

A thesis presented for the degree of
Doctor of Philosophy in Computer Science

PowerAqua: Open Question Answering on the Semantic Web

Vanessa Lopez

Semantic Web and Knowledge Services
The Knowledge Media Institute
The Open University
England

December 2011

PowerAqua: Open Question Answering on the Semantic Web

Vanessa Lopez

A thesis presented for the degree of
Doctor of Philosophy in Computer Science



Semantic Web and Knowledge Services
The Knowledge Media Institute
The Open University
England

December 2011

To my parents,
Argimiro & Mary

Abstract

With the rapid growth of semantic information in the Web, the processes of searching and querying these very large amounts of heterogeneous content have become increasingly challenging. This research tackles the problem of supporting users in querying and exploring information across multiple and heterogeneous Semantic Web (SW) sources.

A review of literature on ontology-based Question Answering reveals the limitations of existing technology. Our approach is based on providing a natural language Question Answering interface for the SW, PowerAqua. The realization of PowerAqua represents a considerable advance with respect to other systems, which restrict their scope to an ontology-specific or homogeneous fraction of the publicly available SW content. To our knowledge, PowerAqua is the only system that is able to take advantage of the semantic data available on the Web to interpret and answer user queries posed in natural language. In particular, PowerAqua is uniquely able to answer queries by combining and aggregating information, which can be distributed across heterogeneous semantic resources.

Here, we provide a complete overview of our work on PowerAqua, including: the research challenges it addresses; its architecture; the techniques we have realised to map queries to semantic data, to integrate partial answers drawn from different semantic resources and to rank alternative answers; and the evaluation studies we have performed, to assess the performance of PowerAqua. We believe our experiences can be extrapolated to a variety of end-user applications that wish to open up to large scale and heterogeneous structured datasets, to be able to exploit effectively what possibly is the greatest wealth of data in the history of Artificial Intelligence.

Preface

Finding Answers on the Semantic Web: *What S said to W.*

The following dialog has been extracted from the article published in the fourth issues of the Nodalities Magazine published by Talis. Reference: Finding Answers on the Semantic Web. Mathieu d'Aquin and Vanessa Lopez. Nodalities Magazine September / October 2008 issue: http://www.talis.com/nodalities/pdf/nodalities_issue4.pdf.

Our story begins with two characters, which are neither turtle nor Greek hero, attending Tim-Berners Lee's keynote speech at the WWW 2008 conference. After the talk, they gather at the bar for a casual discussion about their favourite topic: the Web. Let's call them S and W, which, while these are obviously not their real names, suit them quite well as the main characters of our story.

W: You know, He is always talking about the Semantic Web, but to be honest, I still don't get it.

S: Really? The great giant graph, information available for machines to process, web-scale intelligent applications...

W: I understand the principle, the vision, the idea, but I still don't see how this sort of technology benefits the user. But as you are being so passionate about it, maybe you could try to convince me.

S: Of course I am being passionate about it: there is a great deal out there that can be used to build a new kind applications, something completely different from what have ever been seen before.

W: Yeah, of course, but this is all geek things for geeks. How about the real users? For example, how can this make a difference in the way people query the Web?

S: Oh... I see. And what if I tell you that it is possible to actually ask questions to the Semantic Web and obtain answers?

W: You mean like when querying Google?

S: Of course not! I'm talking about actually getting the answers to your questions thanks to Semantic resources like asking "What are the members of the rock band Nirvana?" and getting a list of names, not a list of documents containing the words "Nirvana" and "member".

W: This is nice, but I have seen it all before, question answering systems. They need to be fed with huge quantities of information, and are still limited to the one domain of application they are meant to work on. This is not even getting close to Google.

S: This is actually the beauty of it. You don't have to feed the system with knowledge acquired the hard way anymore, it can just use all the semantic information available on the Web thanks to Semantic Web technologies. In other terms, it just has to get the list of members of Nirvana from an RDF dataset somewhere on the Web.

W: But this would require to find this dataset, and to extract the answer from it, all that at run-time. How could it be done?

S: That's my friends, is where our story begins ...

Acknowledgements

The long path to my part-time PhD now is reaching its end. It has been a very rewarding and enjoyable path, and I can honestly say, I do believe KMi is one of the best places to work and do research. An unexpectedly pleasant “black hole”, where people initially come for a short time and, then they cannot escape the gravitational attraction of this place, and if they do, they keep coming back. What makes this place so special are its vibe and the people that make up that vibe.

During my PhD I have always been surrounded by very inspiring people that have helped me to discover the thrill of doing research, and made me decide that after all, doing a PhD may not be such an insane idea. In this learning path I was not alone, on the contrary many friends crossed my path, helping me walk (and even dance!) my way to the end of it, making up the whole experience, and acknowledging them properly is a very challenging task.

It all started with a short visit from Oscar, Angel and Miguel when I was living in Leiden, where Oscar Corcho told me about a job position in a cool research lab, named KMi. Thus, the original reason why I came here is my collaboration in Asun's group, as an undergraduate student at the UPM, where they introduced me to the mysterious world of ontologies, guiding my future path.

I truly had a first-class team of people supervising my work: Enrico Motta, Victoria Uren and Marta Sabou, each one is a source of wisdom, both humbling and awesome to watch. Thank you for your warmth, you have been my rock when I truly needed. Although Enrico always says I am his most unmanageable student, behind my Spanish temperament I have learnt a lot from our discussions. I am truly privileged to have him as supervisor. He made my work an amusing experience with his vision to transform difficulties into rewarding opportunities, his good sense of humour and self-confidence to sort out “unbelievable” rules, and a unique combination of honest and fair winning attitude.

