipedia includes different infobox templates according to its nature i.e. person, football biography, etc . The infoboxes are used to normalize the summary of information among articles and also to make comparable different articles. Additionally, the infoboxes are used to extract metadata from the Wikipedia articles, and then used for example by DBpedia. The infoboxes are key-value structures. The key is a fixed term that represents one attribute of the kind of article where it is included and the value is the particular article value for that key. For example, in a person article the *Place of birth* will be valued with the Country or City where the person was born. Figure 8.2 shows a detail of Lionel Messi article infobox.

- **Category list:** Generally at the bottom of a Wikipedia article the list of categories the article belongs to appears. One article belongs to a category if it represents the kind of articles that are grouped in the category. Finally, an article can belong to several categories.

8.1.2 Categories

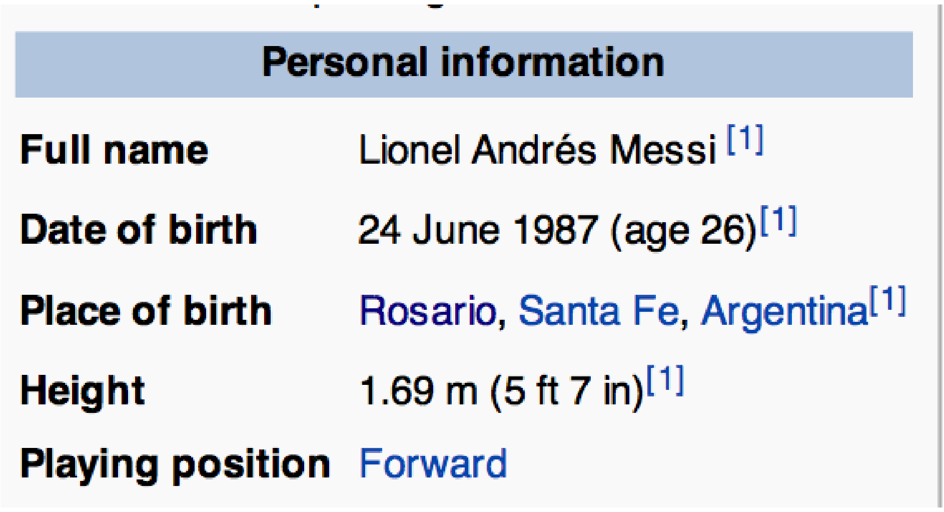
A Wikipedia Category[5] is a feature in Wikipedia that allows users to group articles that share a topic. *"Categories help readers to find, and navigate around, a subject area, to see pages sorted by title, and to thus find article relationships. Categories show how existing information is organised"*[6]. Categories allow representing one-to-many relations.

Categories allow organizing the Wikipedia articles in an indexed tree structure. In addition to articles, a category could also group other categories. The method to add an article, or another category to a category, is by means of

[4] urlhttp://en.wikipedia.org/wiki/Help:Infobox on August, 14th 2013

[5] Actually is a feature from Mediawiki that is the core system of Wikipedia

[6] http://en.wikipedia.org/wiki/Help:Category on August, 14th 2013



Personal information

Full name	Lionel Andrés Messi [1]
Date of birth	24 June 1987 (age 26)[1]
Place of birth	Rosario, Santa Fe, Argentina[1]
Height	1.69 m (5 ft 7 in)[1]
Playing position	Forward

Figure 8.2: Infobox detail

the wikimarkup `[Category:CategoryName]`. The consequence of adding an article to a category is the inclusion of the article in the article category list. In case a category is added to other category, the first will be a *sub-category* of the second one. The sub-category could include other categories, and so forth. This generates the tree organization. Figure 8.3 shows the page of *Rosario, Santa Fe* category.

The structure of a category is simpler than an article. A category has a title, a list of sub-categories, a list of articles that belong to it, and a category list, i.e. categories that belong to the current category it (super-categories).

- **Title:** It is mandatory and it has to be unique. A category is associated with a category page in the `Category:` namespace. This is done to differentiate a category page from an article page with the same title. For example, the title for the category `France` is `Category:France`.
- **List of sub-categories:** It is the list of links to all sub-categories of the current category. This list is automatically generated by Wikipedia.
- **List of articles that belong to it:** It is the list of links to all articles that belong to the specific category. It is also automatically generated by Wikipedia.

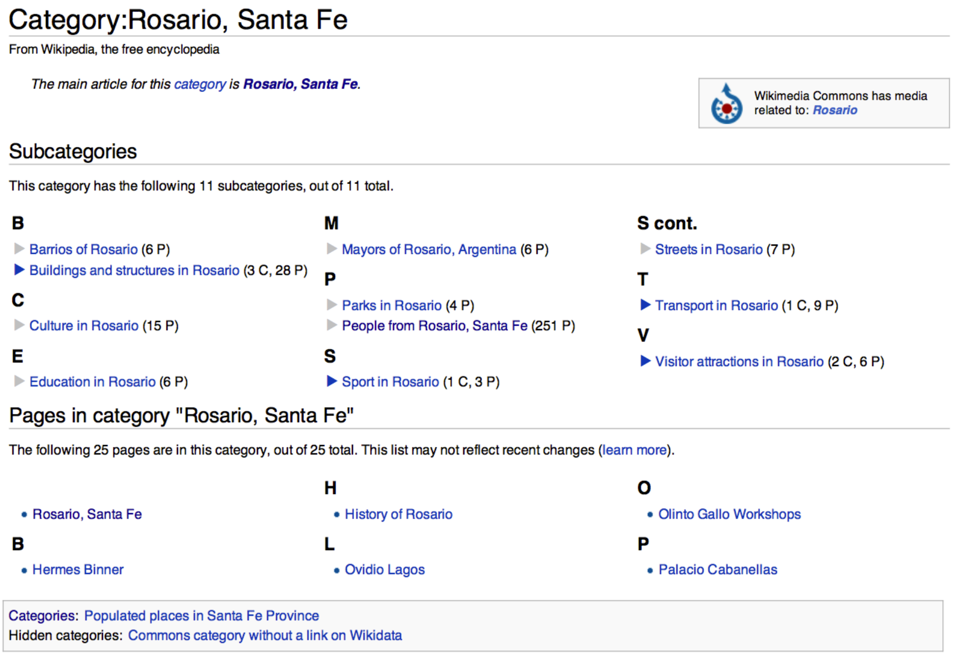
Stand-alone lists

"Stand-alone lists are articles the main components of which are one or more embedded lists or series of item"[7]. According to the Wikipedia conventions, the natural evolution of a *stand-alone list* is a *Category*

Wikipedia Category Tree

Categories in Wikipedia are organized as a overlapped tree thus, a category could be sub-category of more than one category. This generates a complex

[7] `http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Stand-alone_lists` on 2nd July, 2013



Category:Rosario, Santa Fe

From Wikipedia, the free encyclopedia

*The main article for this category is **Rosario, Santa Fe**.*

Wikimedia Commons has media related to: *Rosario*

Subcategories

This category has the following 11 subcategories, out of 11 total.

B

▷ Barrios of Rosario (6 P)
▶ Buildings and structures in Rosario (3 C, 28 P)

C

▷ Culture in Rosario (15 P)

E

▷ Education in Rosario (6 P)

M

▷ Mayors of Rosario, Argentina (6 P)

P

▷ Parks in Rosario (4 P)
▷ People from Rosario, Santa Fe (251 P)

S

▶ Sport in Rosario (1 C, 3 P)

S cont.

▷ Streets in Rosario (7 P)

T

▶ Transport in Rosario (1 C, 9 P)

V

▶ Visitor attractions in Rosario (2 C, 6 P)

Pages in category "Rosario, Santa Fe"

The following 25 pages are in this category, out of 25 total. This list may not reflect recent changes (learn more).

- Rosario, Santa Fe

B

- Hermes Binner

H

- History of Rosario

L

- Ovidio Lagos

O

- Olinto Gallo Workshops

P

- Palacio Cabanellas

Categories: Populated places in Santa Fe Province
Hidden categories: Commons category without a link on Wikidata

Figure 8.3: Category in Wikipedia: Rosario, Santa Fe

category tree. In mathematical terms, the category tree is a lattice where the category `Category:Contents` is the top level category. In this organization, it is natural to think that a super category includes in its definition those pages that belong to its sub-categories. To support this assumption, the Wikipedia community advices users to categorize articles with the most specific category[8].

However, the organization of the category members is not always the same. According to the relations among the elements that belong to a category, the categories could be classified in three types: diffused, non-diffusing, and eponymous categories:

- **Diffused:** When a category has a large number of members it is desirable to organize them into more specific sub-categories. Splitting the category into several more specific sub-categories, and then moving the members to the new subcategories. However, a category could be partially diffused and have members that directly belong to it, and sub-categories.

- **Non-Diffusing:** They are subsets of a bigger category. According to Wikipedia *"They provide an exception to the general rule that pages are not placed in both a category and its subcategory: there is no need to take pages out of the parent category purely because of their membership of a non-diffusing subcategory"*.

- **Eponymous:** They are categories that cover the topics of an article. For example, `France` and `Category:France`.

Although categories in Wikipedia are organized in a hierarchy, according to Suchanek et al. [85], "Wikipedia category hierarchy is barely useful for

[8] `http://en.wikipedia.org/wiki/Wikipedia:Categorization` on August, 15th 2013

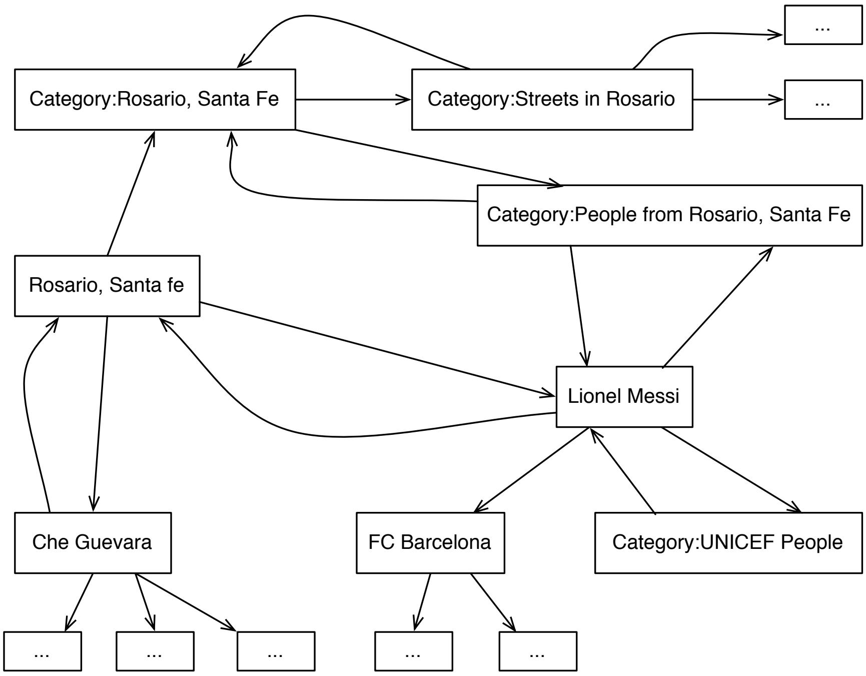


Figure 8.4: Example of Wikipedia Graph. Part of the links between Rosario, Santa Fe and Lionel Messi

ontological purposes. For example, Zidane is in the category `Football in France`, but Zidane is a football player and not a football". Consequently, it is not possible to apply a semantic approach to discover the best category for a particular article or even, analyze semantic relationships in the category tree.

8.1.3 Wikipedia Conventions

As was introduced above, Wikipedia includes a lot of conventions in different topics like: article information, writing style, naming, categorization, etc. Wikipedia is a social software, thus also the conventions are a social building artifact. Several of those conventions are written in Wikipedia articles identified with the prefix *Wikipedia*, and others appear as a common practice such as in comments[99]. *Categorization*[9], *Categories, lists and navigation templates*[10] are examples of these pages. These Wikipedia articles define the bases of an agreement to select the name of the categories, to split a category into subcategories, to define the number and kind of links a page may have and to improve Wikipedia in a general way.

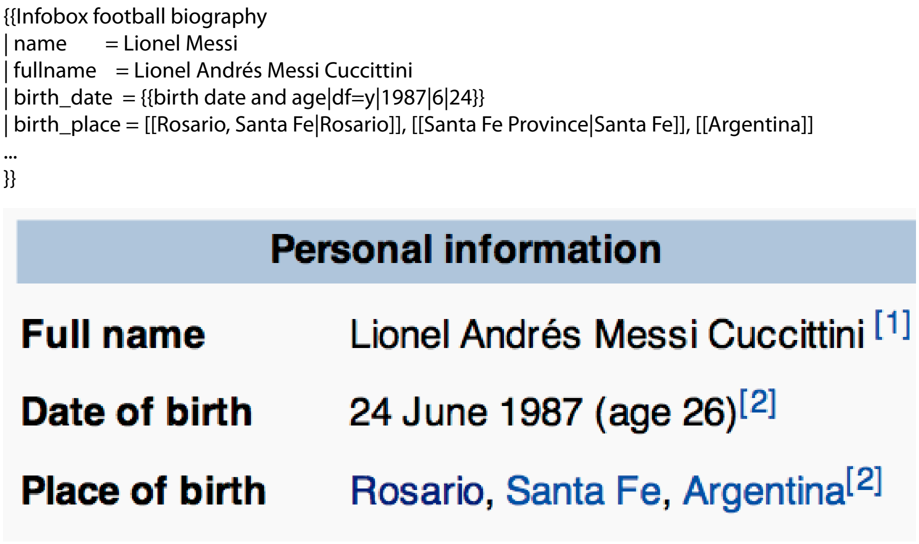
Some examples of conventions in Wikipedia are:

- **Page Title:** Wikipedia recommends to use the name that is most frequently used to refer to the subject rather than the "oficial" name. Such as: *Bill Clinton* instead of *William Jefferson Clinton*. This is an explicit convention[11].

[9] http://en.wikipedia.org/wiki/Wikipedia:Categorization

[10] http://en.wikipedia.org/wiki/Wikipedia:Categories,_lists,_and_navigation_templates

[11] http://en.wikipedia.org/wiki/Wikipedia:Article_titles



```
{{Infobox football biography
| name       = Lionel Messi
| fullname   = Lionel Andrés Messi Cuccittini
| birth_date  = {{birth date and age|df=y|1987|6|24}}
| birth_place = [[Rosario, Santa Fe|Rosario]], [[Santa Fe Province|Santa Fe]], [[Argentina]]
...
}}
```

Personal information	
Full name	Lionel Andrés Messi Cuccittini [1]
Date of birth	24 June 1987 (age 26)[2]
Place of birth	Rosario, Santa Fe, Argentina[2]

Figure 8.5: Example of Wikipedia Infobox Template and its rendered output.

- **Category Name:** Topic categories must have a name in singular, and set categories a plural name[12]. For example: *Law* is a topic category and *Writers* is a set category. This is an explicit convention.

- **Categorization specificity:** A general rule to categorize a Wikipedia page is to use the most specific category that represents the page. In the other hand, it is desirable that a category contains several pages rather than few pages. These convention rules generate that the specificity level of the categories change according to the development context of the articles that it describes. For example, writers from London are grouped with the specific category *Writers from London* , on the contrary, the writers from *Pisa* are only grouped by the general category *People from Pisa* which also includes people with different occupations. This conventions are not written, in consequence are more difficult to detect.

8.2 Semantic Web: DBpedia

DBpedia[12] is a community effort to extract structured information from Wikipedia, and to make this information available in the Semantic Web. The DBpedia project is divided into two main purposes: translating the Wikipedia information to RDF repositories, and offering query tools to manipulate the repositories.

8.2.1 Data extraction

As we described above, additionally to free text, Wikipedia articles include structured information such as infoboxes and wikimarkups (i.e. [Category: CategoryName]). Mediawiki, the software that runs Wikipedia, stores its in-

[12] http://en.wikipedia.org/wiki/Wikipedia:Categorization#Naming_conventions

formation into relational databases. Dumps of Wikipedia relational bases are published periodically on the Web.

The DBpedia project maps Wikipedia structured information to RDF using two different methods: (1) They map the relations that are stored in the Wikipedia dumps onto RDF, and (2) the additionally extract structured information from the articles text like wikimarkups and infoboxes.

DBpedia is a community effort because the mapping rules to translate the information from Wikipedia to RDF are defined in a social process. This effort could be noticed with the DBpedia Mappings Wiki[13]. In this wiki the registered users could define mappings rules, such as translating a Wikipedia infobox into a set of RDF triplets.

The extraction of the semantic information from Wikipedia is illustrated with the infobox template *football biography*. The example illustrates the translation of Lionel Messi infobox information, shown in Figure 8.5, to RDF triplets by using the mapping rule shown in Listing 8.1[14]. The mapping will extract the type information (SoccerPlayer), the name of the player, and the player's place of birth. Therefore, the RDF triplets generated by applying the mapping will be:

- `dbpedia:Lionel_Messi rdf:type dbpedia-owl:SoccerPlayer`
- `dbpedia:Lionel_Messi foaf:name "Lionel Messi"@en`
- `dbpedia:Lionel_Messi dbpedia-owl:birthPlace dbpedia:Rosario,_Santa_Fe`
- `dbpedia:Lionel_Messi dbpedia-owl:birthPlace dbpedia:Santa_Fe_Province`
- `dbpedia:Lionel_Messi dbpedia-owl:birthPlace dbpedia:Argentina`

DBpedia maps each Wikipedia article as a DBpedia resource. The title of a Wikipedia article will be the name of the DBpedia resource. In the example, the DBpedia resource for *Lionel Messi* article will be `dbpedia:Lionel_Messi`. Additionally, the mapping rules generate triplets where the subject of all of them is the DBpedia resource of the Wikipedia article that contains the infobox template. In the example, the subject of all triplets is `dbpedia:Lionel_Messi`.

```
{{TemplateMapping
...
{{Condition
| operator = otherwise
| mapping = {{TemplateMapping | header = 0 | mapToClass =
   SoccerPlayer }}
}}
| mappings =
{{PropertyMapping | templateProperty = name | ontologyProperty =
    foaf:name }}
{{PropertyMapping | templateProperty = birth_place |
   ontologyProperty = birthPlace }}
...
```

[13] http://mappings.dbpedia.org

[14] The complete mapping rule is in http://mappings.dbpedia.org/index.php/Mapping_en:Infobox_football_biography

```
}}}
```

Listing 8.1: Portion of mapping rule for football biography infobox template

8.2.2 Data Query

DBpedia project provides three access mechanisms to exploit the semantic information of the RDF repositories:

- **Linked Data:** The information of each resource can be accessed by an HTTP request to a unique URI, for example `http://dbpedia.org/resource/Lionel_Messi`. The system retrieves detailed RDF information of the resource when the request was made by a Semantic Web Agent, and an HTML page with the same information if the request was done by a regular Web browser. These policies respect the linked data principles[9, 11]

- **SPARQL Endpoint:** It is possible to query the DBpedia repositories by using SPARQL queries. DBpedia project provides an end-point that receives SPARQL queries and retrieves the result in RDF. This service is useful when a developer needs a small portion of the DBpedia resources. For example, applying the SPARQL query in Listing 9.1 in DBpedia SPARQL endpoint will retrieve a list of pairs of DBpedia resources (City,Person) in which (Rosario,_Santa_Fe, Lionel_Messi) will be included.

- **RDF Dumps:** DBpedia periodically generates N-Triple dumps of its semantic repository. The dumps are populated in the Web, and can be downloaded and used as a local version of DBpedia. This mechanism is useful when a developer needs a big portion of the DBpedia base.

In this thesis we have used the SPARQL Endpoint and the RDF Dumps mechanisms.

8.3 Collaborative Recommender Systems

Collaborative Recommender Systems are a specific family of Recommender Systems. In a introductory explanation, Recommender Systems (RS) are software tools that provide recommendations for items. Some definitions are:

- *"Recommender systems are intelligent applications which assist users in their information-seeking tasks, by suggesting those items (products, services, information) that best suit their needs and preferences."* [54]

- *"Recommender systems assist and augment the natural social process of relying on recommendations from other people"* [77].

Several famous web sites use, with an important role, Recommender systems: Amazon.com [52] by recommending books, Youtube[15] (videos), IMDB[16] (movies) and Netflix[17] (movies and TV-shows).

[15] www.youtube.com

[16] www.imdb.com

[17] www.netflix.com

The main problem of a recommender system can be simplified in estimating a rating for items that have not been seen by a user[3]. The new ratings of the unseen items can be estimated using different approximation techniques. Recommender systems are usually classified into the following categories:

- **Content-Based recommendations:** the user is recommended items based on the user preferred items in the past. For example, to recommend a movie to a user, the systems try to understand the common aspects of movies that the user has rated before.
- **Collaborative recommendations (Or collaborative filtering recommendations):** the user is recommended items based on those that people with similar interests and preferences liked in the past. In this dissertation we will use this method.
- **Hybrid approaches:** A combination of the previous ones.

Focus in the context of this dissertation, the collaborative knowledge production involves a social agreement in the definition of shared knowledge. Therefore, it is important to obtain the opinions of the users community that is involved in the knowledge production. In consequence, the kind of Recommender Systems that are more specialized in this approach are **Collaborative Recommender Systems**.

Collaborative Recommender Systems based their approaches into analyze information from the past behavior or the opinions of a community of users to predict which items the current user of the system will most probably like or be interested in [38]. According to the elements in which the prediction is made, the collaborative recommender systems could be divided into:

- **User-based nearest neighbor recommendation**: This approach is based on the idea that similar users will have similar tastes. Then, having an item rating database where each user rates a known element as input, the system identifies other users (nearest neighbors) that had similar preferences to those target users in the past. After that, for a particular item that the target user has not rated yet, the system predicts the rating for the item based on the ratings made by the neighbors of the target user.
- **Item-based nearest neighbor recommendation**: The main idea is to compute predictions using similar items instead of using similar users. User-based nearest recommendations could be hard to compute specially when it must handle millions of users and millions of catalog items. The difficulty resides in scanning a vast number of users before computing a recommendation and this task is complex to be computed in real time. The item-based nearest neighbor recommendation strategy is better to preprocess data and to minimize the real time computation.

The k-nearest neighbor (kNN) algorithm is a well known technique to be applied in User/item-based nearest neighbor recommendation. However, some elements have to be taken into account when it is used, such as evidence of the rating values, similar neighbors computation, and the number of neighbors to take into consideration. In the following sections we describe these three elements.

8.3.1 Implicit and Explicit Ratings

There is a diversity of techniques to obtain the user's opinions. Asking for explicit item ratings is one of the most precise, for example in IMDB movies rating. However, the rating activity is not always present as part of the system and an implicit rating is used. For example, in an ecommerce portal the action of buying a product could be considered as a positive rating.

8.3.2 Similarity Functions

Collaborative filtering is a similar set problem[76]. As we have noticed, finding similar items is the base of the approach. The similarity could be computed according to a wide variety of items or users characteristics and by applying diverse techniques, such as Jaccard Similarity to compare similarity on sets, Cosine similarity to compare text documents, Euclidean distance to measure different attributes in different values, and Minowski distance as a generalization of the Euclidean distance. A common approach is considering items as a document vector in a multi dimensional vector space and computing their similarity as the cosine angle among the vectors.

Additionally, in the field of using Semantic measure distance in Recommender System the work of Passant[74] introduces six different versions of the Linked Data Semantic Distance (LDSD). The LSDS compares resources in the LOD based on direct relations and indirect relations among resources. The semantic distance is defined in terms of C_d function. That function computes the number of direct and distinct links between resources in a Linked Open Data graph, similar to the degree of a path query in a path index for direct links of our work. The C_d can be used to compute the total number of direct and distinct links from a resource to other, and the different instances of a link to any other resource.

- **Direct Distance**: It counts the number of direct output links and the number of input links between semantic resources.
- **Indirect Distance**: It counts the number of indirect output links and indirect input links. Two resources are indirectly related if both have an outgoing link to a same third resource, and both resources have incoming links if they have the same incoming link type from a same third resource.
- **Combined Distance**: A combination of both previous distance measures.

Nunes et. al. introduced a recommender system to discover semantic relationships among resources from the Social Web[65]. The authors introduced two functions to measure resources interlinking. The first metric is called *semantic connectivity score (SCS)* and it is defined between a pair of resources in an undirected graph G with the following function:

Definition $SCS(e_1, e_2) = \sum_{l=1}^{r} \beta^l * |paths^{<l>}_{(e_1,e_2)}|$

where $|paths^{<l>}_{(e_1,e_2)}|$ is the number of paths between the entities e1 and e2 of length l, r is the maximum length of paths and $0 < \beta <= 1$ is a positive damping factor. Two relevant factors are in the SCS function by Nunes et. al., the first is that SCS gives a better score to short paths, that means that the connectivity of two resources will be higher in short distances than those that are far; the second factor is related to the length of the path in the evaluation study. According to the authors, they compute paths with a max length of 4, 1 less than in our approach. The other measure function presented by these authors is the $co-occurrence-basedmeasure(CBM)$. *"The CBM measure between entities that relies on an approximation of the number of existing Web pages that contain their labels"*. The definition of the CBM function is:

$$CBM(e_1, e_2) = \begin{cases} 0 & \text{if } count(e_1) = 0 \text{ or } count(e_2) = 0 \\ 1 & \text{if } count(e_1) = count(e_2) = count(0) \\ \frac{Log(count(e_1,e_2))}{Log(count(e_1))} * \frac{Log(count(e_1,e_2))}{Log(count(e_2))} & \text{,otherwise} \end{cases}$$

where $count(e_i)$ is the number of Web pages that contain an occurrence of the label of entity e_i, and $count(e_1, e_2)$ is the number of Web pages that contain occurrences of the labels of both entities.

8.3.3 Number of Neighbors

Adding all the users in the neighborhood not only is a negative aspect in the time spent calculating but it also has an impact on the accuracy of the recommendations. This is due to the fact the ratings of other users who are not really comparable with the target user could be taken into account. KNN technique only collects the k nearest neighbors in order to reduce the number of neighbors taken into consideration.

8.4 Improving Social Web with Semantic Web

Several relevant works are dedicated to improve the Social Web by means of Semantic Web. In this section we enumerate the most relevant work on this area. The section is divided into two: Section 8.4.1 enumerates general approaches that use the Semantic Web technologies in order to improve the Social Web. Then, Section 8.4.2 enumerates specific related works in the context of improving Wikipedia by means of adding links.

8.4.1 General Approaches

$MORE$[60] is a recommender system that uses DBpedia and LinkedMDB [33] to recommend movies in a Facebook application. Their goal is to collect relevant information about user preferences in order to provide a personalized recommendation of films. Once the application is installed in the Facebook account, the user searches a movie by typing the first characters and the system suggests from DBpedia a list of the top movies that start with the written text. PageRank [71] algorithm was adapted to DBpedia movies sub-graph in order to do this. Having a list of selected movies, the system computes the recommendation of the top 40 movies related to the selected one. In the recommendation part a k-NN algorithm is used in combination with a Vector Space Model approach. In VSM non-binary weights are assigned to index terms in queries and in documents (represented as sets of terms), and are used to compute the degree of similarity between each document in the collection and the query. In their approach, the authors made a semantic version of VSM to deal with RDF graph. The complete RDF graph is represented in a 3D tensor where each slice represents a semantic property. The similarity between two movies is computed by means of the correlation between two vectors and it is quantified by the cosine angle between them. After that, to recommend a movie m to a user u, MORE computes the similarity of movies taking into consideration the user profile.

Di Noia et al. [22] also introduced a creation of a content-based recommender system exploiting exclusively linked open data datasets to be used in the field of movie recommendations. That is the main difference with the previous work detailed before. The system consumes data from DBpedia, LinkedDBM and Freebase [13]. Their approach is a Content Based recommender system that based the similarity between concepts by means of adapting the Vector Space Model to the LOD-based data. They represent the RDF graph in a 3D matrix where each slice refers to an ontology property, as in the previous work. In this case, the authors define the ratings of a movie with a binary scale, so for a user the rating for a movie could be that he or she likes it or dislikes it. Then, a user profile is maintained by a list of movies and its ratings, and it is used to generate the recommendation for an unrated movie. In this new approach, Di Noia et al., two similarity functions are presented and both the similarity among a particular semantic property called α. A weight to α is assigned according to the importance of the property according to the user profile. For example, a user u would like to see another movie with the same genre that a previous one but other user would like a movie with the same cast. To compute the α value, the system is trained with a genetic algorithm and it is compared with an alternative using Amazon's collaborative recommender system.

In both cases, Di Noia et al. used DBpedia as a source data set to base the recommendations. However, the improvement in the Web of Data is not exploited to improve the content of the Social Web. The recommendation is used to offer a service to a group of users but the computed information is not stored and re-used in the Social Web, therefore the computation is lost.

The work of Yan Want et al. [98] introduces a collaborative annotating approach to automatically recommend suitable categories to a Wikipedia article. For example, when a user finished editing a new article in Wikipedia he or she would like to know "Which Categories are suitable for it?" The approach consists of a two-step model. The first step is to find for a target article some evidence from similar articles to propose a proper category. In the second step, the approach uses a k-NN algorithm to sort the evidence. Finally, the top ranked categories in the evidence will be retrieved to the user. According to the authors, the recommendation is done by using the Semantic Web perspective, even when they are not using resources apart from Wikipedia i.e. DBpedia from the Semantic Web. The evidence generation analyzed Wikipedia information in order to infer the relevant data to be used, as in [60], Vector space model. In this case, the authors choose four semantic features:

- **Incoming links**: given a Wikipedia article d, collect all the titles of articles that include a direct link to d.
- **Outgoing links**, given a Wikipedia article d, collect all the titles of

articles that are direct linked from d.

- **Section headings**, given a Wikipedia article d, collect all the section headings in d.

- **Template items**: Given a Wikipedia article d, collect all the template item names in d. This is similar to the template mapping in DBpedia but without using RDF semantic properties.

With these four fields, the work of Yan Want el al. builds a representation of the Wikipedia article, compares it with other articles using the TF-IDF (term frequency–inverse document frequency), and then it retrieves a ranked list with the top N similar articles. Finally, the second part in the approach of Want et al. consisted in sorting the candidates Categories in a ranking. They define five ranking functions:

- **Direct Count**: The more popular is the category, the higher rank it should have.

- **Weighted count**: categories provided by more similar evidence will be more important.

- **Boosted Weighted Count**: It modifies the importance by adding a booted coefficient based on the ranking and the importance of the evidence.

- **Global Popularity Rewarded Boosted Count**: Based on the previous ranking, it includes the popularity of the category

- **Global Popularity Punished Boosted Count**: On the contrary, the popularity of the category is punished.

The work of Panchenko et al. [72] introduces an approach to extract semantic relations between concepts in Wikipedia applying NPL and KNN algorithms called Serelex. In a first step, Wikipedia article texts are preprocessed in order to remove markups tags and special characters and after that, a lemmatization and text tagging is performed. After that, Serelex receives a set of concepts and returns sets where articles are semantically related according to Wikipedia information and using Cosine and Gloss overlap distance functions. In addition to the lack of using DBpedia as semantic base, Serelex cannot describe the way that two concepts are related in Wikipedia according to a semantic property. Serelex is in the line of NPL systems and the semantic relations that can be inferred are not published in a proper semantic language as RDF or OWL and it can not be used by other Semantic Web enabled projects.

8.4.2 Adding missing links in Wikipedia

In the field of improving Wikipedia information by adding missing links, several related works can be mentioned.

The work by Adafre et al. [2] was involved in the task of adding missing links in the first paragraph of a Wikipedia article based on similar articles according to LTArank (Ranking based on Links and Titles). The problem was enunciated as:

"Given a page d, and the pages most similar to d according to LTRank, we extract suggestions for links missing from d from the similar pages; the final step, then, is to identify links that are actually missing from d."

In order to detect the links, the approach of Adafre et al, consists of a two step process, the clustering pages step and the finding links step.

To generate the cluster of similar pages, during the first step the algorithm generates, for any page *d*, a ranked list of similar pages. However, the number of similar pages can be extremely large and it must be filtered. The second step includes a filtering mechanism. Thus, having two pages with their list of related pages, if both pages are similar they expect there to be a high overlap between their corresponding set of related pages.

After the clustering part, the algorithm has to detect the missing links. The authors considered that if a page is similar to another, both share the same outgoing links structure. So, for the target page that has to be completed with new links, the process detects those links and anchor texts that are missing in the target page and, if the target page includes some anchor texts from the related page, the algorithm replaces its with an outgoing link as in the related page.

As we can see, the work by Adafre et al., inserts direct links from a Wikipedia article to another Wikipedia article. In it work, categories are not analyzed as structural information.

The work of Sunercan et al. [86] is aimed at fixing missing direct links in Wikipedia based in the Adafe et. al. work. They assumed that related articles contain related links. The main difference from Adafe is that in Sunercan et al. work diverse sources of related articles are investigated instead of using the LTRank. The different sources from Wikipedia are:

- Articles in the same category.
- Articles linked by the target article.
- Articles linked to the target article.
- Index Search for common links.
- Index Search for Link Term Occurrence in the Text.

Hoffman et al. [35] introduced an approach to complete Wikipedia infobox links with information extracted from Wikipedia by using Kylin in a combination of community content creation and information extraction. Hoffman introduces the concept of having a synergistic pairing of these two approaches. Kylin is an information extraction system that was trained, in this case, with Wikipedia. In the first step, Kylin obtains training data by analyzing existing infoboxes data in Wikipedia articles. Each infobox has a class and Kylin collects examples of articles from a particular class and detects the infobox property entry in the Article text. Next, Kylin selects the best phrase in the article that contains the infobox field.

After that, the system is able to recommend users the missing property entry in an incomplete infobox. In order to combine the automatic process from Kylin with the social support, the authors enhance Wikipedia user interface to make explicit the desire to complete infoboxes and help users with Kylin recommendations. The particular element in Hoffman et al. work is that they do not use Semantic Web features.

9

A Semantic Information Gap Between DBpedia and Wikipedia

DBpedia [12] knowledge base is built from data extracted from Wikipedia[1] infoboxes and categories. Figure 9.2a schematizes information flow between DBpedia and Wikipedia. Despite DBpedia data are retrieved from Wikipedia, the semantic capacities of DBpedia enable SPARQL [75] queries to retrieve information that are not present in Wikipedia [93]. A SPARQL query that retrieves people born in a place[2] could include more people than those obtained by navigating from the place article in Wikipedia.

```
PREFIX db−owl:<http://dbpedia.org/ontology/>
PREFIX db−p:<http://dbpedia.org/property/>

SELECT ?from, ?to WHERE{
?db_from a db−owl:Person.
?db_to a db−owl:Place.
?db_to db−p:birthplace ?db_from.

?db_from <http://xmlns.com/foaf/0.1/isPrimaryTopicOf> ?from.
?db_to <http://xmlns.com/foaf/0.1/isPrimaryTopicOf> ?to.}
```

Listing 9.1: DBpedia query for birthplace property.

The previous query will retrieve 409,812 pairs of (Place, Person) in DBpedia. Meanwhile, if we navigate from places to people in Wikipedia, we will obtain only 221,788 pairs; we will explain later in this thesis how this value is computed. Only 54 % of places in Wikipedia have a navigational path to those people who were born there. We can observe the same phenomenon when querying DBpedia using different properties. We computed this for twenty properties in DBpedia as summarized in Table 9.1.

[1] www.wikipedia.org

[2] A place could be a Country, Province, City or State.

DBpedia Property	fromType	toType	Number of DBpedia connected pairs	Number of Wikipedia disconnected pairs
$prop_1$: birthPlace	Place	Person	409,812	221,788
$prop_2$: deathPlace	Place	Person	108,148	69,737
$prop_3$: party	PoliticalParty	Person	31,371	15,636
$prop_4$: firstAppearance	Work	Person	1,701	142
$prop_5$: recordLabel	Company	Person	25,350	14,661
$prop_6$: associatedBand	MusicalWork	Person	365	73
$prop_7$: Company	Software	developer	14,788	2,329
$prop_8$: recordedIn	PopulatedPlace	MusicalWork	28,351	27,896
$prop_9$: debutstadium	Building	Athlete	595	393
$prop_{10}$: producer	Artist	MusicalWork	70,272	32,107
$prop_{11}$: training	Building	Artist	171	109
$prop_{12}$: previousWork	Album	MusicalWork	72,498	3,887
$prop_{13}$: recordLabel	Company	MusicalWork	118,028	75,329
$prop_{14}$: starring	Person	Film	164,073	42,584
$prop_{15}$: country	PopulatedPlace	Book	19,224	17,281
$prop_{16}$: city	PopulatedPlace	EducationalInstitution	34,061	8,681
$prop_{17}$: associatedBand	Band	MusicalArtist	24,846	4,100
$prop_{18}$: fromAlbum	Album	Single	18,439	1,268
$prop_{19}$: location	PopulatedPlace	Airport	10,049	2,660
$prop_{20}$: notableWork	Book	Person	1,510	73

Table 9.1: Results of 20 SPARQL queries for 20 properties and different resources types.

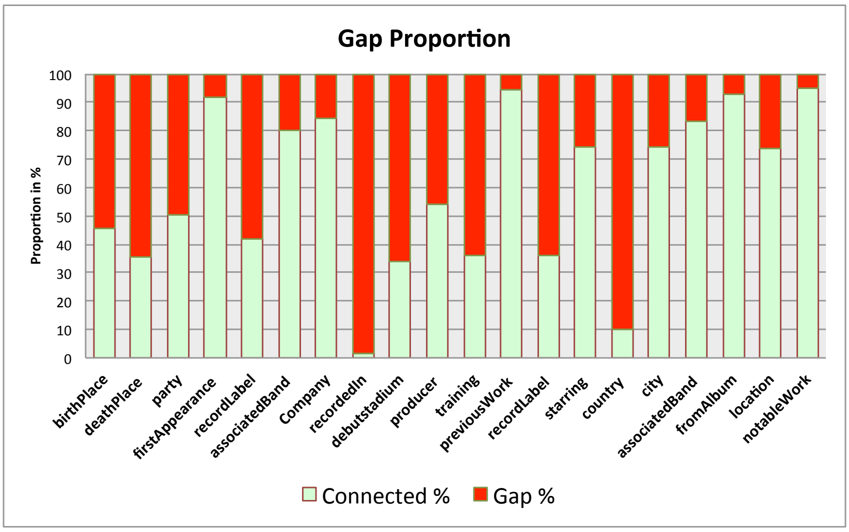


Figure 9.1: Gap proportion for the twenty DBpedia properties of Table 9.1

The last two columns of the table provide, the number of connected pairs obtained by a SPARQL query in DBpedia for the corresponding property and the corresponding number of disconnected pairs in Wikipedia respectively. Two Wikipedia articles are connected if a regular Wikipedia user could navigate from one article to another through a navigational path. According to [46, 48], a navigational path with a length larger than five is unreachable by a regular user; so those articles are unconnected (more details in Section 10.2).

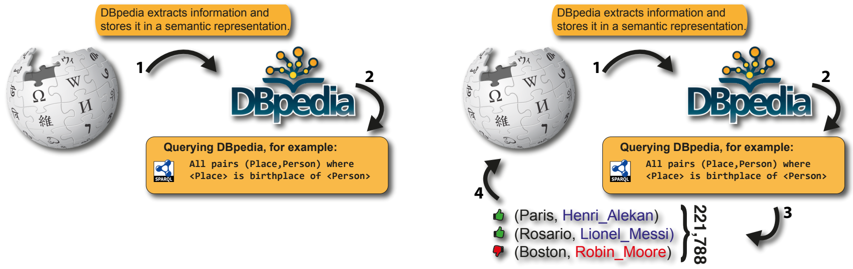
These numbers in the table demonstrate that there is a semantic information gap between DBpedia and Wikipedia. Some connected resources in DBpedia were unconnected in their corresponding Wikipedia articles. This gap can be computed as the difference between the set of connected pairs of resources in DBpedia and their corresponding connected pairs in Wikipedia. Figure 9.1 shows the proportion of the information gap between DBpedia and Wikipedia for twenty studied properties.

The question is : *Is it really necessary to reduce the gap between DBpedia and Wikipedia?* Reducing the gap requires adding missing information to Wikipedia. However, these missing links could be intentionally hidden in Wikipedia to keep the simplicity of the article content i.e. to avoid over linked articles[3]. In other cases, adding these links enable editors to enrich Wikipedia content and to perform better navigation as our social evaluation detailed in Section A.1 demonstrated.

Adding missing links requires learning the Wikipedia conventions [4]. The Wikipedia community has defined conventions that cover a wide diversity of topics: writing style, context of the articles and relations among articles. *Categories*, *List of* pages, and *Navigation templates* are the conventions used to describe one-to-many relationships among Wikipedia articles. The use of categories is the common convention to represent the relationship *"is birth place of"* in Wikipedia. The category `Cat:People_from_<cityName>` seems to be the general rule of this relationship and it is usually subcategory of

[3] http://en.wikipedia.org/wiki/Wikipedia:OVERLINK

[4] http://en.wikipedia.org/wiki/Wikipedia:Conventions


DBpedia extracts information and stores it in a semantic representation.
1
2
DBpedia
Querying DBpedia, for example:
All pairs (Place,Person) where <Place> is birthplace of <Person>
SPARQL
DBpedia extracts information and stores it in a semantic representation.
1
2
DBpedia
Querying DBpedia, for example:
All pairs (Place,Person) where <Place> is birthplace of <Person>
SPARQL
4
3
(Paris, Henri_Alekan)
(Rosario, Lionel_Messi)
(Boston, Robin_Moore)
221,788

(a) Information flow from Wikipedia and DBpedia

(b) Complete cycle of information flow between Wikipedia and DBpedia

Figure 9.2: Information flow between Social Web and Semantic Web

Cat:<cityName>. For example, the *"is birthplace of"* relationship between *Boston*[5] and *Tim_Barsky*[6] is described using the category People_from_Boston. This relation is represented by the navigational path Boston/ Cat:Boston/ Cat:People_from_Boston/Tim_Barsky which must be read: *"from Boston article, the user navigates through a link to the category Boston then he or she navigates to People_from_Boston category, and then to Tim_Barsky article"*. Although categories in Wikipedia are organized in a hierarchy called the category tree, according to Suchanek et al. [85], a *"Wikipedia category hierarchy is barely useful for ontological purposes. For example, Zidane is in the category* ***Football in France****, but Zidane is a football player and not a football"*. Consequently, it is not possible to apply a semantic approach to discover the best representation for a DBpedia property. We have to map a DBpedia property to its syntactic representation in Wikipedia using a navigational path. In Wikipedia, we can distinguish between: a *One-to-one navigational path* and a *one-to-many navigational path*. A one-to-one navigational path denotes navigation only through Wikilink. A one-to-many navigational path denotes navigation through the category tree or stand-alone lists.

These differences in the convention used to express the birthplace of a person triggers the following new question: *How to find the best Wikipedia convention for a semantic property?*

[5] http://en.wikipedia.org/wiki/Boston

[6] http://en.wikipedia.org/wiki/Tim_Barsky on 24th June, 2013

10

BlueFinder

Contents

10.1 **Formal Definitions** 88

10.1.1 DBpedia Model 88

10.1.2 Wikipedia Model 89

10.2 **Wikipedia Pairs Connection** 90

10.3 **Problem Statement** 91

10.4 **BlueFinder Recommender System** 91

10.4.1 Indexing 92

10.4.2 Recommendation 92

10.4.3 BlueFinder Algorithm 93

This chapter introduces the BlueFinder recommender system that recommends the best Wikipedia convention for a DBpedia semantic property. BlueFinder bootstrap the semantic query tools from Semantic Web and the Wikipedia conventions in order to improve the Wikipedia content with new pieces of information. The pieces of information are obtained from the Semantic Web and will trigger the generation of new information in the Social Web.

This chapter begins with the introduction of the formal definition of the problem. Section 10.1 enumerates formal definitions of the DBpedia and Wikipedia Model. These definitions include the basis of RDF semantic and SPARQL queries definitions, which are related to DBpedia in Section 10.1.1. After that, the Wikipedia graph model is defined. As we described before, users navigate through Wikipedia following links. We formalize the navigation by means of path and path queries, all of them described in Section 10.1.2.

The information disparity between DBpedia and Wikipedia is detected by comparing the difference of connected pair of resources in DBpedia versus those that are connected in Wikipedia. The result of that difference will be the *disconnected* pairs of resources. Though, Section 10.2 introduces the description to understand when a pair of DBpedia resources is *connected* or *disconnected* in Wikipedia. Section 10.3 defines the problem of detecting conventions in Wikipedia to describe DBpedia semantic properties in Wikipedia in terms of recommender systems. Finally, in Section 10.4 we introduce the BlueFinder recommender system.

10.1 Formal Definitions

This Section introduces formal definitions of the SPARQL queries, DBpedia model, Wikipedia Model and Path Queries.

10.1.1 DBpedia Model

The knowledge base of DBpedia has a rich set of properties. In addition to specific properties (birthPlace, city, ...), each resource in DBpedia has types definition coming from DBpedia ontology and Yago [85] ontology. For instance, the *rdf:type* describes resources types. This knowledge provides datasets for the BlueFinder recommender through SPARQL queries. Basically, DBpedia knowledge base is an RDF graph without Blank nodes. We recall the following RDF semantics definitions [75]:

Definition The Sets I (IRI Identifiers), B (Blank Nodes), L (Literals) and Υ (Variables) are four infinite and pairwise disjoint sets. We also define $T = I \cup B \cup L$. An *RDF-Triple* is 3-tuple $(s, p, o) \in (I \cup B) \times I \times T$. An *RDF-Graph* is a set of RDF-Triples.

Definition A *triple pattern* is a tuple $t \in (I \cup \Upsilon \cup L) \times (I \cup \Upsilon) \times (I \cup \Upsilon \cup L)$. A *Basic Graph Pattern* is a finite set of triple patterns. Given a triple pattern t, $var(t)$ is the set of variables occuring in t, analogously, given a basic graph pattern B, $var(B) = \cup_{t \in B} var(t)$. Given two basic graph patterns B_1 and B_2, the expression B_1 AND B_2 is a graph pattern.

Definition A *mapping* μ from Υ to T is a partial function $\mu : \Upsilon \to T$. The domain of μ, $dom(\mu)$, is the subset of Υ where μ is defined.

Definition Given a triple pattern t and a mapping μ such that $var(t) \subseteq dom(\mu)$, $\mu(t)$ is the triple obtained by replacing the variables in t according to μ. Given a basic graph pattern B and a mapping μ such that $var(B) \subseteq dom(\mu)$, then $\mu(B) = \cup_{t \in B} \mu(t)$.

Definition Two mappings μ_1, μ_2 are *c*ompatible (we denote $\mu_1 \shortparallel \mu_2$) if and only if for all $?X \in (dom(\mu_1) \cap dom(\mu_2))$, then $\mu_1(?X) = \mu_2(?X)$. This is equivalent to say that $\mu_1 \cup \mu_2$ is also a mapping.

Two important corollaries of this last definition are: *i*) two mappings with disjoint domains are always compatible, *ii*) the empty mapping (the one with empty domain) is compatible with any other mapping.

Definition Let Ω_1, Ω_2 two sets of mappings. The join between Ω_1 and Ω_2 is defined as: $\Omega_1 \bowtie \Omega_2 = \{\mu_1 \cup \mu_2 \,|\, \mu_1 \in \Omega_1 \wedge \mu_2 \in \Omega_2 \wedge \mu_1 \shortparallel \mu_2\}$

Definition Given an RDF-Graph G, the *e*valuation of a triple pattern t over G corresponds to: $[[t]]_G = \{\mu \,|\, dom(\mu) = var(t) \wedge \mu(t) \in G\}$. The *e*valuation of a basic graph pattern B over G is defined as: $[[B]]_G = \bowtie_{t \in B} [[t]]_G$. The *e*valuation of a Graph Pattern B' of the form (B_1 AND B_2) over G is as follows: $[[B']]_G = [[B_1]]_G \bowtie [[B_2]]_G$

Consider the following SPARQL query over a data set D: $Q =$ SELECT x_i, y_i WHERE $x_1 p_1 y_1$ AND ... AND $x_n p_n y_n$

The answer for this query $Q(D)$ is an assignment of distinguished variables (those variables in the SELECT part of the query) i.e. the *e*valuation of a triple pattern t over D. For instance, the SPARQL query Q1 in Listing 9.1 over DBpedia gives a couple of Wikipedia pages of people and their birth place.

In this work, we use SPARQL queries that have a triple pattern of the form: s d:property o where d:property is a specific DBpedia property for s. For instance, in the previous query the property p is $d :< db-p : birthPlace >$, we denote the result of the query by $Q_p(D)$.

10.1.2 Wikipedia Model

In this section we will describe Wikipedia as a graph where nodes are the Wikipedia articles (regular articles and categories) and links are the edges. Wikipedia consists of a set of articles and hyperlinks among them. The Wikipedia graph G can be defined as:

Wikipedia Graph $G = \{W, E\}$ where W is a set of nodes and $E \subseteq W \times W$ is a set of edges. Nodes are Wikipedia articles (wiki pages) and edges are links between articles. Given $w_1, w_2 \in W$, $(w_1, w_2) \in E$ if and only if there is a link from w_1 to w_2.

Path A path $P(w_1, w_n)$ between two Wikipedia articles is a sequence of pages $w_1/\ldots/w_n$, s.t. $\forall i\, w_i \in W \wedge \forall i,j : 1 \leqslant i < j \leqslant n,\, w_i \neq w_j,\, \forall i : 1 \leqslant i \leqslant n-1$ where $(w_i, w_{i+1}) \in E$ is a link between w_i and w_{i+1}. w_1 and w_n are called the *source* page and the *target* page respectively. The length of a path is the number of articles in the sequence, length $P(w_1, w_n) = n$.

Given a $Q_p(D)$, the set of all pairs $(f, t) \in Q_p(D)$ that are connected in Wikipedia by a path with length up to l is defined as:

Wikipedia Connected Pairs $C_p(l) = \{(f,t) \in Q_p(D)$ s.t. $\exists P(f,t)$ and $length(P(f,t)) <= l\}$

A path query is a generalization of similar paths. Usually, regular expressions are used for expressing path queries [4]. Informally, the set of answers of a path query $PQ(w_1, w_2)$ over G is the set of all pairs of nodes in G connected by a directed path such that the concatenation of the labels of the nodes along the path forms a word that belongs to the language denoted by L^*. Many works have been done on path queries in different domains [1, 4, 5]. We adapt the path query definition in [1, 4] to the context of Wikipedia.

Let Σ be an alphabet, a language over Σ is a sequence of elements of Σ called words. Regular expressions can be used to define language over Σ. We use regular expression patterns [4] i.e. patterns that include variables. Let X be a set of variables.

Definition The set of regular expressions $R(\Sigma, X)$ over Σ can inductively defined by: (1) $\forall a \in \Sigma$, $a \in R(\Sigma, X)$; (2) $\forall x \in X, x \in R(\Sigma, X)$; (3) $\epsilon \in R(\Sigma, X)$ (4) If $\forall A \in R(\Sigma, X)$ and $\forall B \in R(\Sigma, X)$ then $A.B$, $A^* \in R(\Sigma, X)$; such that A.B is the concatenation of A and B and A^* denotes the Kleene closure

The language defined by a regular expression pattern is:

Definition Let $R, R' \in R(\Sigma, X)$ two regular expression patterns. $L^*(R)$ is the set of words of $(\Sigma \bigcup X)^*$ defined by: (1) $L^*(\epsilon) = \{\epsilon\}$; (2) $L^*(a) = \{a\}$; (3) $L^*(x) = \Sigma \bigcup X$; (4) $L^*(R.R') = \{w'.w \,|\, w \in L^*(R)$ and $w' \in L^*(R')\}$; (5)L^* $(R^+) = \{w_1 \ldots w_k \,|\, \forall i \in [1...k], w_i \in L^*(R)$;(6) $L^*(R^*) = \{\epsilon\} \bigcup L^*(R+)$.

A path query is a generalization of similar paths by regular expressions patterns. The answer to a path query is defined by:

Path Query A Wikipedia path query (in short path query) $PQ \in R(\Sigma, X)$ is a regular expression pattern. A pair of nodes (x, y) of G covers (or satisfies) a path query $PQ(x, y)$ over Σ and X if there exists a path P from x to y in G and a map μ from $\Sigma \bigcup X$ to $term(G)$ such that $\Lambda(P) \in L^*(\mu(R))$ where $\Lambda(P) = \Lambda(a_1) \ldots \Lambda(a_k)$ over $(\Sigma \bigcup X)^*$ is associated to the path $P = (a_1, ..., a_k)$ of G

In the context of Wikipedia $\Sigma = W$. For the purpose of this work, we limit X to two variables $X = \{\#from, \#to\}$. Given a $Q_p(D)$, $C_p(l)$ is the set of all pairs $(f, t) \in Q_p(D)$ that are connected in Wikipedia by a path with length up to l. The BlueFinder algorithm uses path queries and computes the coverage of path queries for a set of pairs of Wikipedia articles.

Path Index Given a $C_p(l)$, Path Index (PI) is a bipartite graph $(PQ, C_p(l), I)$, it represents the coverage of path queries for a set of pairs of Wikipedia articles that are related by a DBpedia property p. PQ is an ordered set of path (descendent order by element degree), $I = PQ \times C_p(l)$ is the set of edges relating elements from PQ with elements from $C_p(l)$; $(pq, v) \in I \Leftrightarrow pq \in PQ \wedge v \in C_p(l) \wedge v$ covers pq. The first path query in PQ is the general representation of the semantic property p in Wikipedia.

Given a Path Index $PI = (PQ, C_p(l), I)$ and a path query $pq \in PQ$ the rating of the path query in the path index is defined by the degree of the path query in the path index bipartite graph:

rating $rating(pq, PI) = |\{e \in C_p(l) : (pq, e) \in I\}|$

10.2 Wikipedia Pairs Connection

The information disparity between DBpedia and Wikipedia is detected by comparing the difference between the set of connected pairs of resources in DBpedia and those pairs that are connected in Wikipedia. A $Q_p(D)$ is obtained by evaluating a SPARQL query in DBpedia. The pairs of resources included in $Q_p(D)$ are related in DBpedia by the semantic property p. The next step is analyzing which pairs are connected and which are disconnected in Wikipedia.

Two Wikipedia articles (a, b) that are related by a DBpedia semantic property with one-to-many cardinality are connected when at least one of the following sentences is true:

1. There is a navigational path from a to b through the category tree with length less or equal to five[1]. For example: `Rosario,_Santa_Fe / Cat:Rosario,_Santa_Fe / Cat:People_from_Rosario,_Santa_Fe / Lionel_Messi`

2. There is a direct link from a to b i.e. `Rosario,_Santa_Fe / Lionel_Messi`

3. a has a direct link to a `List of` page that has a direct link to b. For example, `Al_Pacino / List_of_awards_and_nominations_received_by_-Al_Pacino / Academy_Award` that connects the pair elements *(Al_Pacino, Academy_Award).*

10.3 Problem Statement

We describe the problem of defining the best representation of missing links in Wikipedia as a collaborative recommender system problem. According to Adomavicioius and Tuzhilin [3], *"collaborative recommender systems try to predict the utility of items for a particular user based on the items previously rated by other users"*. More formally, the utility $u(c, s)$ of item s for user c is estimated based on the utilities $u(c_j, s)$ assigned to item s by those users $c_j \in C$ who are "similar" to user c. In the context of Wikipedia, we do not directly apply recommenders to suggest Wikipedia articles to users but to suggest links between articles. We want to predict the utility of path queries for a particular pair of Wikipedia articles based on those rated by the Wikipedia community. In other words, the pairs of articles *(from,to)* will play the role of users and the path queries will be the items. Then, the utility $u(c, pq)$ of a path query pq for a pair c related by a semantic property p is estimated based on the utilities $u(c_j, pq)$ assigned to pair c by those pairs $c_j \in C_p(l)$, $u : Q_p(D) \times PQ \to R$, where R is a totally ordered set and l is the maximum length of the path queries[2].

Given a property p in DBpedia, $C_p(l)$ and PQ path queries covered by the elements of $C_p(l)$. Then, for a given pair of Wikipedia articles $(from, to)$, we have to recommend the path query that maximizes the utility function.

10.4 BlueFinder Recommender System

BlueFinder is a collaborative filtering recommender system. It uses a memory based algorithm [16] to make rating predictions based on the entire collection of previously rated path queries. As a consequence, the value of the unknown

[1] Two Wikipedia articles are connected in Wikipedia if a regular Wikipedia user could navigate from one article to another. The navigability is determined by how complex the task is to achieve for a user. The complexity to navigate in a hyperlink structure like Wikipedia is related with several variables like depth, breath, width and other cognitive factors [46, 47, 69]. Five links of depth is an average number among the different previous experimentation[48, 46].

[2] We restrict the length to five as we mentioned before.

rating $r_{c,s}$ for a pair c and path query s will be computed as an aggregate rating of other k similar pairs for the same path query s. The recommender returns a set of recommended path queries that can be used to represent the semantic property. The recommendations have to include at least one path query that can represent the semantic relation following the conventions of Wikipedia community. BlueFinder is based on the popular k-Nearest Neighbors(kNN) and Multi label kNN algorithm [102] adapted to the context of DBpedia and Wikipedia. Namely, the BlueFinder algorithm identifies the k connected neighbors of the unconnected pair (from, to), and then it selects the most rated path queries for the k nearest neighbors. Finally, the selected path queries will be the prediction set.

The BlueFinder algorithm is organized in two main steps: indexing and Recommendation.

1. **Indexing:** Given a DBpedia semantic property p, this step computes the item set, the user set and the item ratings. It generates the set of possible path queries for p.

2. **Recommendation:** For a disconnected pair of Wikipedia articles, a set of path queries that best represents the convention to connect the pair of articles is generated by the algorithm. The algorithm estimates the path queries that maximize the utility function for the disconnected pair of articles based on the utility of the path queries with similar connected pairs. It implements the KNN algorithm approach using the generated set in the previous step.

10.4.1 Indexing

In this step, BlueFinder algorithm receives the result set of a DBpedia Semantic query $Q_p(D)$ and computes the item set, user set and item ratings. Note that in the context of this article the items are the path queries, and the users are the pairs of pages retrieved from DBpedia. For each pair of articles $(from, to) \in Q_p(D)$, BlueFinder calculates all the path queries in Wikipedia that connect the *from* article with the *to* article with a maximum number l of links. The user set, item set, and ratings of the items are expressed with a path index $(PQ, C_p(l), I)$: PQ is the item set, $C_p(l)$ is the user set, and the rating of a path query is defined by the *rating* operation. The set of disconnected pairs (the information gap) is defined as: $disconnected_p(D) = Q_p(D) - C_p(l)$. The details of the indexing algorithm are in Section 10.4.3.

10.4.2 Recommendation

After the previous step, BlueFinder performs a *k-NN* algorithm to recommend a set of path queries. Given a disconnected pair of articles in Wikipedia, BlueFinder identifies the k nearest connected pairs to the disconnected one, and then it obtains the path queries that connect the k neighbors. If all neighbors are connected by one particular path query, that path query will be the best convention to recommend. In the other case, the algorithm will select those path queries that maximize the utility of the connected pairs. Indeed, in the first case the level of confidence of the recommendation will be higher than in the second case.

The k-NN algorithm uses a similarity measure function to select the nearest neighbors. We define the *Semantic Pair Similarity (SPS)* function to measure the similarity between pairs of article. SPS, which is based on the well known Jaccard distance [37], measures the degree of overlap in the DBpedia types that describe a pair of Wikipedia articles. The range of the STS is from 0 (identical pairs) to 1 (totally disjoint pairs). The Semantic Pair Similarity function is defined as:

Semantic Pair Similarity (SPS) Given two pairs of pages $c_1 = (a, b)$ and $c_2 = (a', b')$ and the data type set for a, a', b and b' in DBpedia defined as $A = \{t/$ a $rdf : type$ t $\in$ DBpedia$\}$, $A' = \{t/$ a' $rdf : type$ t $\in$ DBpedia$\}$, $B = \{t/$ b $rdf : type$ t $\in$ DBpedia$\}$ and $B' = \{t/$ b' rdf:type t'$\}$ $\in$ DBpedia $\}$. The $SPS(c_1, c_2)$ is defined as:

$SPS(c_1, c_2) = \frac{Jaccard\ distance(A,A')+Jaccard\ distance(B,B')}{2}$

where $Jaccard\ distance(S, S') = \frac{|S \cup S'|-|S \cap S'|}{|S \cup S'|}$, S and S' any set.

Now, we can define the kNN [53] in our context as:

KNN Given a pair $r \in Q_p(D)$ and an integer k, the k nearest neighbors of r denoted $KNN(r, Q_p(D))$ is a set of k pairs from $Q_p(D)$ where $\forall o \in KNN(r, Q_p(D))$ and $\forall s \in Q_p(D) - KNN(r, Q_p(D))$ then $SPS(o, r) \leq SPS(s, r)$.

Having a $PI = (PQ, C_p(l), I)$; the value for an unknown rated $r_{c,s}$ for unconnected pair in Wikipedia c and a path query $s \in C_p(l)$, can be computed as:

$r_{c,s} = rating(s, PI')$

where $PI' = (PQ, C_p(l)', I)$ and $C_p(l)' = KNN(c, C_p(l))$

10.4.3 BlueFinder Algorithm

The *BlueFinder* algorithm returns a set of recommended path queries that can be used to represent the semantic property between two Wikipedia articles. The recommendations have to include at least one path query that can represent the semantic relation following the conventions of Wikipedia community.

The BlueFinder algorithm works as a pipeline process. Firstly, it starts with $Q_p(D)$ semantic query results from DBpedia, and it generates the user set, item set and ratings by indexing the path queries and the connected pairs in a bipartite graph. Secondly, the index is passed to the recommendation step to compute the KNN algorithm and to select a set of path queries. Finally, a noise filter step is done before the recommendation retrieval. Figure 10.1 shows the relations and pipe data flow among the artifacts.

The BlueFinder algorithm (Algorithm 1) receives four inputs: (1) the unconnected pair of Wikipedia articles x, (2) maximum number $maxR$ of recommendations, (3) the number of k neighbors, and (4) the maximum length of paths. To simplify, we use DBpedia (DBp) and Wikipedia (W) as algorithm's global variables. At the beginning, the algorithm calls $PathIndex$ algorithm in line 1 to generate the Path Index bipartite graph path query set and then,

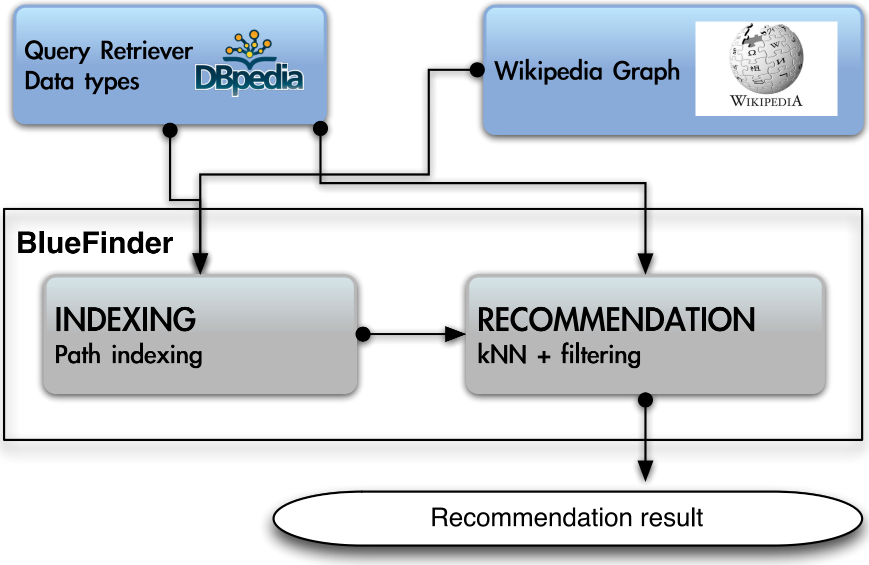


Figure 10.1: BlueFinder data flow

in line 2, it calls the *recommendation* algorithm that obtains the set of recommendations for the unconnected pair.

The *PathIndex* algorithm (algorithm 2) is in charge of computing the indexing part of the algorithm. In this case, for each pair of Wikipedia articles $(from, to)$ included in a given $Q_(D)$, the algorithm performs a Deep First Search up to l starting from the $from$ article and finishing in the *to* article in the Wikipedia graph (from line 1 to 4 in Algorithm 2). For each path reaching the *to* article, it generalizes the path into a path query and builds the path index as a bipartite graph (from line 5 to 8 in Algorithm 2). Finally, it returns the path index that is ready to be used in the next step.

In the end, given the unconnected pair and the number of k neighbors the *recommender* algorithm returns the best *maxR* recommendations to connect the unconnected pair (Algorithm 3). At first, the recommender algorithm computes the k nearest connected pair for the unconnected one using the *kNN* algorithm described in Definition 10.4.2. Then, the algorithm generates a new path index that contains only the path queries, connected pairs and ratings related with the k neighbors. The generated path index contains the path queries that will be recommended and its ratings. Before the recommendations are returned, BlueFinder cleans regular-user-unreachable-paths (e.g. paths that include administrative categories) by means of the noiseFilter (Algorithm 7) and similar path queries are grouped by StarGeneralization algorithm (Algorithm 8). Finally, BlueFinder returns the *maxRecom* best ranked path queries.

The $noiseFilter$ algorithm (Algorithm 7) deletes all the paths queries that are not accessible by a regular user. Wikipedia includes several administrative categories which are used by administrators. In order to recommend path queries that can be utilized by regular users, $noiseFilter$ deletes those categories whose names begin with "Articles_", "All_Wikipedia_", etc, such as `Cat:Articles_to_be_merged`.

BlueFinder filters path queries into star path queries in order to reduce

Algorithm 1 BlueFinder

```
Require: $x$ : unconnected pair, $maxR$ : maximum number of recommendations, $Q_p(D)$, $k$ : number of neighbors, $l$:max path lenght
Ensure: Recommendation path query set
index ← PathIndex(Q_p(D), l)
recommendations ← Recommendation(x, k, maxR, index)
return recommendations
```

Algorithm 2 PathIndex

```
Require: $Q_p(D)$, l: path length
Ensure: PI bipartite graph
$index = (\varnothing, \varnothing, \varnothing)$
for all $(from, to) \in Q_p(D)$ do
    $allPaths \leftarrow \varnothing$ , $curL \leftarrow 0$, $curPath \leftarrow \varnothing$
    $generateAllPath(from, to, l, curL, allPaths, curPath)$
    for all $path \in allPaths$ do
        $pathQuery \leftarrow buildPathQuery(path, from, to)$
        $index \leftarrow insertInIndex(index, pathQuery, (from, to))$
    end for
end for
return $index$
```

Algorithm 3 Recommendation

```
Require: $x = (s, t)$: unconnected pair, $k$: number of neighbors, $maxRecom$: maximum number of recommendation, $PI = (PQ(s), C_p(l), I)$: Path index
Ensure: A recommendation list of star path queries that is ordered in rating descendent order
$k_{neighbors} \leftarrow kNN(x, C_p(l))$
$knnPQ \leftarrow \bigcup_{c_i} pq : (pq, c_i) \in I, c_i \in k_{neighbors}$
$knnI \leftarrow \bigcup_{c_i} (pq, c_i) : (pq, c_i) \in I, c_i \in k_{neighbors}$
$knnPI \leftarrow (knnPQ, k_{neighbors}, knnI)$
$M \leftarrow noiseFilter(knnPI)$ {M ordered in rating descendent order}
$M \leftarrow starGeneralization(M, knnPI)$
return first $maxRecom$ path queries of $M$
```

Algorithm 4 generateAllPath

```
Require: $from, to$ : Wikipedia article, $l, curL$ : integer, $allPaths$ : $setOfPaths$, $curPath$ : $path$
Ensure: All paths that start in $from$ and end in $to$ in Wikipedia Graph with length up to l. The results only include paths through the category tree, the use of List_of_ pages or direct links.
  if $from = to$ then
    $allPaths \leftarrow allPaths \bigcup \{curPath\}$
  else if $l > curL$ then
    {Traverse through Wikipedia graph edges set E}
    for all $neighbor \in \{n : (from, n) \in E\}$ do
      $curPath \leftarrow curPath + neighbor$
      $curL \leftarrow curL + 1$
      $generateAllPath(neighbor, to, l, curL, allPaths, curPath)$
      $curPath \leftarrow curPath - neighbor$
      $curL \leftarrow curL - 1$
    end for
  end if
  return $allPaths$
```

Algorithm 5 buildPathQuery

```
Require: $v_1/.../v_n$ : path $P(v_1, v_n)$, $from, to$ :Wikipedia articles
Ensure: A path query that covers $(v_1, v_n)$
for all $v_i$ in $path$ do
  $v_i \leftarrow stringReplace(v, domain, \#from)$
  $v_i \leftarrow stringReplace(v, range, \#to)$
end for
return $path$
```

Algorithm 6 insertInIndex

```
Require: $index = (PQ, C, I), pathQuery, (domain, range)$
Ensure: Insert in $index$ the path query, the pair and the edge that relates the former elements.
$PQ = PQ \bigcup \{pathQuery\}$
$C = C \bigcup \{(domain, range)\}$
$I = I \bigcup \{(pathQuery, (domain, range)\}$
```

Algorithm 7 noiseFilter

```
Require: $PI = (PQ, C, I)$: Path index
Ensure: Set of regular user navigable path queries.
  $noise = \{"Articles\_", "All\_Wikipedia\_", "Wikipedia\_", "Non-free", "All\_pages\_", "All\_non"\}$
  for all $pq = (p_1, .., p_n) \in PQ;$ do
    if $p_i$ contains any c $\in noise; 1 \leq i \leq n$ then
      $PQ \leftarrow PQ - \{pq\}$
    end if
  end for
  return $PQ$
```

data sparsity.

Definition A star path query PQ*(f,t)) is a group of similar path queries which respects the following construction rules: (1) it starts with `#from` and ends with `#to`. (2) The * element can only be placed between `#from` and `#to` variables. (3) The * can not be the penultimate element in the path query.

Example $PQ^*(f,t)$ =`#from/*/Cat:People- _from_#from/ #to` is a star path query. $PQ*(f,t)$ =`#from/*/#to` is not a star path query.

starGeneralization algorithm 8 groups path queries into a star path query, if possible.

Algorithm 8 starGeneralization

```
Require: $PQ$: set of path queries, $PI$: Path index
Ensure: $PQ^*$: set of star path queries
  $PQ^* \leftarrow \emptyset$
  for all $pq = (p_1, .., p_{n-1}, p_n) \in PQ;$ do
    if $p_{n-1}$ starts with "Cat:" then
      $PQ^* \leftarrow PQ^* \bigcup \{(p_1, *, p_{n-1}, p_n)\}$
    else
      $PQ^* \leftarrow PQ^* \bigcup \{pq\}$
    end if
  end for
  return $PQ^*$
```

11

BlueFinder Evaluation

Contents

11.1 BlueFinder General Experimentation 99

11.1.1 Evaluation metrics 100

11.1.2 Results and discussion 103

11.2 Adapting BlueFinder to French and Spanish Wikipedia 113

11.2.1 Experimentation 114

11.3 Summary of this Part 117

In this chapter different evaluations were run in order to evaluate the BlueFinder algorithm. In a first instance, we run an exhaustive general evaluation over twenty representative DBpedia semantic properties. This evaluation is the most important in this dissertation and it is detailed in Section 11.1. In a second instance, we adapted BlueFinder algorithm to work with data extracted from the French and Spanish Wikipedia in order to analyze differences and similarities in the use of conventions between the different languages of Wikipedia. This evaluation is introduced in Section 11.2.

11.1 BlueFinder General Experimentation

In this section we analyze the behavior of our approach by means of measuring the prediction accuracy of BlueFinder predictions over the 20 properties shown in Table 9.1. The evaluation is conducted to answer the following questions:

1. What is the best combination of k and $maxRecom$ values to get the best accuracy from BlueFinder?
2. Does BlueFinder retrieve path queries that can fix missing relations in Wikipedia?
3. Does BlueFinder have a correlation between the confidence level and the accuracy of the predictions?

4. Does the Wikipedia Community use different conventions to represent a DBpedia semantic property?

In the following of this chapter we describe the evaluation metrics, the method of the evaluation, and then the data sets used in the experimentation are presented. Finally, the results and discussions are introduced.

11.1.1 Evaluation metrics

We measured the accuracy of BlueFinder usage predictions based on the standard metrics of *Precision* (P), *Recall* (R), *F-measure*(F_1) and *hit-rate*.

Precision relates the number of correct path queries that are predicted by BlueFinder to the total of recommended path queries.

$$P = \frac{|\mu_{relevant} \bigcap \mu_{predicted}|}{|\mu_{predicted}|} \tag{11.1}$$

where $\mu_{predicted}$ is the set of predicted path queries and $\mu_{relevant}$ is the set of relevant or expected path queries.

Recall computes the ratio of relevant path queries to the total of recommended path queries.

$$R = \frac{|\mu_{relevant} \bigcap \mu_{predicted}|}{|\mu_{relevant}|} \tag{11.2}$$

F_1 *score* is the combination of precision and recall (11.3)

$$F_1 = 2 \times \frac{P \times R}{P + R} \tag{11.3}$$

We also use the hit-rate recommendation accuracy [21, 70] that measures the number of cases where BlueFinder recommends at least one correct path query (11.4).

$$hit - rate = \begin{cases} 1 & if|\mu_{relevant} \bigcap \mu_{predicted}| > 0 \\ 0 & otherwise \end{cases} \tag{11.4}$$

The previous measures are extended by studying the distribution of path queries that are predicted by BlueFinder. In this work we want to measure the statistical dispersion of each path query (i) according to the proportion $p(i)$ of pair coverage using Gini Index 11.5 as in [28].

$$GI = \frac{1}{n-1} \sum_{j=1}^{n} (2j - n - 1) p(i_j) \tag{11.5}$$

where $i_1, ..., i_n$ is the list of path queries ordered according to increasing p(i).

The Gini Index retrieves 0 when all path queries are chosen often equally and 1 when a particular path query is always chosen.

Finally, we are going to analyze the confidence of the predictions. A high level of confidence in a prediction means that the system is quite sure that the prediction is accurate while a low confidence means that it is not sure whether the prediction is accurate or not. BlueFinder determines confidence in two levels: *featured predictions*, and the position of each prediction in the recommendation set. In order to evaluate the confidence, we will compare the confidence with the hit-ratio of each prediction.

Method

In order to compute the evaluation metrics described in the previous section, an offline evaluation was designed. The central idea of this evaluation is based on disconnecting connected pairs of articles in Wikipedia and then observing if BlueFinder is able to recreate them. This approach is based on assuming that all connected pairs in Wikipedia follow Wikipedian conventions. Figure 11.1 summarizes the idea of the evaluation method.

For the purpose of this evaluation all the path queries that connect a pair of pages that are related in DBpedia by a semantic property p, are considered the correct paths that represent the semantic property p.

The evaluation is done only for the 10% of the connected pairs. They are randomly selected and kept in a set called N. For instance, for $prop_1$ in Table 9.1, 188,324 pairs are connected in Wikipedia (i.e. 409,812 - 221,788), so 18,832 randomly selected of those pairs will be in the set N. After that, for each connected pair (w_1, w_2) in N the evaluation repeats the following steps:

1. All paths currently connecting (w_1, w_2) in Wikipedia are stored in the $\mu_{relevant}$ set, and immediately all them are eliminated from Wikipedia to "disconnect" (w_1, w_2).

2. BlueFinder is executed to predict the paths that could connect (w_1, w_2). The resulting predictions are kept in $\mu_{predicted}$.

3. The $\mu_{predicted}$ set is compared with $\mu_{relevant}$ set in order to compute the metrics detailed in the previous section such as precision, recall and F1.

4. Finally, Wikipedia is restored up to the state before the pair disconnection. This means that the (w_1, w_2) pair is reconnected by means of $\mu_{relevant}$.

BlueFinder behavior is evaluated in each semantic property mentioned in Table 9.1, and then aggregates the values of all the metrics to have a general point of view. For example, the evaluation measures the precision metric for $prop_1$, then for $prop_2$ and then it continues with the rest of metrics and properties. After all the metrics and properties are computed, the mean of all metric values is calculated.

In order to have an analysis of the best combination of the number of neighbors and the number of the BlueFinder recommendations, the BlueFinder execution is configured with many combinations of the parameter k and *maxRecom* for each disconnected pair. The values for k are from 1 to 10, and the values for *maxRecom* are 1, 3, 5 and *unlimited*. The limit of path queries l was fixed in 5 according to the analysis previously presented. Algorithm 9 summarizes the whole evaluation method.

Limitations

In statistical terms, the $\mu_{relevant}$ set is used as the gold standard for each pair (w_1, w_2) in N. However, $\mu_{relevant}$ could contain paths that are not related to the specific semantic property that is under study, or even a potentially correct prediction could be absent in the $\mu_{relevant}$ set. For example, the $\mu_{relevant}$ set for

Algorithm 9 Evaluation schema

```
Require: $Q_p(D)$ : DBpedia SPARQL result-set, $C_p(5)$ : All the connected pairs in $Q_p(D)$.
Ensure: All the values for precision, recall, f1 and hit-Rate for each case with each combination.
  $connected \leftarrow$ Random 10% from $C_p(5)$
  $maxRecom \leftarrow \{1, 3, 5, maxInt\}$
  for all $x \in connected$ do
    $\mu_{relevant} \leftarrow$ path queries that connect x.
    Disconnect x in Wikipedia.
    for all $k$ from 1 to 10 do
      for all $max \in maxRecom$ do
        $\mu_{predicted} \leftarrow BlueFinder(x, max, Q_p(D), k, 5)$
        $precision_{k,max,x} \leftarrow presicion(\mu_{predicted}, \mu_{relevant})$
        $recall_{k,max,x} \leftarrow ...$
        $F1_{k,max,x} \leftarrow ...$
        $hit - rate_{k,max,x} \leftarrow ...$
        ...
      end for
    end for
    Re-Connect x in Wikipedia
  end for
```

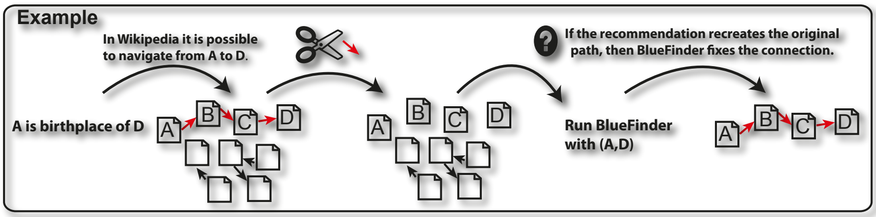

Example
In Wikipedia it is possible to navigate from A to D.
A is birthplace of D
A
B
C
D
A
B
C
D
If the recommendation recreates the original path, then BlueFinder fixes the connection.
Run BlueFinder with (A,D)
A
B
C
D

Figure 11.1: Evaluation method

the pair (London , Richard_Blanshard) with the property `deathPlace` was `#from / * / Cat:People_from_#from / #to` and the first two predictions in the prediction set were `#from / * / Cat:People_from_#from / #to` and `#from / * / Cat:Death_in_#from / #to`. The second prediction could be correct but, as it is not included in $\mu_{relevant}$, the evaluation rejects it as a correct one. Taking into account these considerations, the $\mu_{relevant}$ set is an estimation of the actual path queries and in consequence the BlueFinder is evaluated in the context of the worst case.

Datasets

BlueFinder was evaluated with the twenty semantic properties detailed in Table 9.1. For each property denoted by $prop_i$, a SPARQL query was evaluated on the DBpedia SPARQL end-point. The SPARQL query for each semantic property follows the template showed in Listing 11.1 and the values of *DBpediaSemanticProperty*, *fromType* and *toType* are replaced in each property scenario for the specific values of the first, second and third column respectively that are detailed in Table 9.1. For instance, the SPARQL query in Listing 9.1 corresponds to $prop_1$. The number of the Wikipedia connected pairs of each semantic property is the difference between the number of the DBpedia connected pairs less the number of the Wikipedia disconnected pairs (columns fourth and fifth of Table 9.1).

```
PREFIX db−owl:<http://dbpedia.org/ontology/>
PREFIX db−p:<http://dbpedia.org/property/>

SELECT ?from, ?to WHERE{
?db_from a <fromType>.
?db_to a <toType>.
?db_to db−p:<DBpediaSemanticProperty> ?db_from.

?db_from <http://xmlns.com/foaf/0.1/isPrimaryTopicOf> ?from.
?db_to <http://xmlns.com/foaf/0.1/isPrimaryTopicOf> ?to.}
```

Listing 11.1: "SPARQL query template for evaluation scenarios.

Evaluation results are described and discussed in the next section. The complete values of all the metrics values with the different values for k and *maxRecom* of this evaluation are in https://sites.google.com/site/bfrecommender/publications/.

11.1.2 Results and discussion

First of all in this section, the information gap analysis is presented; then it is discussed the results that were obtained for each metric. Finally, a discussion about the evaluation result is developed.

Gap analysis

According to the last column in the in the Table 9.1, the gap of missing information in Wikipedia is remarkable. In order to analyze the ratio of the gap, shown in Figure 9.1, we noticed that 9 of 20 properties have more than 50 % of

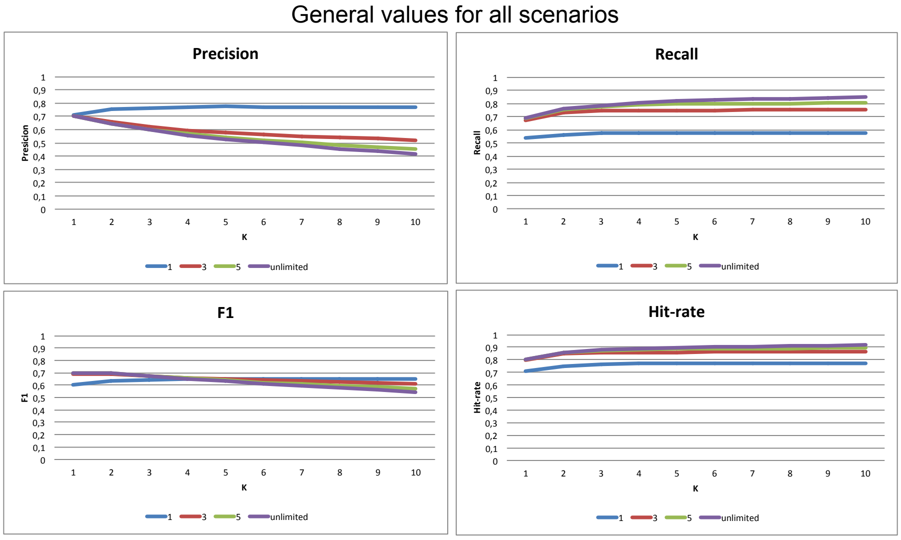


Figure 11.2: Presicion, Recall, F1, and Hit-rate mean of all properties

missing information. In addition to this evidence, the amount of disconnected pairs of the properties $prop_1$ (`birthplace`), $prop_2$ (`deathplace`), and $prop_{13}$ (`recordlabel`) shown in the Table 9.1 is equivalent to more than the 50 % of all disconnected pairs of other properties.

The smallest gap was evidenced in $prop_{20}$ (`notableWork`) and $prop_{12}$ (`previousWork`) only with 5 %. In both cases, this is because the links represent basic information of the connected articles and they are expressed as direct links (`#from / #to`) between the articles.

Accuracy

To assess the best behavior of BlueFinder, we analyze the values of accuracy metrics for the 20 properties from a general perspective. Figure 11.2 shows four line-charts with the mean values of $precision$, $recall$, $F1$ and $hit-rate$ obtained on each property. Each chart describes the relation between $maxRecom$ and k values for each metric.

BlueFinder is able to find, on average, between 75 % and 82 % of the relevant paths, and according to the hit-rate values it is able to fix around 88 % of the cases for k greater than 4 and $maxRecom$ = 3, 5 or *unlimited*. However, the limitations is that the precision values decrease according to the variation of the k values and the number of recommendations.

To detect the best correlation between precision and recall we use the F1 metric. According to the Figure 11.2, all the $maxRecom$ curves converge at k=5 with value 0.65. Therefore, $maxRecom = 5$ and $k = 5$ determine the best accuracy for BlueFinder. The number of correct path queries tips the scales in favor of recall and hit-rate rather than precision. This assumption is based on the fact that the recommendations are presented to the users in descending confidence order, and consequently, the users have extra information to determine the accuracy of the recommendation. Finally, the unlimited $maxRecom$ was dismissed because it had similar recall than $maxRecom = 5$ but lower precision.

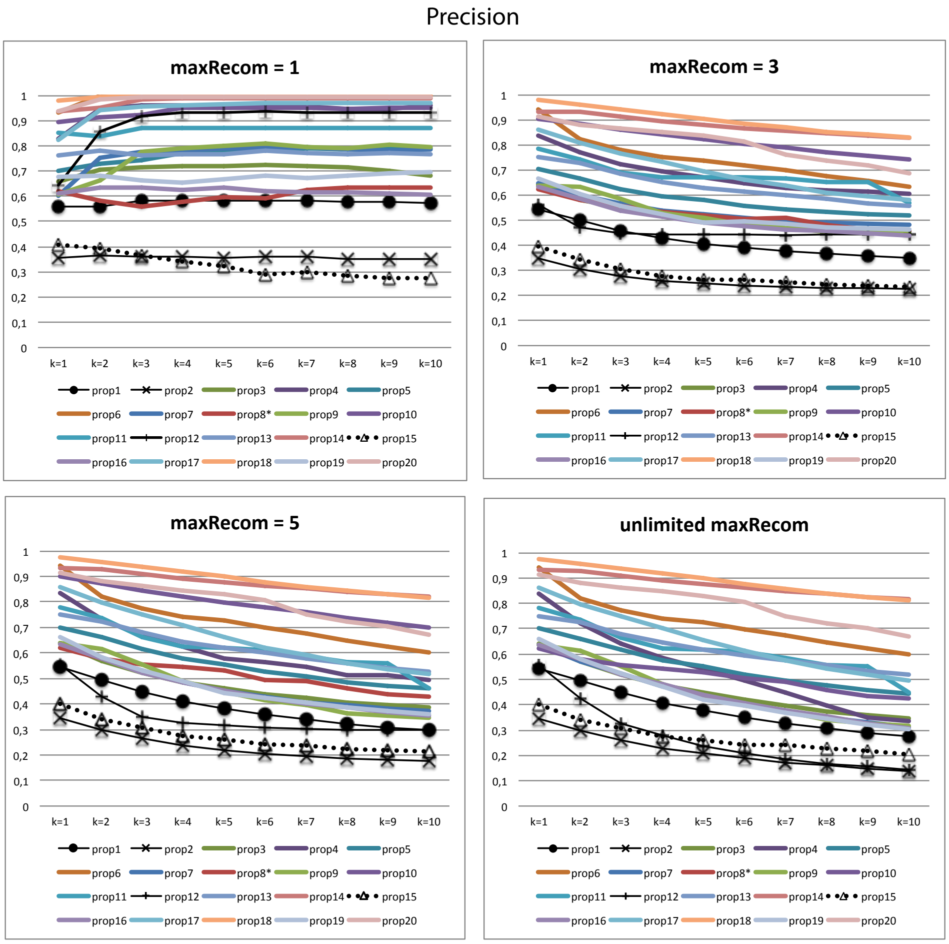


Figure 11.3: Precision all properties

Precision

First, as it can be seen from the Figure 11.3 most of the precision curves decrease, due to BlueFinder introduced non-expected path queries in $\mu_{predicted}$ set. This is because of the size of *maxRecom* has increased but also because the distant neighbors insert noisy paths. Nevertheless, the 70 % of the properties had precision bigger than 0.5 at k=5 and *maxRecom*=5. This evidences that in general terms the precision of BlueFinder was considerably good taking into account that, as we have mentioned, the predictions are presented in confidence order bringing to the users better information (more details in Section 11.1.2).

Second, four of the twenty curves had the lowest precision values. On the one hand, three of them, $prop_1 : birthplace$(● line in charts), $prop_2 : deathplace$ (line with ×), and $prop_{15} : country$ (line with △) had both low precision (less than 0.44 for *maxRecom*=3, 5 and unlimited). This is because of they have a big number of disconnected pairs and shows up that BlueFinder, as many recommender systems, is sensitive to the sparsity of data. On the other hand, the $prop_{12} : previousWork$ (line with +) had a high precision: 0.93 with $maxRecom = 1$ and $k = 5$, but it sharply decreased when the *maxRecom* increased (0.44 with $maxRecom = 3$, 0.31 with $maxRecom = 5$ and 0.23 with $maxRecom = unlimited$). This is because although BlueFinder was able to predict the correct path query for this property, the other path queries that are included in the prediction set are not correct.

Recall and F1

According to Figure 11.4, seventeen of twenty properties had better recall than 0.7, and eleven of twenty had more than 0.8; all of them with k=5 and *maxRecom*=5. Naturally, the three properties out of the norm were $prop_{12}$, $prop_2$, and $prop_{15}$ that we have discussed before. Finally, the lowest recall value with k=5 and *maxRecom*=5 was 0.473 at $prop_2$ property, and the maximum value was 0.972 at $prop_{14}$ property.

F1 metric values are introduced in Figure 11.5. Although the values for the properties $prop_2$, $prop_{12}$ and $prop_{15}$ are at the bottom of the graph as in its general tendency, it is important to remark that most of the properties (12) had F1 values over 0.6 at k=5 and *maxRecom*=5.

Hit-rate

As we have introduced, the hit-rate property indicates if BlueFinder gives at least one path query that can fix the disconnection. The higher is the curve for a property, the bigger is the ratio of fixed cases. As we can observe in Figure 11.6, 80 % of the properties (16 out of 20) had a hit-rate value bigger than 0.84, and only two properties had values lower than 0.6. The properties with the lowest hit-rate were property $prop_{15}$ and $prop_2$; both confirmed the same tendency that appeared in the previous accuracy metric. Although the hit-rate values are low, according to its high level of information gap, the property $prop_8$ has over 75 % of hit-rate on average. These values are promising since most of the 98 % of these property pairs were disconnected in Wikipedia.

The accuracy values demonstrated that BlueFinder retrieves good recommendations. The hit-rate curves confirmed the best combination of k and

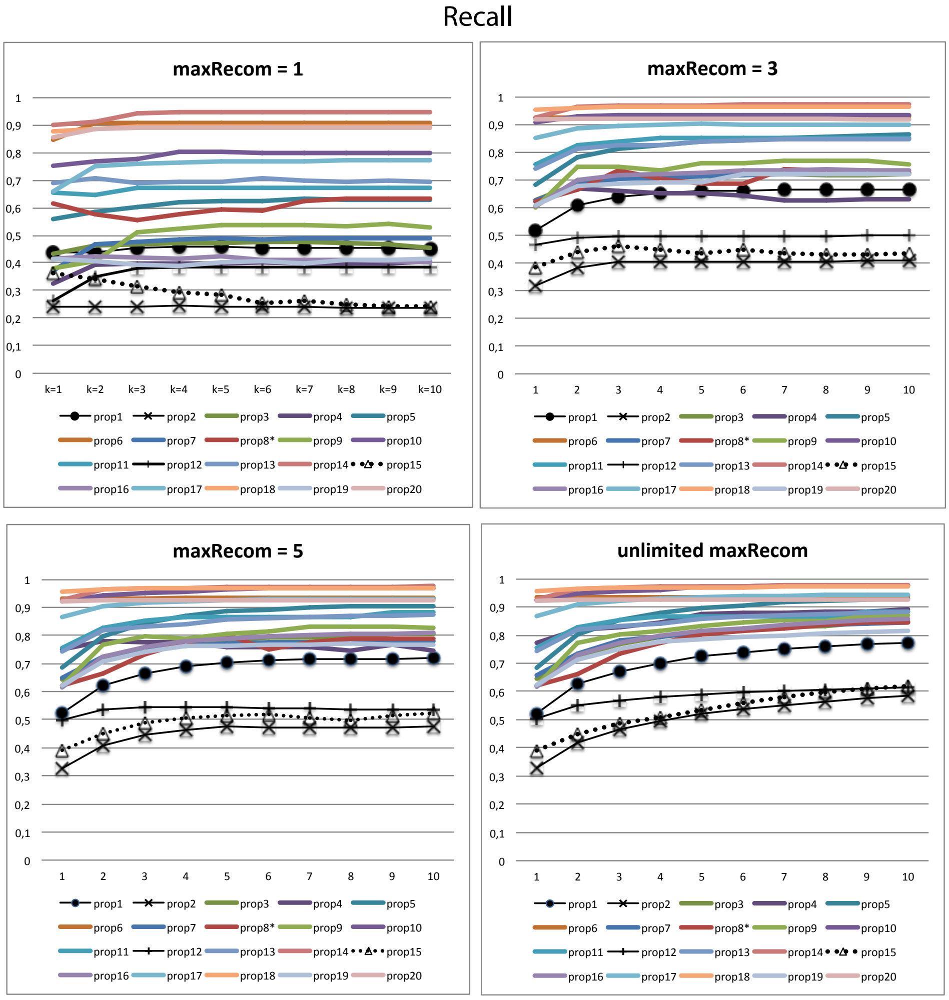


Figure 11.4: Recall for all properties

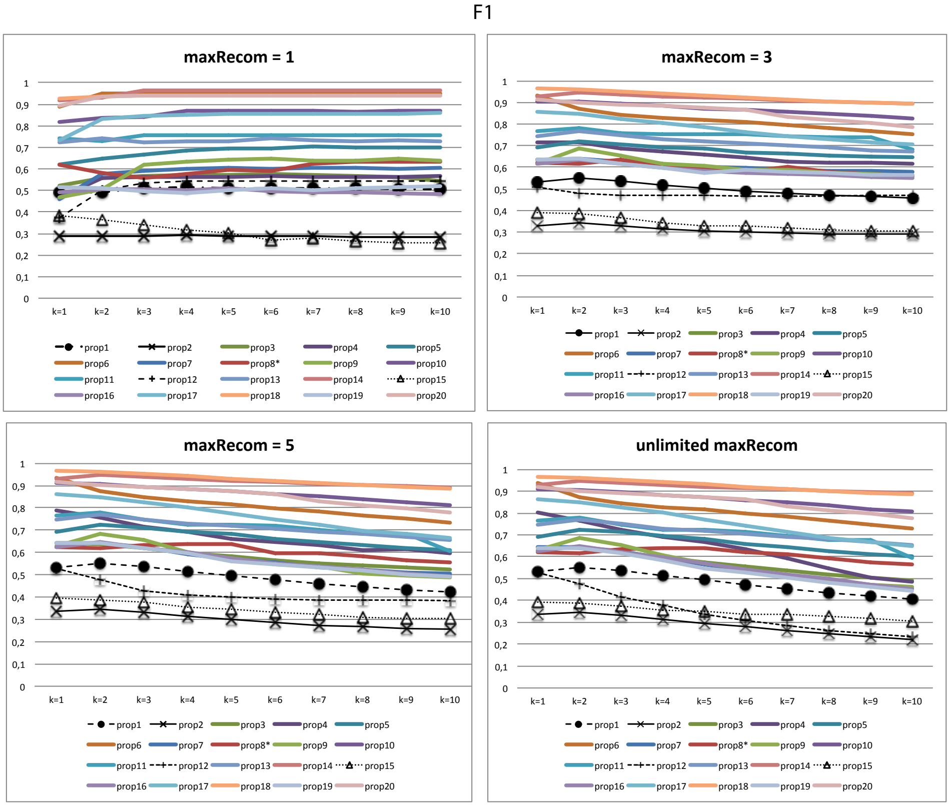


Figure 11.5: F1 for all properties

maxRecom values by setting $k = 5$ and $maxRecom = 5$.

Confidence

The BlueFinder predictions are sorted in confidence descendent order; in consequence the first predictions may have a better hit-ratio than the following predictions. In order to answer the third question of the evaluation *Does BlueFinder have a correlation between the confidence level and the accuracy of the predictions?*, the hit-rate ratio of BlueFinder featured predictions for the twenty semantic properties is shown in Table 11.1. As we can see, featured predictions made by BlueFinder are chiefly prominent: the lowest ratio was 0.84 for property $prop_1$ and the highest ratio was 1 for properties $prop_6$ and $prop_{15}$. The mean of all the hit-rate values was nearly 0.95. This means that BlueFinder is able to fix nearly all the cases where it recommends a featured prediction.

Additionally, Figure 11.7 extends the information from Table 11.1 to all the recommendation positions in a line-chart which compares the hit-rate ratio to the first five positions of the BlueFinder recommendation. As we expected, the curve is in descending order while the first position has the best hit-rate ratio (0.78) and the last position the lowest (0.17). This confirms the correlation between the confidence and hit-rate ratio of the predictions. Unfortunately, the hit-rate ratio curve descends more rapidly than we expected to second position and continues descending until the last position.

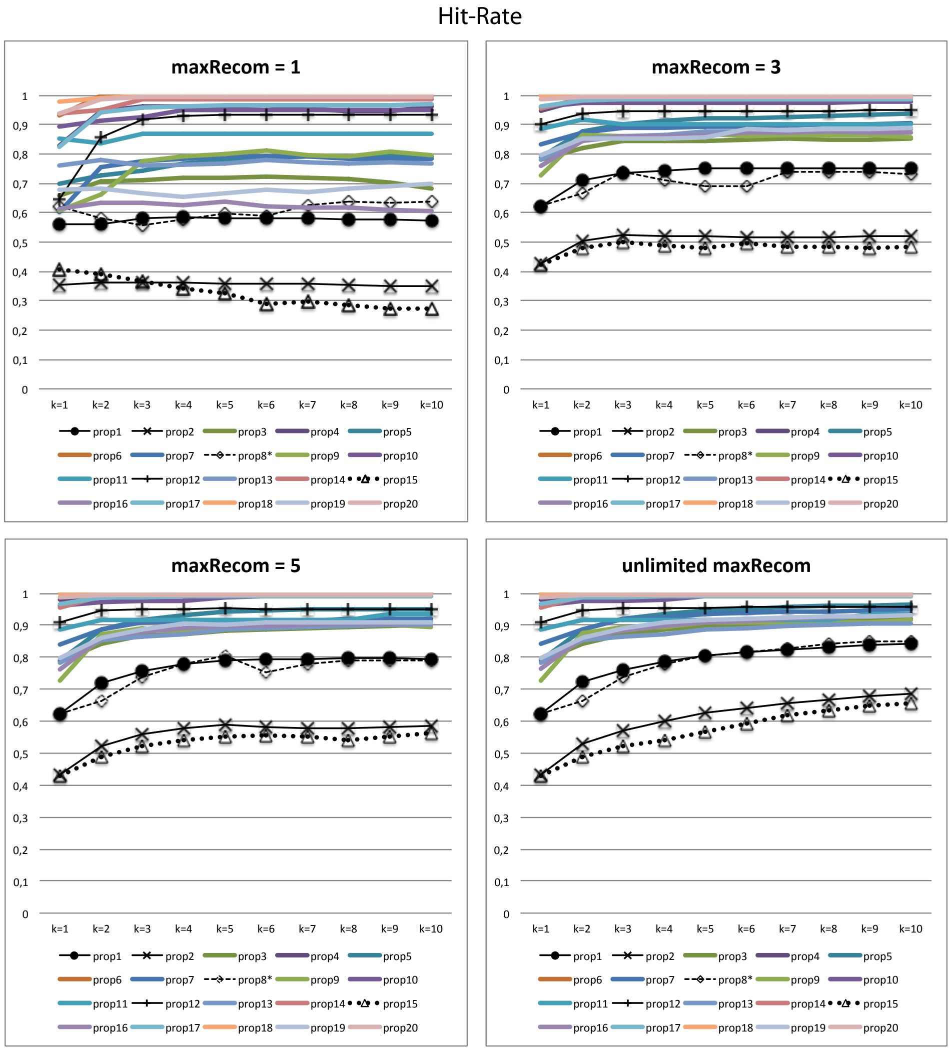


Figure 11.6: Hit-rate for all properties

DBpedia Property	**Hit-rate ratio**
$prop_1$	0.849264
$prop_2$	0.853425
$prop_3$	0.914697
$prop_4$	0.992218
$prop_5$	0.954365
$prop_6$	1
$prop_7$	0.918067
$prop_8$	0.968254
$prop_9$	0.97541
$prop_{10}$	0.977757
$prop_{11}$	0.938776
$prop_{12}$	0.93861
$prop_{13}$	0.950509
$prop_{14}$	0.994295
$prop_{15}$	1
$prop_{16}$	0.868421
$prop_{17}$	0.979371
$prop_{18}$	0.997813
$prop_{19}$	0.938967
$prop_{20}$	0.993056
Mean	**0.95016375**

Table 11.1: Confidence and hit-rate ratio for High Confidence predictions

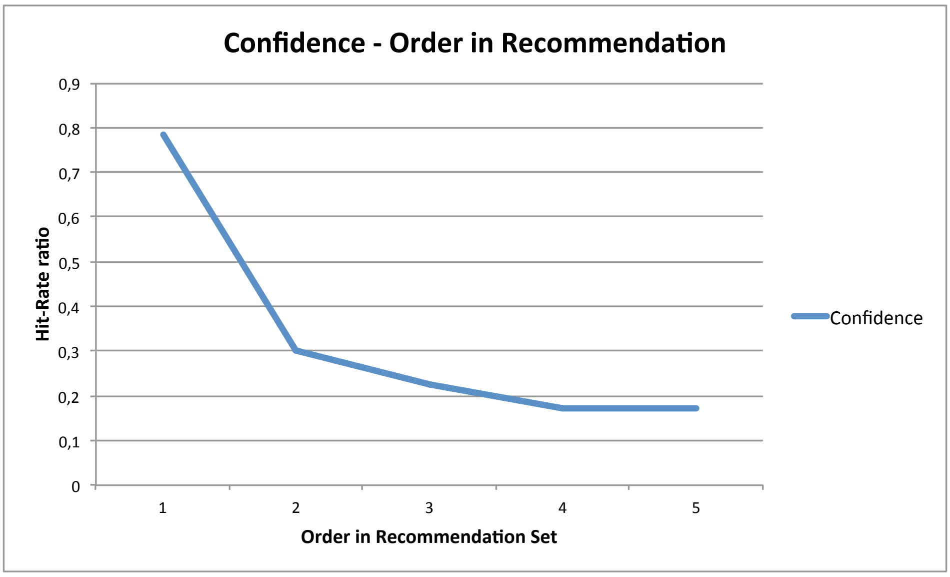


Figure 11.7: Confidence and hit-rate ratio according to prediction order in the recommendation set

Distribution of the BlueFinder Recommendations

In general, the BlueFinder predictions are concentrated in only one path query per semantic property. In 90 % of the properties one convention is mainly utilized by Wikipedia editors to represent the semantic property. This is obtained from the Gini Index values detailed in Table 11.2 where 18 of 20 properties predictions have a Gini Index over 0.8. To benefit the diversity in the predictions, Table 11.2 shows the Gini Index values in the context of BlueFinder with $maxRecom = unlimited$.

However, the predictions for $prop_9$ and $prop_{11}$ properties are better distributed than the rest of the cases. They have Gini Index values between 0.474 and 0.778 in the different values of k. In addition, with a narrowed number of recommendations, we repeated the Gini index study for these two cases with $maxRecom = 5$. The results, shown in Table 11.3, reveal that the distribution of the recommendation remains non-concentrated in both cases but with a small increase in the Gini index.

General Evaluation Conclusions

Evaluation showed that the information gap between DBpedia and Wikipedia is a real and important problem.

According to the evaluations, the best accuracy of BlueFinder was obtained with $k = 5$ and $maxRecom = 5$, and this answers the first question of the evaluation. With these values, the BlueFinder predictions maximize the expected results with a balanced F1 value.

Additionally, 89 % (on average) of the disconnected pairs could be fixed by BlueFinder according to the hit-rate values and almost all the featured predictions fix the disconnection. A Wikipedia editor could use BlueFinder and fix unconnected pairs in Wikipedia.

In order to answer the third question, BlueFinder gives the user the recommendations in descending confidence order. The confidence is also correlated with the hit-rate ratio of the prediction. This allows users to make a better choice of the predictions. The hit-rate of the predictions in the first position is accurate and it is also better when the prediction is a *featured prediction*. BlueFinder gives good prediction even when the contributors have different conventions.

Wikipedia editors, in general, only use one convention to represent a semantic property in Wikipedia but in some cases (two in this evaluation) some communities define particular conventions. We can conclude this by taking into account that the prediction distribution is centralized in one convention. Additionally, the evaluations showed that those predictions are accurate.

BlueFinder gives good predictions even when the contributors have different conventions. Indeed, this can be concluded by taking into account that the Gini Index, in most of the cases, defined a centralized distribution of a path query in the recommendations, and also the accuracy level of BlueFinder is good in those cases.

DBpedia Property	**K=1**	**K=2**	**K=3**	**K=4**	**K=5**	**K=6**	**K=7**	**K=8**	**K=9**	**K=10**
$prop_1$: birthPlace	0.946	0.921	0.902	0.887	0.875	0.866	0.857	0.850	0.843	0.836
$prop_2$: deathPlace	0.922	0.890	0.866	0.849	0.835	0.823	0.814	0.806	0.798	0.792
$prop_3$: party	0.954	0.938	0.928	0.921	0.912	0.906	0.902	0.898	0.894	0.891
$prop_4$: firstAppearance	0.909	0.899	0.885	0.873	0.862	0.854	0.849	0.838	0.835	0.831
$prop_5$: recordLabel	0.956	0.938	0.925	0.912	0.902	0.895	0.889	0.881	0.876	0.871
$prop_6$: associatedBand	0.942	0.929	0.923	0.915	0.907	0.896	0.885	0.872	0.865	0.856
$prop_7$:Company	0.941	0.917	0.901	0.886	0.876	0.872	0.866	0.860	0.855	0.850
$prop_8$:recordedIn	0.861	0.833	0.830	0.813	0.802	0.790	0.769	0.757	0.747	0.736
$prop_9$:**debutstadium**	**0.676**	**0.648**	**0.613**	**0.594**	**0.595**	**0.581**	**0.576**	**0.561**	**0.545**	**0.524**
$prop_{10}$:producer	0.959	0.944	0.933	0.927	0.921	0.915	0.911	0.908	0.904	0.900
$prop_{11}$:training	**0.778**	**0.729**	**0.641**	**0.629**	**0.626**	**0.594**	**0.585**	**0.578**	**0.570**	**0.474**
$prop_{12}$:previousWork	0.955	0.941	0.938	0.931	0.931	0.931	0.927	0.924	0.921	0.919
$prop_{13}$:recordLabel	0.959	0.948	0.938	0.930	0.925	0.920	0.917	0.912	0.906	0.905
$prop_{14}$:starring	0.988	0.977	0.964	0.953	0.942	0.933	0.926	0.918	0.910	0.904
$prop_{15}$:country	0.943	0.929	0.920	0.908	0.900	0.891	0.886	0.882	0.875	0.873
$prop_{16}$:city	0.960	0.942	0.927	0.914	0.904	0.896	0.889	0.882	0.878	0.874
$prop_{17}$:associatedBand	0.963	0.939	0.926	0.915	0.911	0.904	0.899	0.893	0.889	0.885
$prop_{18}$:fromAlbum	0.967	0.953	0.942	0.932	0.923	0.913	0.901	0.885	0.872	0.869
$prop_{19}$:location	0.967	0.945	0.927	0.911	0.898	0.888	0.881	0.872	0.866	0.861
$prop_{20}$:notableWork	0.976	0.962	0.954	0.950	0.948	0.940	0.933	0.934	0.934	0.931

Table 11.2: Gini index of the properties

Property	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8	K=9	K=10
$prop_9$	0.678	0.655	0.630	0.627	0.643	0.649	0.661	0.652	0.662	0.680
$prop_{11}$	0.778	0.729	0.650	0.636	0.648	0.636	0.623	0.616	0.607	0.509

Table 11.3: Gini index of the properties $prop_9$: `debutstadium` and $prop_{11}$: `training` with $maxRecom$=5

11.2 Adapting BlueFinder to French and Spanish Wikipedia

In this section, we adapted and applied BlueFinder algorithm to French Wikipedia (*W@fr*) and Spanish Wikipedia (*W@sp*) in order to analyze if the different language versions of Wikipedia share the same conventions to represent a semantic property from the English DBpedia.

To apply BlueFinder to a given English DBpedia property, we have to provide the set of pairs of pages $Q_p(D)$ that are related by the property p to generate the Path Index. For example, to obtain the pairs of pages for the semantic property *birthplace* that relates Cities and People we execute the SPARQL query on DBpedia shown in Listing 11.2.

If we use the pairs retrieved by the query of Listing 11.2 with BlueFinder in the non-English Wikipedia, BlueFinder will generate neither $C_p(l)$ set nor the Path Index. Most of the pages retrieved by the query in Listing 11.2 do not exist in the non-English Wikipedia because the pages retrieved by this query are only in English. For example, the "Edinburgh" page in the English Wikipedia (*W@en*) does not exist neither in the *W@sp* nor in the *W@fr*: the Spanish name is "Edinburgo" and the French is "Édimbourg". In order to compute the path index, the name of the retrieved pages must be translated.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>.
PREFIX dbpedia−owl:<<http://dbpedia.org/ontology/>.
SELECT DISTINCT ?wpCity ?wpPerson WHERE{
  ?enCity a dbpedia−owl:City.
  ?enPerson a foaf:Person.
  ?enPerson dbpedia−owl:birthPlace ?enCity.
  ?enCity foaf:isPrimaryTopicOf ?wpCity.
  ?enPerson foaf:isPrimaryTopicOf ?wpPerson.
}
```

Listing 11.2: "SPARQL query to obtain the English Wikipedia pages that are related by the birthplace property.

As a first adaptation of BlueFinder, we made page name translation by using the `owl:sameAs`[1] property. In DBpedia, the different language versions of a resource are associated by the `owl:sameAs` property. For example, Listing 11.3 shows the SPARQL query to obtain Spanish resources that are related with *birthplace* property. The `filter` in the SPARQL query helps to select resources in one specific language. With this adaptation we can apply BlueFinder without problems.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>.
PREFIX dbpedia−owl:<<http://dbpedia.org/ontology/>.
PREFIX owl:http://www.w3.org/2002/07/owl#.
```

[1] http://www.w3.org/TR/owlref/#sameAs-def

```
SELECT DISTINCT ?from ?to WHERE{
  ?enCity a dbpedia−owl:City.
  ?enPerson a foaf:Person.
  ?enPerson dbpedia−owl:birthPlace ?enCity.
  ?enCity owl:sameAs ?from.
  ?enPerson owl:sameAs ?to.
  FILTER (regex(?from, "http://es.dbpedia.org") &&
        regex(?to, "http://es.dbpedia.org")). }
```

Listing 11.3: "SPARQL query to obtain the Spanish DBpedia resources that are related by the birthplace property.

The second adaptation of BlueFinder is to apply the *SPS* similarity function in the English DBpedia instead of using French or Spanish DBpedia. Because we observed that English DBpedia has a richer description of resources than non-English Wikipedias. Non-English Wikipedias can derive a wrong selection of similar items producing incorrect prediction. For example, if we compare the types that describe the famous singer *John Lennon* and famous actor *Al Pacino* pages in English DBpedia (*DB@en*) and Spanish DBpedia (*DB@sp*), we can notice that the detailed descriptions for both pages in *DB@en* have 42 and 22 classes respectively and 8 and 7 classes in the *DB@sp*. This difference in the use of types among the different language versions of DBpedia can generate contradictions in the distance measurement. In the English version *John Lennon* and *Al Pacino* are measured as distant (0.857). Meanwhile in the Spanish version, they are measured as close (0.334). We obtain the English DBpedia types of a non-English resource using the `owl:sameAs` property. Listing 11.4 shows an example for the Spanish version of Al Pacino.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>.
PREFIX dbpedia−owl:<<http://dbpedia.org/ontology/>.
PREFIX owl:http://www.w3.org/2002/07/owl#.
SELECT DISTINCT ?type WHERE{
        ?enFrom <owl:sameAs> <es.dbpedia.org/resource/Al_Pacino>.
        ?enFrom a ?type.}
```

Listing 11.4: "SPARQL query to obtain English type description for Spanish Al Pacino resource.

The last adaptation is the translation of path queries to English by means of cross language links. The translation is from the non-English Wikipedia pages to the English version. This enables us to use the same normalization strategy as in the English Wikipedia and also to compare the conventions among non-English Wikipedia versions.

11.2.1 Experimentation

We applied the adapted version of the BlueFinder algorithm to Spanish and French Wikipedia and English DBpedia, all of them are retrieved on January 2013. We want to discover French and Spanish Wikipedia conventions for English DBpedia property *birthplace*. We evaluate precision and recall for both languages.

Data set	$\|PQ\|$	$\|C_p(l)\|$	$\|I\|$	$\|Q_{birthplace}(D)\|$
$Q_{birthplace@sp}(D)$	1,502	7,486	14,998	22,281
$Q_{birthplace@fr}(D)$	3,721	30,407	114,156	36,952

Table 11.4: Path Index bipartite graphs used in the experimentation. Columns show the number of path queries, the number of connected pairs, the number of edges and the number of elements in the Data set

K	1	2	3	4	5	6	7	8	9	10
$precision$	0.426	0.400	0.347	0.313	0.283	0.264	0.245	0.232	0.224	0.214
$recall$	0.421	0.547	0.602	0.634	0.652	0.664	0.669	0.674	0.674	0.670
$hit - rate$	0.485	0.614	0.667	0.697	0.714	0.727	0.732	0.735	0.734	0.731
$F1$	0.424	0.462	0.44	0.419	0.395	0.377	0.358	0.345	0.336	0.324

Table 11.5: Detail of BlueFinder evaluation in the Spanish Wikipedia.

Dataset

For this experimentation, we used the semantic property *birthplace* that relates Cities with People. We run $Q_{birthplace@sp}$ and $Q_{birthplace@fr}$ given in Listing 11.3 and 11.5 respectively. We used a Path Index for $Q_{birthplace@sp}(D)$ and another for $Q_{birthplace@fr}(D)$. Table 11.4 details these indexes and then we run BlueFinder algorithm on the same datasets.

The metrics and the method of this evaluation are the same as in the previous evaluation. In this case, we exercised BlueFinder with 5000 pairs of pairs from the data sets.

```
  PREFIX foaf: <http://xmlns.com/foaf/0.1/>.
  PREFIX dbpedia−owl:<<http://dbpedia.org/ontology/>.
  PREFIX owl:http://www.w3.org/2002/07/owl#.
  SELECT DISTINCT ?from ?to WHERE{
    ?enCity a dbpedia−owl:City.
    ?enPerson a foaf:Person.
    ?enPerson dbpedia−owl:birthPlace ?enCity.
    ?enCity owl:sameAs ?from.
    ?enPerson owl:sameAs ?to.
FILTER (regex(?from,"http://fr.dbpedia.org") &&
regex(?to,"http://fr.dbpedia.org")). }
```

Listing 11.5: SPARQL query to obtain the French Wikipedia resources that are related by the birthplace property.

Results

The Path Index for the Spanish Wikipedia had 7,486 connected pairs out of 22,281 retrieved by the SPARQL query (Table 11.4). This means that the set of pairs to learn is smaller than the set of pairs to fix. This can generate a low rate of fixed values.

Surprisingly, the execution over the Spanish Wikipedia demonstrated that BlueFinder can fix 70 % of the cases. BlueFinder performs well for Spanish Wikipedia. This is evidenced with the curves of *precision*, *recall* and $hit-rate$ shown in Figure 11.8. For $K = 5$ the $hit - rate$ and *recall* rate begin to be stabilized and the *precision* begins to decrease.

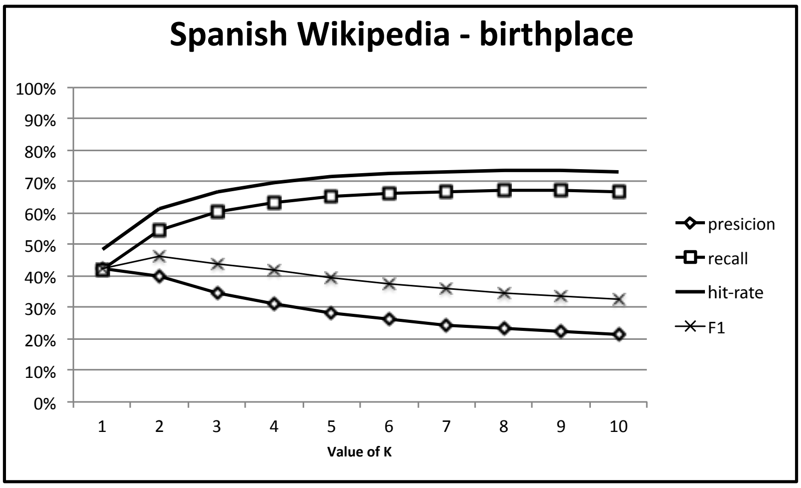


Figure 11.8: BlueFinder evaluation in the Spanish Wikipedia.

K	1	2	3	4	5	6	7	8	9	10
precision	0.559	0.499	0.460	0.438	0.424	0.418	0.413	0.412	0.408	0.407
recall	0.552	0.642	0.670	0.685	0.693	0.706	0.710	0.719	0.721	0.724
$hit - rate$	0.720	0.807	0.837	0.862	0.870	0.887	0.891	0.902	0.906	0.910
$F1$	0.555	0.561	0.545	0.534	0.526	0.524	0.521	0.523	0.520	0.521

Table 11.6: Details of BlueFinder Evaluation in the French Wikipedia.

In Table 11.5, the best $hit - rate$ value was 73,5% with K=10, the best *recall* was 0.674 for K=8 and 9 and the best *precision* was 0.426 with K=1. The worst value of $hit-rate$ was 48,5 % with K=1, the worst value of *precision* was 0.214 with K=10 and the worst value of *recall* was 0.421.

The SPARQL query for *birthplace* property in French language retrieves a set of 36,952 pairs and 30,407 are connected in Wikipedia by a navigation path (Table 11.4). That is equivalent to more than 80% of connected pairs. This means that the French Wikipedia is well developed for the birthplace property. Additionally, this is a good context to BlueFinder because it has a big number of cases to learn and find the proper neighbors.

The execution of BlueFinder over French Wikipedia confirms the effectiveness of the algorithm. In this case, BlueFinder can fix 90% of the cases with a balanced value of *precision* and *recall*. The curve of *precision*, *recall* and $hit - rate$ in Figure 11.9 showed that for K=8 the $hit - rate$, *recall* and *precision* begin to be stabilized with the 90% of fixed ratio. In Table 11.6 the best value for $hit-rate$ was 91% for K=10, the best value for *recall* was 0.724 for K=10 and the best value for *precision* was 0.559 for K=1. The worst value for $hit - rate$ was 0.72 for K=1, the worst value for *recall* was 0.552 for K=1 and the worst value for *precision* was 0.407 for K=10.

Finally, the path query `#from/Cat:#from/Cat:People_from_#from/#to` was the most used path query for the connected pairs in the Path Index for the Spanish Wikipedia with 3,190 pairs covered (out of 7,486), and the second best ranked in the French Wikipedia Path Index with 11,335 pairs covered (out of 30,407). This demonstrates that both Wikipedias share, in general, the same convention for *birthplace* with some minor differences.

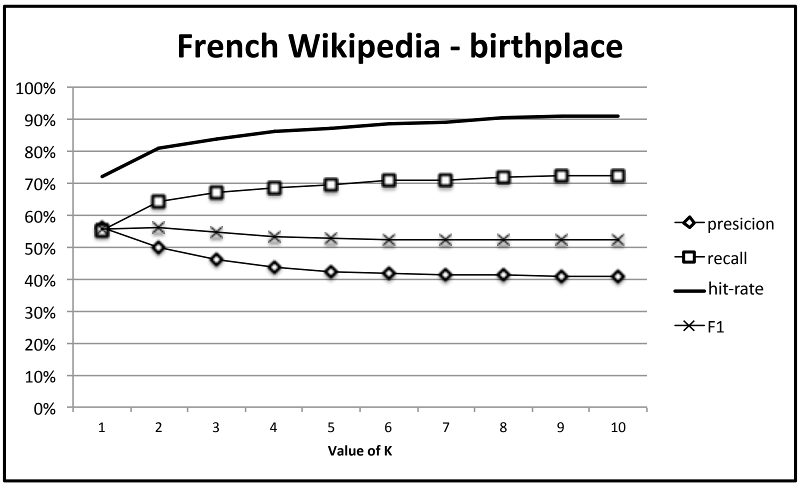


Figure 11.9: BlueFinder evaluation in the French Wikipedia.

11.3 Summary of this Part

In this Part, we introduced the BlueFinder approach in order to improve the Social Web with the Semantic Web. BlueFinder recommends the best representation in Wikipedia of a DBpedia semantic property by following the social conventions.

In relation with the co-evolution elements, BlueFinder is the tool that relates the semantic queries capabilities, the social web tools like pages (articles) and links, and the social conventions. The first improvement is related to the human activities: new pieces of knowledge could be added into the community knowledge base. Moreover, adding new information to Wikipedia reinforce the collaborative knowledge process. This enable users to generate more information in Wikipedia that will be added to the DBpedia knowledge base.

In conclusion, both Wikipedia community and DBpedia have benefits by means of BlueFinder.

IV

Conclusions and Perspective

12

Conclusions

In this thesis we propose to stimulate the co-evolution between the Social and the Semantic Web. The co-evolution involves using the forces from the Social Web to support the Semantic Web, and the new pieces of knowledge that are obtained from the Semantic Web to support the Social Web.

Most of the efforts to use the social forces to improve the Semantic Web are related to the organization of the development of complex ontologies. For example, Ontowiki and Co-Protégé. However, most of the social web information is not written as semantic data, and in consequence is missing in the Semantic Web.

The first half of this thesis introduces an approach to *semantify* the Web by means of using the social forces from the Social Web. The process is focus on regular users rather than knowledge modeling experts. Our objective is to provide a tool that allows the emergence of semantic data by means of identifying and modeling a tagging activity.

We introduced the use of a collaborative knowledge building process to generate semantic data. The process needs the definition of two integrated spaces: the personal space and the shared space. We showed that, on one hand, collaborative tools as Wikis and Semantic Wikis are mainly designed to only support a shared space where users can write the contributions in the shared space. On the other hand, personal semantic wikis only include a private space. The wiki owner must manage with external tools the socialization of it contributions.

In order to integrate the personal and shared spaces, we introduced a model to manage personal and shared knowledge in a P2P semantic wiki. The use of P2P semantic wikis allowed us to exploit the decentralized architecture of the P2P network to represent the shared knowledge, and each peer of the network maintains a space with personal information. In this context, we developed P-Swooki that is a P2P semantic wiki where users can both handle shared semantic knowledge and personal knowledge. With P-Swooki we adapt the collaborative knowledge building process in a semantic wiki. The results of the evaluation of P-Swooki showed that personal knowledge improves the shared navigation with the personal one. Most of the participants have used the personal spaces.

The previous works such as Semantic Mediawiki, SemperWiki, SweetWiki or OntoWiki do not include in the same Wiki the management of personal and shared understanding as introduced by P-Swooki. This makes the P-Swooki approach novel. However, P-Swooki is limited to adding semantic data only to the resources that are described in the network of peers, and it only allows users to use two semantic types of semantic annotations: categories and individuals. In addition, it was necessary to extend the personal annotations management with a mechanism that allows users to publish personal annotations in the shared space and in the body of the wiki text. In order to make the idea of *semantifying* the Web more general, we introduced Semdrops.

Semdrops bootstrap the collaborative knowledge building process, the Semantic Web ontology description and the social tagging activity to generate a semantic description of any Web resource. Semdrops is the evolution of P-Swooki. According to the evaluation results, the simple social tagging model promotes the semantic data proliferation. As the generation of semantic data is part of the collaborative knowledge building process, the continuous social validation generates good quality semantic data. This is important to show the knowledge of the community involved in the *semantification* process.

The evaluation and usability studies demonstrated that *Semdrops* :

- promotes semantic data proliferation thanks to its simple social semantic tagging model.
- promotes the semantic data quality by community regulation.
- is applicable to any web resources thanks to using URI to identify them.
- can be used by a wide community thanks to its simple social semantic tagging and its implementation as a Firefox add-on.
- provides an easy interaction mode as the usability study demonstrated.

Most of the previous work that semantically annotate resources are limited to specific resources such as bookmarks. Semdrops is not restricted to a particular kind of resource. In addition, the open structure of the Semantic Social Tagging model allows users to define new properties instead of having a fixed annotation language as SOBOLEO, Limpens et al, Moninn et al works and Huynh-Kim et al works. Finally, other approaches propose a semantic enrichment of social tagging with a tag ontology [31, 41, 42, 63, 73]. Although the purpose of these works is different from Semdrops, these models can be reused to give semantic to Semdrops tags.

Although Semdrops is integrated in a web browser, the use of it requires an effort from the user to associate the semantic information that appears in the Semdrops side-bar with the page body. Indeed, there is not a visual linkage between a semantic property in the side-bar and a link in the web page. Different client side adaptation artifacts [26, 27] could be used to improve the visual aspects of the existent web page within the Semdrops definition.

Part III introduced the problematic of the semantic gap between Social Web and Semantic Web by means of the cases of DBpedia and Wikipedia. We have presented the BlueFinder approach to fill the gap and then to improve

Wikipedia. In the co-evolution context, BlueFinder uses the Semantic Web to derive new pieces of knowledge that are not currently in the Social Web, and after that it can augment the Social Web by means of injecting the new pieces of knowledge. The bootstrapping is made with the semantic query engines and social conventions to represent information in Wikipedia. BlueFinder is the tool that computes the differences and detects the conventions. Finally, Wikipedia community knowledge is augmented with new relations.

We introduced the definition of BlueFinder algorithm as a collaborative filtering recommender system. BlueFinder learns from Wikipedia the conventions to represent a DBpedia semantic property, and it uses the obtained knowledge to recommend a navigational path that connects disconnected pairs of Wikipedia articles that are related in DBpedia. In this sense, for a disconnected pair of Wikipedia articles, BlueFinder detects similar pairs that are connected in Wikipedia. In order to compute the similarity, we have defined the Semantic Pair similarity function that is based on DBpedia types.

In order to analyze the behavior of our approach, an empirical evaluation has been carried out. The evaluation measured precision, recall, F1, and hit-rate over 20 different properties of DBpedia semantic properties. The results showed that BlueFinder approach could fix more than 80 % of the properties with more than 0.84 hit-rate.

Featured recommendations denoted a high level of hit-rate. In those cases, BlueFinder has a high accuracy level. Although the predictions are good, the evaluation showed us that the algorithm suffered the well-known sparsity problem when the connected pair set was small.

In comparison with other works, Yan Wang et al. [98] introduce a “collaborative approach" to recommend categories to Wikipedia Articles. However, in our approach we deal with the categorization of the article but in the context of expressing a semantic property from DBpedia. Several works like *MORE*[60] use DBpedia as a source data to base the recommendations but the improvement in the Social Web is not exploited to be part of the virtuous cycle as we propose in our work. Panchenko et al. [72] introduces an approach to extract semantic relations between concepts in Wikipedia applying kNN algorithms called Serelex. In addition to the lack of using DBpedia as semantic base, Serelex cannot describe the way that two concepts are related in Wikipedia according to a semantic property, in other words the semantic property is lost in Panchenko et al. approach. Di Noia et al.[22] introduce a strategy to find a similarity among RDF resources in the Linked Open Data. Their work presents a Content based recommender system approach based on the Web of Data. As in our approach, the similarity between resources is done by means of semantic relationships. The main difference lies in the fact that in Di Noia et al. work it is mandatory to discover the semantic relation among the resources and then to analyze a potential similarity. In our work, we already know that the pair of resources are related by the same semantic property and then we only have to compare the types of description. Finally, the Di Noia et al. approach is applied to the Semantic Web world and in our case we complement and augment the information of the Social Web with information from the Semantic Web.

Additionally, in the field of using Semantic measure distance in Recommender System the work of Passant[74] introduces six different versions of the

Linked Data Semantic Distance (LDSD). This approach could be applied to extent the Semantic Pair Similarity (SPS) function and analyze it as a future work.

Nunes et. al. introduced a recommender system to discover semantic relationships among resources from the Social Web[65]. Although the approach of Nunes et. al. is closely related to ours, the main difference is the direction of the information flow. In Nunes et al work, the information of the Social Web is used to improve the Semantic Web and in our case it is in the other way: from the Semantic Web to the Social Web. The work of Nunes is a complement of Semdrops approach.

In the field of improving Wikipedia information, the articles of Adafre et al [2], Sunercan et al. [86] and Hoffman et al. [35] introduced an approach to complete Wikipedia links. The main difference with our work is that they do not use semantic Web features. Besides, our approach proposes to fix more general relations than direct links; we discover and insert navigational paths.

The Wikidata [97] project is a Wikimedia project to build a platform for the collaborative acquisition and maintenance of structured information from Wikipedia. Although Wikidata is not a pure Semantic Web project, it extracts structured information from Media Wiki projects in terms of facts and relationships among facts. Currently, the project is dedicated to interlanguage links among the different versions of a Wikipedia article written in different languages. Wikipedia uses that information in the interlanguage tool bar. Despite the fact that Wikidata is not a pure semantic web project, as in our approach, it is involved in closing the virtuoso cycle of the information.

From the general perspective of this dissertation, the BlueFinder approach promotes the co-evolution between the Semantic Web and the Social Web.

On one hand, as we have described, Wikipedia is a Social Web system that is built by millions of users around the world. Each contribution made on a Wikipedia article is continuously regulated by the community. Additionally, the generation of semantic data from Wikipedia to the Semantic Web is done by the DBpedia project. DBpedia translate the information following social built rules that are regulated in a Wiki environment [1]. These two steps demonstrate that social forces are used in Wikipedia to create the content, and in DBpedia to define the mapping rules to generate the semantic data. Although the personal space is not included in Wikipedia as we proposed with Semdrops, the information flow from the Social Web to the Semantic Web is covered. On the other hand, BlueFinder makes the translation of information from the Semantic Web to the Social Web. BlueFinder detects the Wikipedia conventions that are used to represent a DBpedia semantic property in Wikipedia. In brief, Wikipedia is improved with new pieces of information extracted from the Semantic Web.

Adding these new pieces of information in Wikipedia has both an immediate and a medium term effect. The immediate effect is caused by the fact that a new pair of Wikipedia articles is connected by a navigational path. Therefore, a new pair of articles will be included in the connected pair set that is used by BlueFinder to make the recommendations. Consequently, this reduces the amount of sparse data that is a problem for BlueFinder: there are more connected pairs to learn the social conventions from them. The medium term

[1] http://mappings.dbpedia.org

effect is related to the potential new information that is generated with the presence of the new connected pairs. As we described, the collaborative knowledge building process is retrofit and *augmented* with new pieces of information, and this new cognitive artifacts trigger the generation of new knowledge pieces. The new pieces could be included in DBpedia translation rules, and after that they will be incorporated to the Semantic Web. Then, the virtuous co-evolution cycle continues.

13

Perspectives

This work opens several perspectives:

13.0.1 Semdrops Improvements

As future work, in order to consolidate results we will conduct a bigger evaluation with more resources and we will compare results with other work such as [59]. Also, we will evaluate how lightweight ontologies can emerge from the semantic data. For this pourpose we will profit from the existence of the subsumption concept in our model. Further, we will study inconsistency and conflict problems.

For a more pragmatical point of view, Semdrops needs to be extended to make the captured semantic data more usable in the context of the Semantic Web. For instance, *Semdrops* should adopt the LOD philosophy to share semantic data, and even it could incorporate properties already existing in the semantic web technologies like the *sameAs* property.

In addition to lightweight ontologies, the system can import other lightweight ontologies in order to re-use the vocabulary. The approach could also incorporate the `sameAs` semantic property.

Semdrops with client-side adaptation

As we mentioned in the previous chapter, having in different places the generated semantic data and the current web page requires an extra effort from the user to match them. An example is the use of a *Semdrops property tag* for a navigation link in the web page. From a visual point of view, the link on the web page has no reference to the Semdrops tag. This context could generate a misunderstanding to locate the semantic information that appears in the Semdrops side-bar to the position of the link in the web page, specially in large web pages.

In order to have a better visual linkage between the Semdrops tags and the current web page we could investigate the augmentation of the DOM document in the client side[1] by including the Semdrops tags visualization and management. This approach enriches the user experience with any web resource with the personal and shared information spaces management.

13.0.2 BlueFinder improvements

The BlueFinder approach lies particular strategies that could be improved separately. We can enumerate them and propose different lines to continue researching on:

- **Path Generalization**: The current path generalization strategy tokenizes the paths that connect two Wikipedia articles by means of replacing both pages with the symbols #from and #to. However, we detected some issues when the pattern uses pages that are relative to the semantic of the content instead of the syntax of the name. In consequence, the path queries generalization strategy could include the use of semantic property relations.

- **Similarity Measure**: We introduced the Semantic Pair Similarity (SPS) function to measure the similarity between two pairs of Wikipedia articles. The similarity function is centered in the analysis of similarity in the DBpedia type description of the concerned Wikipedia articles. Although the results of BlueFinder were promising by using the SPS function, other aspects could be analyzed in the selected similarity function.

- **Sparsity problem handling**: Sparsity is a well-known problem in collaborative filtering systems. We have detected sparse data in some of the semantic properties analyzed in the BlueFinder evaluation. Hybrid recommender system techniques are a research line in recommender systems that must be in our approach in order to make BlueFinder more accurate in the recommendations for semantic properties that are highly disconnected in Wikipedia.

13.0.3 Cross Language Combinations

The whole Wikipedia project includes a Wikipedia version in nearly any spoken language in the world. Several social conventions are followed in the English version of Wikipedia and, naturally, in the other languages as well. Additionally, DBpedia has a corresponding mirror of each Wikipedia version. This means that the knowledge base of a non-English DBpedia could contain different pieces of knowledge that are not included in the English version. In this context, BlueFinder approach could be used to analyze the conventions to represent a DBpedia semantic property in different languages of Wikipedia and DBpedia. Several analysis could be done in these directions:

Improving Wikipedia by means of Cross Language Knowledge

- Expressing a semantic property from the English DBpedia in a non-English Wikipedia: the DBpedia connected pairs are taken from the

[1] The adaptation is done in the Web Browser instead in the Web page server

English DBpedia. This implies that some of the pairs are not necessarily connected in the non-English Wikipedia and we re-use the English community knowledge.

- Expressing in a specific language version of Wikipedia a semantic property queried from another language version of DBpedia. This is a generalization of the previous one.

Discovering cross language conventions

In this case, the context is the same that we have defined in the initial adaptation of BlueFinder to be executed in other languages of Wikipedia shown in Section 11.2. Based on the same set of semantic properties, we would like to analyze the similarities and divergences in the conventions that are used in the different Wikipedia version languages.

The analysis must be more detailed and it could include the divergences or similarities based on contrasting particular cases. The analysis involves several issues, such as the equivalence in the similarities among the k neighbors in the different versions of Wikipedia, the development of the articles among the different language versions, and the presence of sparsity in several non-English Wikipedia versions.

Detecting Wikipedia Borders

The idea of this research line is to discover within a specific language version of Wikipedia the articles that are more related with a particular concern by means of their conventions, content and use of the language. The concern could be any particular subject that is not explicit in the Wikipedia structures such as articles or categories. The objective of this research line is to be able to detect groups of articles in Wikipedia that are cross cut by the same concern. A scenario for this could be:

- **County limits:** Wikipedia project has versions of the encyclopedia in different languages. In consequence, people from different countries that have the same spoken language collaborate in the same Wikipedia version, for example the Spanish version of Wikipedia is built by people from Argentina, Spain, Uruguay, etc. The same happens with other languages. The objective in this scenario is to detect the bounds in Wikipedia that could group the articles that are more related to, for example, Argentina, Spain, or Uruguay.

13.0.4 Evolution of the Conventions

Wikipedia is evolving. The evolution is done in terms of the length of the content, the quality of the articles and other aspects. The conventions are also part of this evolution. For example, in the structure of the English Wikipedia the use of *List of* pages eventually evolves to the use of *Categories*. The purpose of this research line is to have a better understanding in the evolution through time of the conventions used in Wikipedia to represent semantic properties.

13.0.5 Semdrops and BlueFinder Mash-up

BlueFinder detects in Wikipedia the conventions that represent a semantic property from DBpedia. The idea of this line of research is to use BlueFinder to detect conventions in other sites of the Social Web different from Wikipedia.

On one hand, the main involved heuristics used in BlueFinder are based on the Wikipedia structures like articles and categories. However, social sites that are not instances of Mediawiki do not have the Wikipedia like structure. On the other hand, Semdrops allows users to tag any web resource with semantic tags that match with Wikipedia structures. Indeed, Semdrops brings to BlueFinder the semantic relationships of a social site and it will be used to obtain the pairs of pages related by a semantic property. Besides, Semdrops defines the idea of articles and categories in the non Wikipedia-like social site to emulate the Wikipedia structure.

A

Appendix

A.1 Social Evaluation

We conducted an evaluation to answer the following question:

- *Does the Wikipedia community accept links added through BlueFinder?*

The results of this evaluation have encouraged us to continue in this research line but also including the pairs context analysis. This evaluation is part of [89].

A.1.1 Method

To answer the above question, we run the Path Index algorithm on the local copy of Wikipedia and on the results of three representative SPARQL queries detailed in Listing A.1. After that, we started enriching Wikipedia by adding new contributions according to the results obtained from the most representative path queries in the path index of each query. After that, we observed whether the Wikipedia community accepted or rejected such contributions.

```
PREFIX r:<http://dbpedia.org/resource/>
PREFIX o:<http://dbpedia.org/ontology/>
PREFIX p:<http://dbpedia.org/property/>

#Q1: Cities and People born there.
```

Query	Contributions	Last edition	Other editions	Rejected
Q_1	78	35.9%	47.4%	16.6%
Q_2	63	25.4%	73%[a]	1.6%
Q_3	70	25.7%	64%	10%

Table A.1: Community Evaluation results. 13 of these contributions were made by Wikipedia users who respect the most representative path queries for Q_2

```
SELECT ?city, ?person WHERE{
?person a o:Person.
?city a o:City.
?person p:birthPlace ?city}

#Q2: Composer and its works.
SELECT ?musician, ?work WHERE{
?work a o:Work.
?work p:musicBy ?musician }

#Q3: Cities and their universities.
SELECT ?city, ?university WHERE{
?university a o:University.
?city a o:City.
?university p:city ?city}
```

Listing A.1: Semantic Queries of the Evaluation

A.1.2 Results and Discussion

Table A.1 details these results for Q_1, Q_2 and Q_3. The first column identifies the query, the second column details the number of contributions that we made in Wikipedia. The *Last edition* column indicates the number of pages that were not edited by any Wikipedia user after our edition; then, the *Other editions* column details the number of pages that were edited by another Wikipedia user but preserves our contribution; and the last column details the contributions that were rejected by the community[1].

According to Table A.1, the rate of rejection is 16.6% for Q_1, 10% for Q_3 and 1.5% for Q_2. Q_1 had 13 rejections. The contribution for these queries consisted in categorizing articles. Two causes of rejections were identified: a more general categorization was preferred to a specific one and a specific category was unnecessarily created. For instance, the category People from Edinburgh was changed by the more specific *Sportspeople from Edinburgh* and the *People from Dayton, Kentucky* category was deleted because *"Dayton, Kentucky is a small community"* and *"this category contains one article and has little possibility for growth"* [2]. The solution was *upmerging* the categorization. On the other hand, the rejections related to the query Q_3 were produced in city articles that include a list of educational institutes in a Wikipedia special page.

The social evaluation demonstrated that contributions derived from the most representative path query in the Path Index were generally accepted by the Wikipedia community.

However, some editions were rejected. Why? Is there another possible interpretation of the Path Index? Although the levels of representation of the most representative path query in the Path Index are sustainable, the Wikipedia community feedback shows us that there are different conventions for the same property of DBpedia. These conventions depend on the articles development and on the community that sustains them; for example,

[1] All the contributions could be checked in the contribution page of magic.towers user in Wikipedia

[2] http://en.wikipedia.org/wiki/Wikipedia:Categories_for_discussion/Log/2012_March_25

Dayton_Kentucky and Edinburgh articles. Therefore, we can conclude that adding missing links in Wikipedia is socially accepted but convention detection must be improved. In this direction, we need to analyze not only the general conventions but also the conventions of a specific community.

A.2 Tables

	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
SC1	0.55909747	0.56096977	0.58041286	0.58425349	0.58175707	0.58103698	0.58055687	0.57801247	0.57671624	0.57460392
SC2	0.35461989	0.36304095	0.36023393	0.36163744	0.35766083	0.35812864	0.35812864	0.35321638	0.34947369	0.35040936
SC3	0.65750802	0.7060703	0.71246004	0.71821088	0.71884984	0.72204471	0.71757185	0.71309906	0.70287538	0.68370605
SC4	0.82781458	0.94701988	0.96026492	0.96026492	0.96026492	0.96688741	0.95364237	0.94701988	0.94701988	0.96026492
SC5	0.69842827	0.72789782	0.74459726	0.7691552	0.77996069	0.78585464	0.79076624	0.78781927	0.7897839	0.7897839
SC6	0.93127149	1	1	1	1	1	1	1	1	1
SC7	0.60176992	0.75382137	0.77473855	0.78278357	0.79243767	0.79082865	0.79324216	0.78519708	0.78278357	0.78519708
SC8*	0.62114537	0.58149779	0.55726874	0.57709253	0.59691632	0.59030837	0.62555069	0.6365639	0.63436121	0.6365639
SC9	0.60696518	0.66169155	0.77611941	0.79104477	0.80099499	0.81094527	0.79601991	0.79104477	0.80597013	0.79601991
SC10	0.89323193	0.91474295	0.92444909	0.95068204	0.95120674	0.95068204	0.95120674	0.9483211	0.95068204	0.95173138
SC11	0.85245901	0.83606559	0.86885244	0.86885244	0.86885244	0.86885244	0.86885244	0.86885244	0.86885244	0.86885244
SC12	0.64381343	0.85626012	0.91864055	0.9304105	0.93364722	0.9343828	0.93364722	0.93394142	0.93423569	0.93408859
SC13	0.76059991	0.77963656	0.76059991	0.76463801	0.76809925	0.77905971	0.77213728	0.76636863	0.77069515	0.76781076
SC14	0.93653274	0.95151466	0.98551202	0.98641753	0.9868291	0.98756999	0.9876523	0.98781693	0.9876523	0.98707604
SC15	0.40721649	0.39175257	0.36597937	0.34020618	0.32474226	0.28865978	0.29896906	0.28350514	0.27319589	0.27319589
SC16	0.6129542	0.6327014	0.6327014	0.62440759	0.63586098	0.62124801	0.61650866	0.6172986	0.61018956	0.60505527
SC17	0.82310295	0.94055098	0.95698404	0.96230066	0.96665055	0.96761721	0.96761721	0.96761721	0.96761721	0.96810055
SC18	0.9778555	0.9918415	0.99766898	0.99766898	0.99825174	0.99766898	0.99825174	0.99825174	0.99825174	0.99825174
SC19	0.67711174	0.68119889	0.66485012	0.65395093	0.66757494	0.67983651	0.67029971	0.68119889	0.6907357	0.69754767
SC20	0.93706292	0.98601401	0.993007	0.993007	0.993007	0.993007	0.993007	0.993007	0.993007	0.993007

Table A.2: Precision with maxRecom=1

	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
SC1	0.54362297	0.4986718	0.45859337	0.42732438	0.40537685	0.38865417	0.37570012	0.3655625	0.35643303	0.34956792
SC2	0.34670565	0.30705652	0.27766082	0.25734892	0.24651071	0.23875244	0.23294348	0.22982456	0.22803119	0.22604288
SC3	0.642492	0.5820021	0.55260915	0.52768904	0.50788075	0.49318424	0.48370606	0.46911609	0.46038339	0.45835996
SC4	0.83885211	0.7682119	0.72295809	0.69205296	0.66887414	0.64900661	0.62582779	0.61810154	0.61479026	0.60485649
SC5	0.70219386	0.66371971	0.6227898	0.59397513	0.58005893	0.55697447	0.54322201	0.53143418	0.52308446	0.51702684
SC6	0.94272625	0.82359678	0.77777779	0.74914092	0.73654068	0.7193585	0.70045817	0.67583048	0.65521193	0.63287514
SC7	0.64494503	0.59761328	0.56503081	0.53848219	0.52118528	0.50992221	0.49678198	0.48913917	0.48659158	0.48163047
SC8*	0.62224668	0.57929516	0.55800295	0.53450811	0.51872247	0.50220263	0.50807637	0.48017621	0.4640235	0.45704845
SC9	0.63764513	0.63349915	0.58457708	0.52819234	0.50829184	0.48175788	0.47346601	0.4543947	0.45024875	0.44527364
SC10	0.90394372	0.88186431	0.86210215	0.8422088	0.82069778	0.80469573	0.78825641	0.77133614	0.75520289	0.73950684
SC11	0.78415298	0.74316943	0.6885246	0.67213112	0.67213112	0.67213112	0.66393441	0.65300548	0.65300548	0.56830603
SC12	0.56032073	0.4701584	0.44951203	0.4442156	0.44306311	0.44080722	0.44016969	0.44063556	0.44156736	0.44220489
SC13	0.75074512	0.72670895	0.68661666	0.65229303	0.63003558	0.61345065	0.59725028	0.58388615	0.5682146	0.5564369
SC14	0.93269128	0.93001592	0.91032815	0.89446825	0.87982935	0.86707002	0.85728788	0.84627098	0.83668095	0.82813358
SC15	0.39776632	0.34278351	0.30584192	0.2757732	0.26288661	0.26030928	0.25171822	0.24312715	0.2371134	0.23367698
SC16	0.63296473	0.58583462	0.53923118	0.51230913	0.48920485	0.47446024	0.46228278	0.4560295	0.44530016	0.43852028
SC17	0.85991621	0.80763656	0.76703721	0.73127115	0.69486064	0.65981954	0.6362977	0.61124539	0.59569842	0.57918477
SC18	0.9770785	0.95862472	0.94046229	0.92103732	0.90297204	0.88480961	0.86820126	0.8529526	0.84090906	0.82925409
SC19	0.66734785	0.60558581	0.55767483	0.52361488	0.49046323	0.49455041	0.48410535	0.47320619	0.46639419	0.46253407
SC20	0.9125874	0.87995338	0.86363637	0.84965032	0.83449882	0.81351984	0.76223779	0.73776221	0.72027969	0.68648016

Table A.3: Precision with maxRecom=3

	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
SC1	0.54347336	0.49525765	0.44925189	0.41172829	0.38200751	0.3580541	0.33863339	0.32267722	0.30907586	0.29807329
SC2	0.34651852	0.29963744	0.26306042	0.23622613	0.21867836	0.20506433	0.19321248	0.18536842	0.1791501	0.17509942
SC3	0.63962728	0.57005328	0.52351439	0.48652822	0.46111822	0.4402343	0.42246005	0.40749735	0.39644301	0.38623002
SC4	0.83697569	0.73256069	0.66523176	0.62428254	0.58013242	0.56456953	0.54392934	0.51357615	0.51280355	0.49359822
SC5	0.70075309	0.66098559	0.61614275	0.5773412	0.55538636	0.526981	0.50717092	0.4868533	0.47293714	0.4599869
SC6	0.94100803	0.82142037	0.77428406	0.74347079	0.7281214	0.69874001	0.67651772	0.64839631	0.62674683	0.60189003
SC7	0.63684636	0.57676321	0.53094661	0.48251542	0.452856	0.43046394	0.40701261	0.39423439	0.38061142	0.37160096
SC8*	0.62224668	0.57929516	0.55690163	0.54405284	0.5294053	0.49629223	0.48715124	0.45969164	0.43836269	0.42834067
SC9	0.63839138	0.61417907	0.5553897	0.4871476	0.44660032	0.41285241	0.39461029	0.36210614	0.35174128	0.34585407
SC10	0.90013552	0.87048793	0.84516001	0.82218868	0.79883701	0.77945524	0.7593258	0.73801154	0.7176854	0.69864899
SC11	0.78005463	0.73497266	0.66338795	0.6224044	0.61967212	0.61256832	0.5877049	0.56393445	0.55874318	0.46010929
SC12	0.55640972	0.42917463	0.3514418	0.32520476	0.31448188	0.30562502	0.301893	0.29943603	0.29887697	0.29832524
SC13	0.75037014	0.72399288	0.68115085	0.64514953	0.61866164	0.59914434	0.57871836	0.56018651	0.5412268	0.52467549
SC14	0.93277359	0.929286	0.90866673	0.89166802	0.87553096	0.86175776	0.85109073	0.8391189	0.8288182	0.81951487
SC15	0.39879724	0.34166667	0.30506873	0.2726804	0.26099655	0.24295533	0.23694158	0.22336771	0.21829897	0.21245705
SC16	0.63407713	0.58013427	0.52343339	0.48619667	0.45221826	0.42360452	0.40231043	0.38657188	0.36781859	0.35430491
SC17	0.85961819	0.79875141	0.75112778	0.708426	0.66236508	0.62185436	0.59117126	0.55980343	0.5374577	0.51637667
SC18	0.97694248	0.9576146	0.93825758	0.91769618	0.89789242	0.87787491	0.85919774	0.84267676	0.82883644	0.81527776
SC19	0.66144413	0.58444595	0.53283381	0.48782924	0.44227976	0.42302451	0.40747049	0.38801089	0.37241143	0.35994551
SC20	0.91317016	0.87937063	0.86188811	0.84580421	0.82937062	0.80722612	0.75116551	0.72202796	0.7041958	0.66981351

Table A.4: Precision with maxRecom=5

	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
SC1	0.54350775	0.49473447	0.44723323	0.40734592	0.37549341	0.34838548	0.32574067	0.30656999	0.28965253	0.27518252
SC2	0.34654367	0.29877707	0.25906479	0.22933082	0.20701353	0.18837811	0.17203227	0.1591152	0.14727108	0.13667633
SC3	0.63993055	0.5691244	0.52003288	0.47811496	0.44618806	0.41908005	0.39633927	0.3748126	0.35954657	0.34581435
SC4	0.83697569	0.71877718	0.64038682	0.58429706	0.54252863	0.49678722	0.44956097	0.39520466	0.35119212	0.33373263
SC5	0.70068765	0.6605038	0.61551255	0.57475436	0.54962188	0.5192467	0.49701139	0.47454065	0.45802611	0.44331408
SC6	0.94100803	0.82142037	0.77428406	0.74172395	0.72603095	0.6964308	0.67398989	0.64477211	0.62236071	0.59685653
SC7	0.63709569	0.5704338	0.52188414	0.47067127	0.43490884	0.40142128	0.37090793	0.34934136	0.3285538	0.31106028
SC8*	0.62224668	0.57929516	0.55690163	0.54405284	0.52933186	0.50460982	0.48954532	0.45721716	0.43498182	0.42340919
SC9	0.63805968	0.61432123	0.55256653	0.48242447	0.43524712	0.39675504	0.3728897	0.33770436	0.32376763	0.31487057
SC10	0.90027034	0.86946613	0.84278983	0.81530678	0.79065698	0.76957369	0.74840403	0.7257753	0.70427018	0.68384778
SC11	0.78005463	0.73497266	0.66284156	0.62185794	0.61873537	0.60821623	0.58225995	0.55685014	0.5500195	0.44797033
SC12	0.55616909	0.42262211	0.32566789	0.27983388	0.23704994	0.20650569	0.18521896	0.16834213	0.15498786	0.14311276
SC13	0.75022769	0.72332799	0.68020552	0.64364415	0.61657268	0.59614813	0.57448816	0.55479264	0.53488505	0.51704645
SC14	0.93276441	0.92909771	0.90835053	0.8911674	0.87466568	0.8605749	0.84953934	0.83708423	0.82634413	0.81666124
SC15	0.39879724	0.34149486	0.30477417	0.27182743	0.26020086	0.24341556	0.23899236	0.22568183	0.21547723	0.20298772
SC16	0.63390976	0.57956731	0.52013373	0.47918814	0.44057199	0.40509385	0.37682956	0.35280409	0.32744601	0.30595511
SC17	0.85969871	0.7979027	0.7472803	0.70265079	0.65512443	0.61204183	0.57691401	0.54231858	0.51682508	0.49322581
SC18	0.97694248	0.95749527	0.9378469	0.91715091	0.89732975	0.87658536	0.85764474	0.84037608	0.82583982	0.81186306
SC19	0.66141385	0.58068913	0.52256006	0.47029126	0.41976944	0.39408207	0.36934301	0.34309176	0.32197419	0.30327809
SC20	0.91317016	0.87937063	0.86188811	0.84580421	0.82937062	0.80722612	0.75116551	0.72132868	0.7032634	0.66888112

Table A.5: Precision with maxRecom=unlimited

	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
SC1	0.43800494	0.43897673	0.45430332	0.45827684	0.45671302	0.45633605	0.45629445	0.45458904	0.45367873	0.45196778
SC2	0.24155983	0.23986882	0.23925191	0.24366897	0.24112476	0.24309236	0.24269702	0.23881842	0.23700945	0.23737203
SC3	0.43306077	0.46208102	0.46893582	0.47190353	0.47292006	0.47668293	0.4739939	0.47276437	0.46584743	0.4536536
SC4	0.32488135	0.39541113	0.39872241	0.39872241	0.39839128	0.40170252	0.39706677	0.39788184	0.39788184	0.40450436
SC5	0.5578258	0.58511788	0.60112965	0.61756712	0.62335581	0.62556601	0.63105059	0.6300683	0.62875855	0.62990457
SC6	0.84508264	0.90762562	0.90762562	0.90762562	0.90762562	0.90762562	0.90762562	0.90762562	0.90762562	0.90762562
SC7	0.37127456	0.46678564	0.47769052	0.48368409	0.49018717	0.48566183	0.48917931	0.48954037	0.48769	0.490841
SC8*	0.61674011	0.57709253	0.55506611	0.5748899	0.59471369	0.58810574	0.62224668	0.63325989	0.63215858	0.63436121
SC9	0.3784056	0.40659797	0.51273394	0.52600098	0.53512204	0.53678036	0.53495616	0.53246862	0.54241884	0.52915186
SC10	0.75174481	0.76809043	0.77502686	0.80048591	0.80073518	0.79983884	0.79986322	0.79679394	0.79924864	0.80005062
SC11	0.65437156	0.64617485	0.67076504	0.67076504	0.67076504	0.67076504	0.67076504	0.67076504	0.67076504	0.67076504
SC12	0.2611765	0.35191169	0.37843123	0.38302147	0.3842195	0.38424015	0.38408077	0.38410529	0.38426468	0.38431373
SC13	0.69052666	0.70837629	0.68747443	0.69256049	0.69563711	0.70578998	0.69996363	0.69448334	0.6979686	0.69573325
SC14	0.8995508	0.91153979	0.94370562	0.94439161	0.94476199	0.94544798	0.94557148	0.9457773	0.94557148	0.94503641
SC15	0.36228523	0.33908936	0.31589347	0.29424399	0.28393471	0.25300688	0.26331615	0.25042954	0.2426976	0.2426976
SC16	0.4130598	0.42236403	0.42110962	0.41720849	0.4237887	0.41265264	0.40963352	0.40996563	0.40539524	0.40174749
SC17	0.65804219	0.74884349	0.76080173	0.76519197	0.76863527	0.76954556	0.76950526	0.77012551	0.76982749	0.77023023
SC18	0.87559897	0.88589418	0.89172173	0.89172173	0.89230442	0.89172173	0.89230442	0.89230442	0.89230442	0.89230442
SC19	0.41337961	0.40677956	0.39467692	0.38888672	0.40080771	0.40848255	0.39939991	0.41029909	0.41294819	0.41567299
SC20	0.8525641	0.88636363	0.88986015	0.88986015	0.88986015	0.88986015	0.88986015	0.88986015	0.88986015	0.88986015

Table A.6: Recall with maxRecom=1

	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
SC1	0.51699674	0.60863918	0.63945735	0.65055877	0.65807134	0.66195214	0.66333818	0.66377044	0.66331786	0.66279823
SC2	0.31598151	0.38482603	0.40648943	0.40459415	0.4061445	0.40468439	0.40431985	0.40532571	0.40714452	0.40725368
SC3	0.62686551	0.68408805	0.71142513	0.71532291	0.72051507	0.72081935	0.72189063	0.71753573	0.71731412	0.72019202
SC4	0.62402904	0.66746962	0.65830845	0.65256894	0.65316814	0.64228827	0.62720931	0.62489146	0.62923855	0.62824517
SC5	0.68020862	0.78142715	0.8129105	0.82700676	0.83951491	0.84181637	0.84946203	0.85637105	0.85871226	0.86296892
SC6	0.92396498	0.92396498	0.92396498	0.9256832	0.9256832	0.9256832	0.9256832	0.92740142	0.92740142	0.92691052
SC7	0.62514704	0.68368024	0.70161337	0.70956647	0.71303928	0.71824175	0.72103423	0.72256178	0.72736967	0.72565436
SC8*	0.62114537	0.66299558	0.73458147	0.70484579	0.68612337	0.68722469	0.73678416	0.73458147	0.73458147	0.72577095
SC9	0.5997631	0.74628049	0.74628049	0.7345534	0.75818527	0.75896704	0.7680881	0.7705757	0.7680881	0.75689411
SC10	0.90668595	0.92884719	0.93302476	0.93327647	0.93249387	0.93310159	0.93373805	0.93511093	0.93507159	0.93459249
SC11	0.75546449	0.82650274	0.83743167	0.85109288	0.85109288	0.85109288	0.85109288	0.85109288	0.85109288	0.85109288
SC12	0.46564922	0.4925403	0.49517292	0.49533552	0.49596646	0.49600425	0.49635589	0.49733678	0.49858236	0.49953908
SC13	0.74006552	0.81209981	0.82372129	0.82447428	0.83855504	0.84063172	0.84453249	0.84560895	0.84633005	0.84693092
SC14	0.92459154	0.96306562	0.9665491	0.96734893	0.96945488	0.97010523	0.97079808	0.97103816	0.97123021	0.9714703
SC15	0.38462198	0.43874571	0.46022338	0.44733676	0.43445018	0.4499141	0.43445018	0.43256015	0.43084192	0.43599656
SC16	0.6019963	0.6974259	0.71639317	0.7226032	0.72418451	0.73263025	0.73407632	0.73863417	0.73320746	0.73185432
SC17	0.84923017	0.88726574	0.89412898	0.89944553	0.90128183	0.90011781	0.90012586	0.89948547	0.89974725	0.89915919
SC18	0.95274866	0.96077114	0.96255827	0.9635101	0.96409285	0.96419001	0.96475333	0.96475333	0.96533602	0.96553028
SC19	0.60766077	0.67900503	0.68714911	0.69164503	0.69075948	0.72012675	0.71887785	0.72288936	0.72186756	0.72204924
SC20	0.9178322	0.92132866	0.92132866	0.92132866	0.92132866	0.92132866	0.92132866	0.92132866	0.9178322	0.9178322

Table A.7: Recall with maxRecom=3

	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
SC1	0.52174634	0.62300438	0.66581887	0.69106036	0.70459533	0.71079922	0.71412855	0.71638829	0.71759957	0.71824574
SC2	0.32639778	0.40929681	0.447438	0.46501175	0.47752386	0.4735688	0.47192442	0.47312355	0.47415483	0.47592676
SC3	0.64310545	0.7196005	0.75509501	0.76880562	0.78634554	0.79108644	0.79470301	0.79826605	0.80415303	0.80481333
SC4	0.74868441	0.78067744	0.77372378	0.77478027	0.7608099	0.75953269	0.75681579	0.74587286	0.76622438	0.74624646
SC5	0.68399054	0.79933107	0.84558189	0.86789691	0.88526756	0.8918491	0.89990413	0.90375149	0.90563428	0.90615821
SC6	0.93142694	0.93142694	0.93142694	0.9326542	0.9321633	0.9321633	0.9321633	0.93388152	0.93388152	0.93388152
SC7	0.64921999	0.72544682	0.75582618	0.76266766	0.77152872	0.77475601	0.77454686	0.78033358	0.7792753	0.78077352
SC8*	0.62114537	0.66299558	0.73458147	0.77422905	0.80066079	0.74889868	0.77422905	0.78744495	0.78744495	0.78744495
SC9	0.62074155	0.76891732	0.79702675	0.78849798	0.80781806	0.81569535	0.83073914	0.83322674	0.8322317	0.82891494
SC10	0.9205482	0.94464034	0.9516052	0.95417666	0.96528757	0.96716762	0.96799457	0.96879494	0.96858919	0.96874595
SC11	0.75546449	0.82650274	0.85382515	0.86748636	0.86748636	0.86748636	0.86748636	0.86748636	0.88387978	0.88387978
SC12	0.49880654	0.53617585	0.54335934	0.54449874	0.54438698	0.5407173	0.5393784	0.53812462	0.53826004	0.53795624
SC13	0.74303913	0.8183952	0.83237463	0.83961535	0.85607362	0.85996747	0.86643314	0.86939406	0.87113911	0.87345827
SC14	0.92631084	0.96479023	0.96872032	0.97037351	0.97223747	0.97316766	0.97408277	0.9743818	0.97472757	0.9752599
SC15	0.3910653	0.45034364	0.48728523	0.50790375	0.51649487	0.51735395	0.50790375	0.4993127	0.51374573	0.524055
SC16	0.61519593	0.7253623	0.760566	0.78001982	0.7902953	0.79582405	0.80041766	0.8056013	0.80660844	0.80892587
SC17	0.86527866	0.90318173	0.91553932	0.9232077	0.9273603	0.929461	0.92935634	0.93008631	0.92962712	0.92888606
SC18	0.95583075	0.96437776	0.96676701	0.96791309	0.96839875	0.96878725	0.96901387	0.96920812	0.96972609	0.96992034
SC19	0.62034404	0.70754296	0.74172616	0.76332015	0.76241189	0.76634014	0.76708943	0.7728796	0.76920116	0.76556808
SC20	0.92132866	0.92482519	0.92482519	0.92482519	0.92482519	0.92482519	0.92482519	0.92482519	0.92482519	0.92482519

Table A.8: Recall with maxRecom=5

	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
SC1	0.52223217	0.6256057	0.67232114	0.70206177	0.72386509	0.73824543	0.74960518	0.75908047	0.76648486	0.77287817
SC2	0.32784864	0.41767994	0.46473888	0.4958798	0.52083838	0.53677511	0.55121046	0.56406033	0.57583469	0.5840981
SC3	0.64556527	0.72518849	0.76951396	0.79182494	0.81031907	0.81816381	0.82687366	0.83192688	0.84419531	0.84887046
SC4	0.77263844	0.81995004	0.82646215	0.84298694	0.87345052	0.87919003	0.88029379	0.88139755	0.88139755	0.88967568
SC5	0.68427122	0.80180562	0.85296798	0.8776499	0.89669055	0.90543318	0.91553468	0.9219197	0.92432642	0.92853403
SC6	0.93339062	0.93339062	0.93339062	0.93510884	0.93510884	0.93510884	0.93510884	0.936827	0.936827	0.936827
SC7	0.65574986	0.73508817	0.77924335	0.79577529	0.8122744	0.82135665	0.83232796	0.83654779	0.84149545	0.84651023
SC8*	0.62114537	0.66299558	0.73458147	0.77422905	0.80066079	0.81387663	0.82268721	0.83810574	0.84471369	0.84691632
SC9	0.62074155	0.77306324	0.8024165	0.81337363	0.83244491	0.84405357	0.85483301	0.85856432	0.86519784	0.8708837
SC10	0.92311311	0.94841415	0.95672655	0.9597674	0.97230488	0.9745481	0.97590595	0.97714108	0.97786689	0.97845715
SC11	0.75546449	0.82650274	0.85382515	0.86748636	0.86748636	0.86748636	0.86748636	0.86748636	0.88387978	0.88387978
SC12	0.50386643	0.55052716	0.56817997	0.58043253	0.59017897	0.59744	0.60334343	0.60836595	0.61234164	0.61631191
SC13	0.74361086	0.81908232	0.83405679	0.84276789	0.85959625	0.86460608	0.87207723	0.87645179	0.8795765	0.8835184
SC14	0.92656755	0.96504581	0.96899158	0.97077101	0.97279739	0.97386754	0.97506666	0.97560173	0.97605449	0.97671986
SC15	0.3910653	0.45034364	0.48728523	0.50790375	0.53195876	0.55773199	0.58178693	0.59725088	0.61271477	0.61786944
SC16	0.6167503	0.73162514	0.77111274	0.79692	0.81356668	0.8254813	0.83591413	0.8437674	0.85194439	0.85694528
SC17	0.8674283	0.90771544	0.9210431	0.92971075	0.9352836	0.93831843	0.93920046	0.94088811	0.94189501	0.94335765
SC18	0.95589548	0.96444249	0.96704543	0.96879369	0.9695707	0.97007579	0.97075564	0.97075564	0.97143549	0.97162974
SC19	0.62175184	0.71166581	0.75244689	0.77592558	0.78616625	0.79477209	0.79967672	0.80698824	0.80948597	0.8141408
SC20	0.92132866	0.92482519	0.92482519	0.92482519	0.92482519	0.92482519	0.92482519	0.92482519	0.92482519	0.92482519

Table A.9: Recall with maxRecom=unlimited

	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
SC1	0.49119818	0.49253169	0.50967306	0.51365381	0.51170665	0.51119143	0.51097947	0.50892454	0.50785166	0.5059607
SC2	0.28736946	0.28887308	0.28753525	0.29115775	0.28805262	0.28960511	0.28932437	0.28496492	0.28245848	0.28302148
SC3	0.52218795	0.55859488	0.56559879	0.56956917	0.57051033	0.57425284	0.57088697	0.56857783	0.56032568	0.54541361
SC4	0.46663046	0.55788672	0.5634771	0.5634771	0.56314635	0.5675931	0.56068283	0.56034136	0.56034136	0.56922638
SC5	0.62025881	0.64874476	0.66521597	0.68507582	0.69291997	0.69660866	0.70193779	0.70016825	0.70013183	0.70084178
SC6	0.88608611	0.95157623	0.95157623	0.95157623	0.95157623	0.95157623	0.95157623	0.95157623	0.95157623	0.95157623
SC7	0.45922229	0.57655412	0.59098798	0.59791493	0.60569978	0.6017676	0.60516399	0.60308212	0.60096604	0.60406804
SC8*	0.61893487	0.57928675	0.55616522	0.57598907	0.59581292	0.58920497	0.62389427	0.6349076	0.63325799	0.63546067
SC9	0.46617785	0.50368828	0.61751443	0.631854	0.64160556	0.64597642	0.63988495	0.63649756	0.6484381	0.63571441
SC10	0.81640357	0.83502656	0.84316921	0.86914289	0.86950904	0.86876136	0.86899471	0.86597902	0.86841303	0.86932421
SC11	0.74039507	0.7289567	0.75706577	0.75706577	0.75706577	0.75706577	0.75706577	0.75706577	0.75706577	0.75706577
SC12	0.371604	0.4988164	0.53604168	0.5426504	0.54440325	0.54454899	0.54426396	0.54433858	0.54454863	0.54457283
SC13	0.72387141	0.74230009	0.7221908	0.72681665	0.73007452	0.7406171	0.73428118	0.72865731	0.73253119	0.72999716
SC14	0.91766936	0.93109834	0.96415585	0.96494716	0.96533751	0.96605003	0.96615392	0.96634007	0.96615392	0.96559888
SC15	0.38343909	0.36352354	0.33909693	0.31556025	0.30297056	0.26965997	0.28001228	0.2659429	0.25704524	0.25704524
SC16	0.49353465	0.50656635	0.50566304	0.50019991	0.50860333	0.49590769	0.4922176	0.49270907	0.48714384	0.48287401
SC17	0.73137528	0.83382004	0.84769022	0.85250115	0.85634506	0.85728931	0.85726428	0.85764909	0.85746419	0.85790384
SC18	0.92390645	0.93587893	0.94172484	0.94172484	0.94230944	0.94172484	0.94230944	0.94230944	0.94230944	0.94230944
SC19	0.51335424	0.509381	0.49531722	0.48773238	0.50088644	0.5103308	0.50054735	0.51213157	0.51688361	0.52092409
SC20	0.89281869	0.93353707	0.93860829	0.93860829	0.93860829	0.93860829	0.93860829	0.93860829	0.93860829	0.93860829

Table A.10: F1 with maxRecom=1

	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
SC1	0.52997565	0.548195	0.53412998	0.51582515	0.50170171	0.48975617	0.47970557	0.47146958	0.46369833	0.4577257
SC2	0.33063135	0.34157053	0.32994562	0.31459463	0.3068051	0.30032283	0.29558793	0.2933284	0.29233375	0.29072255
SC3	0.63458258	0.62892944	0.62204015	0.60734421	0.5957945	0.5856601	0.57926983	0.56732321	0.56082231	0.56019115
SC4	0.71566731	0.71430624	0.68912029	0.67173123	0.66092783	0.64563	0.62651783	0.62147796	0.62193054	0.61632901
SC5	0.69102639	0.71777982	0.70526189	0.69138312	0.68607652	0.67039359	0.66267216	0.65586269	0.65013766	0.6466369
SC6	0.93325132	0.87089866	0.84459233	0.82810742	0.82035077	0.809582	0.79747355	0.78187835	0.76790011	0.75217849
SC7	0.63489175	0.63775617	0.62595826	0.61229807	0.60220093	0.59641457	0.58826089	0.58336717	0.58310258	0.57898062
SC8*	0.62169558	0.61832565	0.63423109	0.60797125	0.59079361	0.58032304	0.601421	0.58073896	0.5687663	0.56088442
SC9	0.61812425	0.68528056	0.65560508	0.61451089	0.60858488	0.58939475	0.58582002	0.57167995	0.56770951	0.56069547
SC10	0.90531278	0.90474617	0.89616245	0.88540715	0.87303138	0.8641547	0.85485178	0.84536445	0.83556694	0.82568282
SC11	0.76954144	0.78262401	0.75571275	0.75109899	0.75109899	0.75109899	0.74595338	0.73900527	0.73900527	0.68152964
SC12	0.50861704	0.48108917	0.471239	0.46838486	0.46802455	0.46677962	0.46657735	0.46727234	0.46834603	0.4691267
SC13	0.74536705	0.76703513	0.74894595	0.72834611	0.71949184	0.70929414	0.69968551	0.69078845	0.679932	0.6716181
SC14	0.92862374	0.94625229	0.93759662	0.9294821	0.92247027	0.91569835	0.91051894	0.90437162	0.89894885	0.89409363
SC15	0.39108375	0.38487312	0.36747676	0.34120256	0.32756376	0.32980275	0.31875271	0.31128931	0.30588391	0.30427471
SC16	0.61709219	0.63677835	0.61531383	0.5995506	0.58394212	0.57593679	0.56730598	0.56390595	0.55408621	0.54842776
SC17	0.85453975	0.84558058	0.82572132	0.80668652	0.78472352	0.76146048	0.74556005	0.72786802	0.71681374	0.70454419
SC18	0.96476018	0.95969677	0.95138204	0.94179511	0.93253201	0.92279583	0.91393429	0.9054147	0.89883685	0.89221853
SC19	0.63610727	0.64019734	0.61567867	0.59601349	0.57362956	0.58639282	0.57858264	0.57198727	0.5666678	0.56386471
SC20	0.91520226	0.90016586	0.89155018	0.88403898	0.87576675	0.86407447	0.83426648	0.8193903	0.80714375	0.78547496

Table A.11: F1 with maxRecom=3

	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
SC1	0.53238827	0.55183434	0.53650475	0.51601744	0.49541691	0.47621983	0.45941585	0.44494247	0.43206003	0.42130446
SC2	0.33615735	0.34598595	0.3313258	0.31329718	0.29998219	0.28619903	0.27417418	0.26637277	0.26004666	0.25600967
SC3	0.64136165	0.63615614	0.61833286	0.59593016	0.58133674	0.56567448	0.5516603	0.53956074	0.53107101	0.52196771
SC4	0.79037195	0.75585407	0.71538782	0.69143689	0.65829903	0.64769775	0.63295156	0.60830176	0.61440814	0.59418076
SC5	0.6922704	0.72360516	0.7128554	0.69341189	0.68255883	0.66250008	0.64872903	0.63281012	0.62137961	0.61021471
SC6	0.93619299	0.87297171	0.84561694	0.82738596	0.81760442	0.79874724	0.7840274	0.76538438	0.75009185	0.73200208
SC7	0.6429736	0.64261681	0.62373614	0.59107596	0.57072163	0.55343348	0.53361738	0.52382553	0.5114311	0.50354499
SC8*	0.62169558	0.61832565	0.63351911	0.63904619	0.63737291	0.59697288	0.5980221	0.58050072	0.56319845	0.55485886
SC9	0.62944275	0.68289232	0.65462154	0.60222822	0.57520175	0.54822743	0.53506076	0.50482422	0.49448806	0.48806804
SC10	0.91022742	0.90604949	0.89522958	0.8832792	0.8742097	0.86322457	0.8510561	0.83780074	0.82447106	0.81182134
SC11	0.76756269	0.77805507	0.74665493	0.72478777	0.72293186	0.71807432	0.70069963	0.68352425	0.68467206	0.60518545
SC12	0.52603585	0.47674513	0.42681929	0.40720448	0.39866352	0.39051986	0.38711539	0.3847695	0.38434216	0.3838084
SC13	0.74668664	0.76830506	0.74920797	0.72964746	0.71825761	0.70624417	0.69393528	0.68135065	0.66765112	0.6555627
SC14	0.92953098	0.94670534	0.93773299	0.92935741	0.92135352	0.9140805	0.90844274	0.90170598	0.89587075	0.89062983
SC15	0.39489344	0.388549	0.37522495	0.35485068	0.34676495	0.33063886	0.32313693	0.30865741	0.3064025	0.30234176
SC16	0.62449384	0.64467049	0.62010252	0.59901774	0.5752629	0.55290598	0.53547662	0.52244562	0.50524312	0.49277651
SC17	0.86243916	0.84776264	0.82522416	0.80168033	0.77277631	0.7451604	0.72265542	0.69893163	0.68112659	0.66376179
SC18	0.96627134	0.96098429	0.952299	0.94213599	0.93181372	0.92109323	0.91080761	0.90152436	0.89376301	0.88590109
SC19	0.64023519	0.64013028	0.62016189	0.59524447	0.55981022	0.54513252	0.5322271	0.5166477	0.50185031	0.48966587
SC20	0.91723126	0.90152532	0.89224815	0.88355136	0.87450081	0.86203343	0.82899839	0.81094015	0.79956985	0.77692884

Table A.12: F1 with maxRecom=5

	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
SC1	0.53265762	0.5525263	0.53715008	0.515558	0.49448216	0.47337875	0.45413649	0.43674976	0.42042691	0.40585923
SC2	0.33693704	0.34836197	0.33267993	0.31362066	0.29627067	0.27888361	0.2622245	0.2482124	0.23455434	0.22151835
SC3	0.64273554	0.63774759	0.62064064	0.59622246	0.57549167	0.55425793	0.53583884	0.51679206	0.50430667	0.49142933
SC4	0.80352128	0.76603746	0.72162235	0.69019872	0.66932046	0.63485116	0.59517133	0.54571801	0.50225943	0.48538792
SC5	0.6923821	0.72432774	0.71504182	0.69461805	0.68151349	0.65999836	0.6442709	0.62656754	0.61252916	0.60011339
SC6	0.93718386	0.87383324	0.84642524	0.82726508	0.8174122	0.79831171	0.78336263	0.76383442	0.74788213	0.72916126
SC7	0.64628822	0.64237773	0.62511128	0.59149528	0.56650102	0.5392803	0.51314467	0.49286351	0.47258958	0.45494542
SC8*	0.62169558	0.61832565	0.63351911	0.63904619	0.63731968	0.62297213	0.61382818	0.59166145	0.57425392	0.56456739
SC9	0.62928146	0.68461078	0.6544562	0.60563654	0.57162029	0.53978109	0.5192678	0.48474213	0.4712047	0.46251681
SC10	0.91154861	0.90722585	0.89615124	0.88165879	0.87212282	0.86001629	0.84714693	0.83290631	0.81881851	0.80504572
SC11	0.76756269	0.77805507	0.74630868	0.72441715	0.72229397	0.71507537	0.69681507	0.6782946	0.67808264	0.59458929
SC12	0.52872741	0.47816908	0.41402566	0.37761486	0.33824226	0.30692312	0.28342879	0.26371199	0.24736574	0.23228665
SC13	0.74690461	0.76823288	0.74931544	0.72986847	0.71807986	0.70570844	0.69267249	0.67947727	0.66523165	0.6523369
SC14	0.92965567	0.94673061	0.93769163	0.92926759	0.92112529	0.91372281	0.90798503	0.90105051	0.89498168	0.88954794
SC15	0.39489344	0.38843787	0.37500206	0.35412762	0.34946525	0.33891544	0.33880639	0.32758126	0.31882966	0.30558279
SC16	0.62521231	0.64677924	0.62123191	0.59849882	0.57160294	0.54348147	0.5194785	0.49756256	0.47306776	0.45091873
SC17	0.86354625	0.84927404	0.82511264	0.80038887	0.77052826	0.74084735	0.71477175	0.68805122	0.66742754	0.6477707
SC18	0.96630442	0.96095634	0.95222241	0.94226521	0.93205249	0.920964	0.91070145	0.90087295	0.89274037	0.88459039
SC19	0.64096987	0.63954043	0.61677891	0.58563006	0.54730707	0.52690303	0.50530368	0.48148131	0.4607031	0.4419311
SC20	0.91723126	0.90152532	0.89224815	0.88355136	0.87450081	0.86203343	0.82899839	0.81049889	0.79896843	0.77630126

Table A.13: F1 with maxRecom=unlimited

	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
SC1	0.55909747	0.56096977	0.58041286	0.58425349	0.58175707	0.58103698	0.58055687	0.57801247	0.57671624	0.57460392
SC2	0.35461989	0.36304095	0.36023393	0.36163744	0.35766083	0.35812864	0.35812864	0.35321638	0.34947369	0.35040936
SC3	0.65750802	0.7060703	0.71246004	0.71821088	0.71884984	0.72204471	0.71757185	0.71309906	0.70287538	0.68370605
SC4	0.82781458	0.94701988	0.96026492	0.96026492	0.96026492	0.96688741	0.95364237	0.94701988	0.94701988	0.96026492
SC5	0.69842827	0.72789782	0.74459726	0.7691552	0.77996069	0.78585464	0.79076624	0.78781927	0.7897839	0.7897839
SC6	0.93127149	1	1	1	1	1	1	1	1	1
SC7	0.60176992	0.75382137	0.77473855	0.78278357	0.79243767	0.79082865	0.79324216	0.78519708	0.78278357	0.78519708
SC8*	0.62114537	0.58149779	0.55726874	0.57709253	0.59691632	0.59030837	0.62555069	0.6365639	0.63436121	0.6365639
SC9	0.60696518	0.66169155	0.77611941	0.79104477	0.80099499	0.81094527	0.79601991	0.79104477	0.80597013	0.79601991
SC10	0.89323193	0.91474295	0.92444909	0.95068204	0.95120674	0.95068204	0.95120674	0.9483211	0.95068204	0.95173138
SC11	0.85245901	0.83606559	0.86885244	0.86885244	0.86885244	0.86885244	0.86885244	0.86885244	0.86885244	0.86885244
SC12	0.64381343	0.85626012	0.91864055	0.9304105	0.93364722	0.9343828	0.93364722	0.93394142	0.93423569	0.93408859
SC13	0.76059991	0.77963656	0.76059991	0.76463801	0.76809925	0.77905971	0.77213728	0.76636863	0.77069515	0.76781076
SC14	0.93653274	0.95151466	0.98551202	0.98641753	0.9868291	0.98756999	0.9876523	0.98781693	0.9876523	0.98707604
SC15	0.40721649	0.39175257	0.36597937	0.34020618	0.32474226	0.28865978	0.29896906	0.28350514	0.27319589	0.27319589
SC16	0.6129542	0.6327014	0.6327014	0.62440759	0.63586098	0.62124801	0.61650866	0.6172986	0.61018956	0.60505527
SC17	0.82310295	0.94055098	0.95698404	0.96230066	0.96665055	0.96761721	0.96761721	0.96761721	0.96761721	0.96810055
SC18	0.9778555	0.9918415	0.99766898	0.99766898	0.99825174	0.99766898	0.99825174	0.99825174	0.99825174	0.99825174
SC19	0.67711174	0.68119889	0.66485012	0.65395093	0.66757494	0.67983651	0.67029971	0.68119889	0.6907357	0.69754767
SC20	0.93706292	0.98601401	0.993007	0.993007	0.993007	0.993007	0.993007	0.993007	0.993007	0.993007

Table A.14: Hit-rate with maxRecom=1

	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
SC1	0.62006724	0.70969754	0.73590976	0.74349499	0.74930388	0.75242442	0.7523284	0.75204033	0.75084013	0.74997598
SC2	0.42549708	0.50385964	0.52421051	0.52046782	0.51929826	0.51625729	0.51415205	0.51625729	0.51976609	0.51976609
SC3	0.78083068	0.82044727	0.84472841	0.84281152	0.84536743	0.84984028	0.85047925	0.84728438	0.84664536	0.85431308
SC4	0.94701988	0.98675495	0.98675495	0.98675495	0.99337751	0.99337751	0.99337751	0.98675495	0.99337751	1
SC5	0.78880155	0.8772102	0.90176815	0.9125737	0.92043221	0.92043221	0.92534381	0.92927307	0.93516701	0.93811393
SC6	1	1	1	1	1	1	1	1	1	1
SC7	0.83266288	0.87369269	0.88736928	0.8889783	0.89139181	0.89139181	0.89702332	0.89943683	0.90265489	0.90345937
SC8*	0.623348	0.66519827	0.73788548	0.70925111	0.68942732	0.68942732	0.74008811	0.73788548	0.73788548	0.7290749
SC9	0.72636819	0.86567163	0.86069649	0.85074627	0.87064677	0.86069649	0.86567163	0.86567163	0.86567163	0.85572141
SC10	0.9598636	0.9732424	0.97455406	0.97455406	0.97376704	0.97455406	0.97402936	0.97586566	0.97612804	0.97665268
SC11	0.88524592	0.91803277	0.90163934	0.90163934	0.90163934	0.90163934	0.90163934	0.90163934	0.90163934	0.90163934
SC12	0.8995145	0.93673682	0.94659412	0.9471826	0.9474768	0.9471826	0.9473297	0.9473297	0.94777107	0.94777107
SC13	0.78079033	0.85203344	0.86126333	0.86270553	0.87655032	0.87856936	0.88203055	0.88347274	0.88578022	0.88549179
SC14	0.9545604	0.99094498	0.99209744	0.99217975	0.99300295	0.99333227	0.99374384	0.99374384	0.99382615	0.99382615
SC15	0.42268041	0.47938144	0.5	0.48969072	0.47938144	0.49484536	0.48453608	0.48453608	0.47938144	0.48453608
SC16	0.75750393	0.84557664	0.85900474	0.86176938	0.86532384	0.87203789	0.87361771	0.87835705	0.87282783	0.87282783
SC17	0.96085066	0.98356694	0.98453361	0.98646688	0.98791689	0.98598355	0.98598355	0.98501694	0.98501694	0.98598355
SC18	0.99825174	0.99883449	0.99883449	0.99883449	0.99883449	0.99883449	0.99883449	0.99883449	0.99883449	0.99883449
SC19	0.79427791	0.85149866	0.85149866	0.85149866	0.85694826	0.88419616	0.8814714	0.88555861	0.88555861	0.88692099
SC20	0.98601401	0.993007	0.993007	0.993007	0.993007	0.993007	0.993007	0.993007	0.993007	0.993007

Table A.15: F1 with maxRecom=unlimited

	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
SC1	0.62289965	0.72021121	0.75693709	0.77839655	0.78915024	0.79318291	0.79486316	0.79625541	0.79577529	0.79543924
SC2	0.4311111	0.52374268	0.55859649	0.57637429	0.58760232	0.58222222	0.5794152	0.57871348	0.58058482	0.58362573
SC3	0.78722048	0.84089458	0.86837059	0.87156552	0.88434505	0.88753992	0.89137381	0.89456868	0.8996805	0.90031952
SC4	0.98013246	0.98675495	0.98675495	0.99337751	0.99337751	0.99337751	0.99337751	0.99337751	1	1
SC5	0.7897839	0.88506877	0.91552061	0.9312377	0.94204324	0.94499016	0.94891948	0.95186639	0.95186639	0.95186639
SC6	1	1	1	1	1	1	1	1	1	1
SC7	0.83990347	0.88576025	0.9106999	0.91150445	0.91552693	0.91391796	0.91150445	0.91874498	0.91874498	0.92035401
SC8*	0.623348	0.66519827	0.73788548	0.77753305	0.80396473	0.75110132	0.77753305	0.79074889	0.79074889	0.79074889
SC9	0.72636819	0.87064677	0.89054728	0.880597	0.89552242	0.89552242	0.9004975	0.9004975	0.9004975	0.89552242
SC10	0.96091288	0.97402936	0.97743964	0.97796434	0.98898214	0.98976916	0.9902938	0.99055612	0.99003148	0.9902938
SC11	0.88524592	0.91803277	0.91803277	0.91803277	0.91803277	0.91803277	0.91803277	0.91803277	0.93442625	0.93442625
SC12	0.90804768	0.94541711	0.95086068	0.95174342	0.9523319	0.95189053	0.95144916	0.95130205	0.95086068	0.95041931
SC13	0.78107876	0.8531872	0.865013	0.87078166	0.88722241	0.8892414	0.89529854	0.8984713	0.90077877	0.90193254
SC14	0.95505434	0.9912743	0.99250907	0.99275601	0.99357921	0.99390846	0.99440235	0.99440235	0.99448466	0.99464929
SC15	0.42783505	0.48969072	0.52061856	0.54123712	0.55154639	0.556701	0.55154639	0.54123712	0.55154639	0.56185567
SC16	0.76184833	0.85900474	0.8807267	0.89218009	0.89691943	0.89810425	0.90284359	0.90402842	0.90679306	0.91034752
SC17	0.96616721	0.98550022	0.98743355	0.98936683	0.99081683	0.99081683	0.99081683	0.99081683	0.99081683	0.9903335
SC18	0.99825174	0.99883449	0.99883449	0.99883449	0.99883449	0.99883449	0.99883449	0.99883449	0.99883449	0.99883449
SC19	0.79700273	0.86239785	0.88828337	0.90463215	0.90326977	0.90735698	0.90735698	0.91008174	0.91008174	0.91008174
SC20	0.98601401	0.993007	0.993007	0.993007	0.993007	0.993007	0.993007	0.993007	0.993007	0.993007

Table A.16: Hit-rate with maxRecom=5

	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
SC1	0.62304372	0.72160345	0.76207393	0.78650981	0.80456072	0.81603456	0.82491601	0.83197314	0.83730197	0.8419587
SC2	0.43157896	0.52795321	0.57192981	0.60210526	0.62690055	0.64280701	0.65660816	0.66736841	0.67812866	0.6853801
SC3	0.78722048	0.8421725	0.87348241	0.886262	0.89648563	0.90095848	0.90607029	0.90926516	0.91884983	0.92012781
SC4	0.98013246	0.99337751	0.99337751	0.99337751	1	1	1	1	1	1
SC5	0.7897839	0.88506877	0.91944993	0.93516701	0.9459725	0.94990176	0.95677799	0.96070725	0.96267194	0.96365422
SC6	1	1	1	1	1	1	1	1	1	1
SC7	0.84070796	0.88656473	0.92035401	0.92276752	0.93563956	0.9396621	0.94288015	0.94448912	0.94609815	0.94851166
SC8*	0.623348	0.66519827	0.73788548	0.77753305	0.80396473	0.81718063	0.82599121	0.84140968	0.84801763	0.85022026
SC9	0.72636819	0.87562191	0.89552242	0.89552242	0.91044778	0.91044778	0.91044778	0.91044778	0.91044778	0.91542292
SC10	0.96091288	0.97455406	0.97796434	0.97848898	0.98976916	0.99055612	0.99108082	0.99160546	0.99160546	0.99186778
SC11	0.88524592	0.91803277	0.91803277	0.91803277	0.91803277	0.91803277	0.91803277	0.91803277	0.93442625	0.93442625
SC12	0.90819478	0.94644696	0.952479	0.95409739	0.9552744	0.95600998	0.95689273	0.9573341	0.95777547	0.95806974
SC13	0.78107876	0.8531872	0.865013	0.87107009	0.88751084	0.89039516	0.89731759	0.90135562	0.90423995	0.90712428
SC14	0.95521897	0.99143893	0.9926737	0.99292064	0.99374384	0.99407309	0.99473166	0.99489629	0.99506092	0.99522555
SC15	0.42783505	0.48969072	0.52061856	0.54123712	0.56701028	0.59278351	0.61855668	0.63402063	0.64948452	0.65463918
SC16	0.76224327	0.86137444	0.88428122	0.89968401	0.90837282	0.91390204	0.91943127	0.92338073	0.92851502	0.93246442
SC17	0.96665055	0.98598355	0.98791689	0.98985016	0.9917835	0.9917835	0.9917835	0.9917835	0.9917835	0.9917835
SC18	0.99825174	0.99883449	0.99883449	0.99883449	0.99883449	0.99883449	0.99883449	0.99883449	0.99883449	0.99883449
SC19	0.79700273	0.86239785	0.8910082	0.90735698	0.91553134	0.92098093	0.9237057	0.92643052	0.92779291	0.93188012
SC20	0.98601401	0.993007	0.993007	0.993007	0.993007	0.993007	0.993007	0.993007	0.993007	0.993007

Table A.17: Hit-rate with maxRecom=unlimited

Bibliography

[1] Serge Abiteboul and Victor Vianu. Regular path queries with constraints. In *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, PODS '97, pages 122–133, New York, NY, USA, 1997. ACM. 89

[2] Sisay Fissaha Adafre and Maarten de Rijke. Discovering missing links in wikipedia. In *Proceedings of the 3rd international workshop on Link discovery*, LinkKDD '05, pages 90–97, New York, NY, USA, 2005. ACM. 80, 124

[3] G. Adomavicius and A. Tuzhilin. Towards 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, 2005. 76, 91

[4] Faisal Alkhateeb, Jean-François Baget, and Jérôme Euzenat. Extending sparql with regular expression patterns (for querying rdf). *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(2), 2011. 89

[5] Marcelo Arenas, Sebastián Conca, and Jorge Pérez. Counting beyond a yottabyte, or how sparql 1.1 property paths will prevent adoption of the standard. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 629–638, New York, NY, USA, 2012. ACM. 89

[6] Aaron Bangor, Philip T. Kortum, and James T. Miller. An Empirical Evaluation of the System Usability Scale. *International Journal of Human-Computer Interaction*, 24(6):574–594, 2008. 58

[7] Benjamin Huynh-Kim Bang, Eric Dané, and Monique Grandbastien. Merging semantic and participative approaches for organising teachers' documents. In *Proceedings of ED-Media 08 ED-MEDIA 08 - World Conference on Educational Multimedia, Hypermedia & Telecommunications*, pages p. 4959–4966, Vienna France, 07 2008. 33

[8] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001. 12

[9] Chris Bizer, Richard Cyganiak, and Tom Heath. How to publish linked data on the web. *Retrieved June*, 20:2008, 2007. 75

[10] Chris Bizer, Anja Jentzsch, and Richard Cyganiak. State of the lod cloud, October 2010. 32

[11] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data-the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22, 2009. 75

[12] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154 – 165, 2009. 12, 32, 63, 73, 83

[13] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008. 12, 79

[14] Matteo Bonifacio, Paolo Bouquet, Gianluca Mameli, and Michele Nori. Peer-mediated distributed knowledge management. In *Proc. of the AAAI Spring Symposium on Agent Mediated Knowledge Management (AMKM-03*, 2003. 27

[15] Matteo Bonifacio and Roberta Cuel. Knowledge nodes: the building blocks of a distributed approach to knowledge management. *J. of Universal Computer Science*, 8, 2002. 26, 27

[16] J.S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998. 91

[17] Jeen Broekstra, Arjohn Kampman, and Frank van Harmelen. Sesame: A generic architecture for storing and querying rdf and rdf schema. In Ian Horrocks and James A. Hendler, editors, *International Semantic Web Conference*, volume 2342 of *Lecture Notes in Computer Science*, pages 54–68. Springer, June 9-12 2002. 42, 45

[18] J. Brooke. SUS: A quick and dirty usability scale. In P. W. Jordan, B. Weerdmeester, A. Thomas, and I. L. Mclelland, editors, *Usability evaluation in industry*. Taylor and Francis, London, 1996. 58

[19] Michel Buffa, Guillaume ErÈtÈo, Catherine Faron-Zucker, Fabien Gandon, and Peter Sander. SweetWiki: A Semantic Wiki. *Journal of Web Semantics, special issue on Web 2.0 and the Semantic Web*, 6(1), february 2008. 30

[20] Michel Buffa, Fabien Gandon, Guillaume Ereteo, Peter Sander, and Catherine Faron. Sweetwiki: A semantic wiki. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(1):84–97, 2008. 28

[21] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004. 100

[22] Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito, and Markus Zanker. Linked open data to support content-based recommender systems. In *8th International Conference on Semantic Systems (I-SEMANTICS 2012)*, ICP. ACM Press, 2012. 79, 123

[23] Alicia Diaz, Guillermo Baldo, and Gérôme Canals. Co-protégé: Collaborative ontology building with divergences. In *Database and Expert Systems Applications, 2006. DEXA'06. 17th International Workshop on*, pages 156–160. IEEE, 2006. 27

[24] Alicia Díaz, Guillermo Baldo, and Gerome Canals. A framework for collaborative knowledge sharing with divergence. *IADIS Int. Journal on WWW/Internet*, 5(2):86–99, November 2007. 26

[25] Douglas Engelbart and Harvey Lehtman. Working together. *Byte*, 13(13):245–252, 1988. 12

[26] Sergio Firmenich, Marco Winckler, Gustavo Rossi, and Silvia E. Gordillo. A crowdsourced approach for concern-sensitive integration of information across the web. *J. Web Eng.*, 10(4):289–315, 2011. 122

[27] Sergio Firmenich, Marco Winckler, Gustavo Rossi, and Silvia E. Gordillo. A framework for concern-sensitive, client-side adaptation. In Sören Auer, Oscar Díaz, and George A. Papadopoulos, editors, *ICWE*, volume 6757 of *Lecture Notes in Computer Science*, pages 198–213. Springer, 2011. 122

[28] Daniel M Fleder and Kartik Hosanagar. Recommender systems and their impact on sales diversity. In *Proceedings of the 8th ACM conference on Electronic commerce*, pages 192–199. ACM, 2007. 100

[29] Dragan Gaaevic, Dragan Djuric, Vladan Devedzic, and Bran Selic. *Model Driven Architecture and Ontology Development*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. 40

[30] John H Gennari, Mark A Musen, Ray W Fergerson, William E Grosso, Monica Crubézy, Henrik Eriksson, Natalya F Noy, and Samson W Tu. The evolution of protégé: an environment for knowledge-based systems development. *International Journal of Human-computer studies*, 58(1):89–123, 2003. 27

[31] Thomas Gruber. Ontology of folksonomy: A mash-up of apples and oranges. *Int'l Journal on Semantic Web Information Systems*, 3(2):1–7, 2007. 22, 33, 43, 44, 122

[32] Tom Gruber. Collective knowledge systems: Where the social web meets the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(1):4 – 13, 2008. <ce:title>Semantic Web and Web 2.0</ce:title>. 11

[33] Oktie Hassanzadeh and Mariano P Consens. Linked movie data base. In *LDOW*, 2009. 78

[34] Paul Heymann and Hector Garcia-Molina. Collaborative creation of communal hierarchical taxonomies in social tagging systems. 2006. 30

[35] Raphael Hoffmann, Saleema Amershi, Kayur Patel, Fei Wu, James Fogarty, and Daniel S. Weld. Amplifying community content creation with mixed initiative information extraction. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, pages 1849–1858, New York, NY, USA, 2009. ACM. 81, 124

[36] Gina Hope, Taehyung (George) Wang, and Shan Barkataki. Convergence of web 2.0 and semantic web: A semantic tagging and searching system for creating and searching blogs. In *Proceedings of the International Conference on Semantic Computing*, pages 201–208, Washington, DC, USA, 2007. IEEE Computer Society. 21

[37] P. Jaccard. Nouvelles recherches sur la distribution florale. *Bulletin de la Sociète Vaudense des Sciences Naturelles*, 44:223–270, 1908. 93

[38] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems: An Introduction*. Cambridge University Press, 1 edition, September 2010. 64, 76

[39] Robert Jäschke, Andreas Hotho, Christoph Schmitz, Bernhard Ganter, and Gerd Stumme. Discovering shared conceptualizations in folksonomies. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(1):38–53, 2008. 31

[40] José Kahan and Marja-Ritta Koivunen. Annotea: an open rdf infrastructure for shared web annotations. In *Proceedings of the 10th international conference on World Wide Web*, WWW '01, pages 623–632, New York, NY, USA, 2001. ACM. 13, 33

[41] Hak L. Kim, Simon Scerri, John G. Breslin, Stefan Decker, Hong G. Kim, Jong-Ro Gu, and Yeon-Gun Dong. The State of the Art in Tag Ontologies: A Semantic Model for Tagging and Folksonomies. In *International Conference on Dublin Core and Metadata Applications, DC-2008–Berlin Proceedings*, pages 128–137, September 2008. 33, 122

[42] Hak-Lae Kim, John Breslin, Sung-Kwon Yang, and Hong-Gee Kim. Social semantic cloud of tag: Semantic model for social tagging. In Ngoc Nguyen, Geun Jo, Robert Howlett, and Lakhmi Jain, editors, *Agent and Multi-Agent Systems: Technologies and Applications*, volume 4953 of *Lecture Notes in Computer Science*, pages 83–92. Springer Berlin / Heidelberg, 2008. 33, 44, 122

[43] Markus Krötzsch, Denny Vrandecic, Max Völkel, Heiko Haller, and Rudi Studer. Semantic wikipedia. *Journal of Web Semantic*, 5(4):251–261, 2007. 28, 45

[44] Markus Krötzsch, Denny Vrandečić, Max Völkel, Heiko Haller, and Rudi Studer. Semantic wikipedia. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(4):251–261, 2007. 30

[45] Paul Lamere. Social tagging and music information retrieval. *Journal of New Music Research*, 37(2):101–114, 2008. 30

[46] T. K. Landauer and D. W. Nachbar. Selection from alphabetic and numeric menu trees using a touch screen: breadth, depth, and width. *SIGCHI Bull.*, 16(4):73–78, April 1985. 85, 91

[47] Thomas K Landauer and DW Nachbar. Selection from alphabetic and numeric menu trees using a touch screen: breadth, depth, and width. *ACM SIGCHI Bulletin*, 16(4):73–78, 1985. 91

[48] Kevin Larson and Mary Czerwinski. Web page design: implications of memory, structure and scent for information retrieval. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '98, pages 25–32, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co. 85, 91

[49] Heeseok Lee and Byounggu Choi. Knowledge management enablers, processes, and organizational performance: An integrative view and empirical examination. *Journal of Management Information Systems*, 20(1):179–228, 2003. 26

[50] Bo Leuf and Ward Cunningham. *The Wiki Way: Quick Collaboration on the Web*. Addison-Wesley Professional, April 2001. 68

[51] Freddy Limpens. Multi-points of view enrichment of folksonomies. Phd, Université Nice - Sophia Antipolis, October 2010. 33

[52] G.D. Linden, B. Smith, and J. York. Amazon.com recommendations. item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003. 75

[53] W. Lu, Y. Shen, S. Chen, and B.C. Ooi. Efficient processing of k nearest neighbor joins using mapreduce. *Proceedings of the VLDB Endowment*, 5(10):1016–1027, 2012. 93

[54] Tariq Mahmood and Francesco Ricci. Improving recommender systems with adaptive conversational strategies. In *Proceedings of the 20th ACM conference on Hypertext and hypermedia*, pages 73–82. ACM, 2009. 75

[55] Yogesh Malhotra. Why knowledge management systems fail? enablers and constraints of knowledge management in human enterprises. In C.W. Holsapple, editor, *Handbook on Knowledge Management*, volume 1 of *International Handbook on Information Systems*. Springer, Heidelberg, 2002. 27

[56] m.c. schraefel, D.A. Smith, A. Russell, and M.L. Wilson. Semantic web meets web 2.0 (and vice versa): The value of the mundane for the semantic web. Technical report, PsycPrints (United Kingdom), 2006. 44

[57] Brian McBride. Jena: A semantic web toolkit. *IEEE Internet Computing*, 6(6):55–59, 2002. 45

[58] Catherine McLoughlin and Mark JW Lee. Social software and participatory learning: Pedagogical choices with technology affordances in the web 2.0 era. In *ICT: Providing choices for learners and learning. Proceedings ascilite Singapore 2007*, pages 664–675, 2007. 11

[59] Liang Meiyu, Du Junping, Jia Yingmin, and Sun Zengqi. Image semantic description and automatic semantic annotation. In *Control Automation and Systems (ICCAS), 2010 International Conference on*, IEEE Xplore, pages 1192–1195. IEEE Xplore, 2010. 127

[60] Roberto Mirizzi, Tommaso Di Noia, Azzurra Ragone, Vito Claudio Ostuni, and Eugenio Di Sciascio. Movie recommendation with dbpedia. In *IIR*, pages 101–112. Citeseer, 2012. 78, 79, 123

[61] Alexandre Monnin, Freddy Limpens, Fabien Gandon, and David Laniado. Speech acts meet tagging: Nicetag ontology. In *Proceedings of the 6th International Conference on Semantic Systems*, I-SEMANTICS '10, pages 31:1–31:10, New York, NY, USA, 2010. ACM. 33

[62] Saikat Mukherjee, I. V. Ramakrishnan, and Michael Kifer. Semantic bookmarking for non-visual web access. In *Proceedings of the 6th international ACM SIGACCESS conference on Computers and accessibility*, Assets '04, pages 185–192, New York, NY, USA, 2004. ACM. 13, 33

[63] Richard Newman, Danny Ayers, and Seth Russell. Tag ontology, December 2005. 33, 44, 122

[64] Ikujiro Nonaka and Hirotaka Takeuchi. *The Knowledge - Creating Company: How Japanese Companies Create the Dynamics of Innovation.* Oxford University Press, 1995. 21, 26

[65] Bernardo Pereira Nunes, Stefan Dietze, Marco Antonio Casanova, Ricardo Kawase, Besnik Fetahu, and Wolfgang Nejdl. Combining a co-occurrence-based and a semantic measure for entity linking. In *The Semantic Web: Semantics and Big Data*, pages 548–562. Springer, 2013. 77, 124

[66] Eyal Oren. Semperwiki: a semantic personal wiki. In *Semantic Desktop Workshop*, 2005. 30

[67] Eyal Oren, Max V˜lkel, John G. Breslin, and Stefan Decker. Semantic wikis for personal knowledge management. In StÈphane Bressan, Josef K¸ng, and Roland Wagner, editors, *DEXA*, volume 4080 of *Lecture Notes in Computer Science*, pages 509–518. Springer, 2006. 30

[68] Gérald Oster, Pascal Urso, Pascal Molli, and Abdessamad Imine. Data consistency for p2p collaborative editing. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 259–268. ACM, 2006. 30, 37, 39

[69] Malcolm Otter and Hilary Johnson. Lost in hyperspace: metrics and mental models. *Interacting with computers*, 13(1):1–40, 2000. 91

[70] Derry O'Sullivan, Barry Smyth, David C Wilson, Kieran Mcdonald, and Alan Smeaton. Improving the quality of the personalized electronic program guide. *User Modeling and User-Adapted Interaction*, 14(1):5–36, 2004. 100

[71] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: bringing order to the web. 1999. 78

[72] A. Panchenko, S. Adeykin, A. Romanov, and P. Romanov. Extraction of semantic relations between concepts with knn algorithms on wikipedia. In *Proceedings of Concept Discovery in Unstructured Data Workshop (CDUD) of International Conference On Formal Concept Analysis*, pages 78–88, 2012. 80, 123

[73] A. Passant and P. Laublet. Meaning Of A Tag: A collaborative approach to bridge the gap between tagging and Linked Data. *Proceedings of the WWW 2008 Workshop Linked Data on the Web (LDOW2008), Beijing, China, Apr*, 2008. 33, 122

[74] Alexandre Passant. Measuring semantic distance on linking data and using it for resources recommendations. In *AAAI Spring Symposium: Linked Data Meets Artificial Intelligence*, 2010. 77, 123

[75] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. Semantics and complexity of sparql. *ACM Trans. Database Syst.*, 34(3):16:1–16:45, September 2009. 63, 83, 88

[76] Anand Rajaraman and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2012. 77

[77] Paul Resnick and Hal R. Varian. Recommender systems. *Commun. ACM*, 40(3):56–58, March 1997. 75

[78] S. Schaffert. Ikewiki: A semantic wiki for collaborative knowledge management. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2006. WETICE '06. 15th IEEE International Workshops on*, pages 388–396, 2006. 30, 68

[79] Sebastian Schaffert, François Bry, Joachim Baumeister, and Malte Kiesel. Semantic wikis. *software, IEEE*, 25(4):8–11, 2008. 28

[80] Shilad Sen, Shyong K. Lam, Al Mamunur Rashid, Dan Cosley, Dan Frankowski, Jeremy Osterhouse, F. Maxwell Harper, and John Riedl. Tagging, communities, vocabulary, evolution. In *CSCW '06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 181–190, New York, NY, USA, 2006. ACM. 52

[81] Katharina Siorpaes and Martin Hepp. Ontogame: Weaving the semantic web by online games. In Sean Bechhofer, Manfred Hauswirth, Jörg

Hoffmann, and Manolis Koubarakis, editors, *The Semantic Web: Research and Applications, 5th European Semantic Web Conference, ESWC 2008*, volume 5021 of *Lecture Notes in Computer Science*, pages 751–766. Springer, June 1-5 2008. 33

[82] Katharina Siorpaes and Elena Paslaru Bontas Simperl. Human intelligence in the process of semantic content creation. *World Wide Web*, 13(1-2):33–59, 2010. 21

[83] Hala Skaf-Molli, Charbel Rahhal, and Pascal Molli. Peer-to-peer semantic wikis. In *Database and Expert Systems Applications*, pages 196–213. Springer Berlin Heidelberg, 2009. 30, 36, 39, 40, 42

[84] Gerry Stahl, editor. *Group cognition: Computer support for building collaborative knowledge*. Cambridge, MA: MIT Press, 2006. 13, 17, 21, 25, 36

[85] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 697–706, New York, NY, USA, 2007. ACM. 12, 71, 86, 88

[86] Omer Sunercan and Aysenur Birturk. Wikipedia missing link discovery: A comparative study. In *AAAI Spring Symposium: Linked Data Meets Artificial Intelligence*. AAAI, 2010. 81, 124

[87] York Sure, Michael Erdmann, Jürgen Angele, Steffen Staab, Rudi Studer, and Dirk Wenke. *OntoEdit: Collaborative ontology development for the semantic web*. Springer, 2002. 27

[88] Carlo Torniai, Jelena Jovanovic, Dragan Gasevic, Scott Bateman, and Marek Hatala. E-learning meets the social semantic web. In *Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on*, pages 389–393. IEEE, 2008. 13

[89] D. Torres, P. Molli, H. Skaf-Molli, and A. Diaz. From dbpedia to wikipedia: Filling the gap by discovering wikipedia conventions. In *2012 IEEE/WIC/ACM International Conference on Web Intelligence (WI'12)*, 2012. 64, 131

[90] D. Torres, H. Skaf-Molli, P. Molli, and A. Diaz. BlueFinder: Recommending Wikipedia Links Using DBpedia Properties. In *ACM Web Science Conference 2013 (WebSci 13)*, Paris, France, May 2013. 64

[91] Diego Torres, Alicia Diaz, Hala Skaf-Molli, and Pascal Molli. Personal Navigation in Semantic Wikis. In *International Workshop on Adaptation and Personalization for Web 2.0 - AP-WEB 2.0 2009*, volume 485 of *CEUR Workshop Proceedings*, pages 148–151, Trento, Italie, June 2009. CEUR-WS.org. in connection with The First and Seventeenth International Conference on User Modeling, Adaptation and Personalization - UMAP'09. 22

[92] Diego Torres, Alicia Diaz, Hala Skaf-Molli, and Pascal Molli. Semdrops: A social semantic tagging approach for emerging semantic data. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, volume 1, pages 340–347. IEEE, 2011. 22

[93] Diego Torres, Pascal Molli, Hala Skaf-Molli, and Alicia Díaz. Improving Wikipedia with DBpedia. In Alain Mille, Fabien L. Gandon, Jacques Misselis, Michael Rabinovich, and Steffen Staab, editors, *WWW (Companion Volume)*, pages 1107–1112. ACM, 2012. 63, 64, 83

[94] Diego Torres, Hala Skaf-Molli, Alicia Dias, and Pascal Molli. Personal and shared knowledge building in p2p semantic wikis. In *6th European Semantic Web Conference (ESWC 2009)*, 2008. 22

[95] Diego Torres, Hala Skaf-Molli, Alicia Diaz, and Pascal Molli. Supporting personal semantic annotations in p2p? semantic wikis. In *20th International Conference on Database and Expert Systems Applications- DEXA 2009*, volume 5690 of *Lecture Notes in Computer Science*, Linz, Austria, August 2009. Springer. 22

[96] Thomas Vander Wal. Understanding folksonomy: Tagging that works. *Retrieved January*, 29:2008, 2006. 31

[97] Denny Vrandečić. Wikidata: a new platform for collaborative data collection. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 1063–1064, New York, NY, USA, 2012. ACM. 124

[98] Y. Wang, H. Wang, H. Zhu, and Y. Yu. Exploit semantic information for category annotation recommendation in wikipedia. *Natural Language Processing and Information Systems*, pages 48–60, 2007. 79, 123

[99] Martin Wattenberg, Fernanda B Viégas, and Katherine Hollenbach. Visualizing activity on wikipedia with chromograms. In *Human-Computer Interaction–INTERACT 2007*, pages 272–287. Springer, 2007. 72

[100] StÈphane Weiss, Pascal Urso, and Pascal Molli. Wooki: a p2p wiki-based collaborative writing tool. In *Web Information Systems Engineering*, Nancy, France, December 2007. Springer. 40

[101] Valentin Zacharias and Simone Braun. Soboleo – social bookmarking and lighweight engineering of ontologies. In Natalya Fridman Noy, Harith Alani, Gerd Stumme, Peter Mika, York Sure, and Denny Vrandecic, editors, *Proceedings of the Workshop on Social and Collaborative Construction of Structured Knowledge (CKC 2007) at the 16th International World Wide Web Conference (WWW2007)*, volume 273 of *CEUR Workshop Proceedings*. CEUR-WS.org, May 8 2007. 13, 33

[102] M.L. Zhang and Z.H. Zhou. A k-nearest neighbor based algorithm for multi-label classification. In *Granular Computing, 2005 IEEE International Conference on*, volume 2, pages 718–721. IEEE, 2005. 92

L'UNAM
l'université nantes angers le mans
PÔLE DE RECHERCHE ET D'ENSEIGNEMENT SUPÉRIEUR

UNIVERSITÉ DE NANTES

Thèse de Doctorat

Diego TORRES

Co-Evolution between Social and Semantic Web

Résumé

Le Web social et le Web sémantique ont eu un impact sur la façon dont la création de connaissances est effectuée sur le Web. Le Web social favorise la participation des utilisateurs à créer et modifier des contenus et des connaissances sur le Web. La prolifération de contenu et la nécessité d'une gestion automatisé de l'information a déclenché l'émergence du Web sémantique.
Actuellement, le Web social et le Web sémantique coexistent et partagent un thème commun : une meilleure gestion de la connaissance. Cependant, la plupart des informations sur le Web social ne fait pas partie du Web sémantique et l'information du Web sémantique n'est pas utilisée pour améliorer le Web social.
Cette thèse présente une approche innovante pour stimuler la co-évolution entre le Web sémantique et le Web social : les utilisateurs et les ordinateurs qui travaillent ensemble afin d'obtenir des avantages mutuels. Nous affirmons que la co-évolution du Web social et du Web sémantique permettra, d'une part d'améliorer la production de l'information sémantique au Web sémantique, et d'une autre part d'améliorer la production de connaissances sur le Web social.

Mots clés

Web social, Web sémantique, construction de connaissances collaborative, Système de recommandation collaborative, DBpedia, Wikipedia, l'étiquetage social.

Abstract

Social and Semantic Web has impacted in the manner of knowledge building is fulfill in the Web. The Social Web promoted the participation of users to create and edit Web content and knowledge. The content proliferation and the need to have a better machine management of such information trigger the Semantic Web. Currently, the Social and the Semantic Web are living together and they share a same topic: a better management of knowledge. However, most of the Social Web information is not part of the Semantic Web, and Semantic Web information is not used to improve the Social Web. This thesis introduced an innovative approach to stimulate a co- evolution between the Semantic and Social Web: social and machine forces work together in order to have mutual benefits. We claim that having a co-evolution between Social and Semantic Web will improve the generation of semantic data and a knowledge production improvement in the Social Web.

Key Words

Social Web, Semantic Web, Collaborative Knowledge Building Process, Collaborative Recommender System, DBpedia, Wikipedia, Social Tagging