I am grateful to my examiners, Fabio Ciravegna, Anne De Roeck, and the VIVA chair, Stefan Ruger, for the thoughtful and useful discussion, to Philip Cimiano and Christina for giving me the opportunity to co-organize the QALD workshop with them, and to many people within KMi. The extremely supportive administrative and technical teams: Ortenz, Jane, Aneta, Paul, Harriet, Peter, Damian, Lewis, Robbie, etc. A great team to work and travel the globe with: Andriy, Mathieu, Laurian, Tom, Carlo, Davide, Fouad, Nico, Silvio, Jorge, John, and in particular my other busy bee Miriam, for all those long fun hours working and traveling together. In fact, I can't think of a more diverse bunch of people than the ones I have met at the OU!. Many friends along the way made my walk meaningful at a personal level, all my "octopusses" friends, which became my MK family since the beginning ("Atisha" times): Dinar, George, Pejman, Liliana, Joan, Ana banana, Carlos, Miriam, Lili, and along the way: Anna Lisa, Sofia, Maca, Thomas, Alba, Koula, Jakub, Eva, Enrico, Carmen, Nieves, Dario, Ainhoa, Maria, Matthew, Vlad, and the "old" ones that keep in contact wherever I am: Lucia, Noe, Alicia, Ana, Lisa, Josevi. I had the chance to meet so many great people that I cannot list them all here. In fact, my knowledge about international food and celebrations throughout the globe can hardly be more elaborated.

Finally, to Philippe, which provided me relentless support, feeding me and being there for me while this dissertation was being written, and to my family which have always been unconditional and supportive of who I am, trusting me and giving me the self-confidence to follow my own choices.

The research reported in this thesis has been supported by a number of projects: the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), sponsored by the UK EPSRC under grant GR/N15764/01; the Open Knowledge project, sponsored by the European Commission (EC) as part of the Information Society Technologies (IST) programme under grant IST-2001-34038, the X-Media project, sponsored by the EC under grant IST-FP6-026978; and the Smart-products project sponsored by the EC under grant EC-231204.

Related Publications

PAPERS – CONFERENCE

- Fernandez, M., Zang, Z., Lopez, V., Uren, V., Motta, E. (2011) **Ontology Augmentation: Towards a Healthy Meal Planning**. In Proc. of the Knowledge Capture Conference (K-CAP 2011), Banff, Canada.
- Lopez, V., Nikolov, A., Sabou, M, Uren, V., Motta, E., and d'Aquin, M. (2010) **Scaling up Question-Answering to Linked Data**. In Proc. of the Knowledge Engineering and Knowledge Management by the Masses (EKAW 2010), Lisbon, Portugal.
- Lopez, V., Nikolov, A., Fernandez, M., Sabou, M, Uren, V. and Motta, E. (2009) **Merging and Ranking answers in the Semantic Web: The Wisdom of Crowds**. In Proc. of the Asian Semantic Web Conference (ASWC 2009), Shanghai, China. Best paper award.
- Lopez, V., Sabou, M., Uren, V. and Motta, E. (2009). **Cross-Ontology Question Answering on the Semantic Web – an initial evaluation**. In Proc. of the Knowledge Capture Conference (K-CAP 2009), California, USA.
- Fernandez, M., Lopez, V., Sabou, M., Uren, V., Vallet, D., Motta, E., and Castells, P. (2008). **Semantic Search meets the Web**. In the International conference on Semantic Computing (ICSC 2008), California, USA.
- Lopez, V., Sabou, M. and Motta, E. (2006) **PowerMap: Mapping the Real Semantic Web on the Fly**. In Proc. of the International Semantic Web Conference (ISWC 2006), Georgia, Atlanta.
- Lopez, V., Motta, E. and Uren, V. (2006) **PowerAqua: Fishing the Semantic Web**. In Proc. of the European Semantic Web Conference (ESWC 2006), Montenegro.

- Sabou, M., Lopez, V. and Motta, E. (2006) **Ontology Selection on the Real Semantic Web: How to Cover the Queens Birthday Dinner?**. In Proc. of the International Conference on Knowledge Engineering and Knowledge Management (EKAW 2006), Podebrady, Czech Republic.
- Lei, Y., Sabou, M., Lopez, V., Zhu, J., Uren, V. and Motta, E. (2006) **An Infrastructure for Acquiring High Quality Semantic Metadata**. In Proc. of the 3rd European Semantic Web Conference (ESWC 2006), Budva, Montenegro.
- Lopez, V., Motta, E. and Pasin, M. (2005) **AquaLog: An Ontology portable Question Answering Interface for the Semantic Web**. In Proc. of the European Semantic Web Conference (ESWC 2005), Heraklion, Crete.
- Lopez, V. and Motta, E. (2004) **Ontology Driven question answering in AquaLog**, In Proc. of the International Conference on Applications of Natural Language to Information Systems (NLDB 2004), Manchester, UK.

PAPERS - JOURNALS

- Lopez, V., Uren, V., Sabou, M. and Motta, E. (2011) **Is Question Answering fit for the Semantic Web? A Survey**. In the Semantic Web – Interoperability, Usability, Applicability, 2(2).
- Lopez, V., Fernandez, M., Motta, E. and Stieler, N. (2011) **PowerAqua: supporting users in querying and exploring the Semantic Web**. In the Semantic Web – Interoperability, Usability, Applicability. To appear.
- Fernandez, M., Cantandor, I., Lopez, V., Vallet, D., Castells, P. and Motta, E. (2010) **Semantically enhanced Information Retrieval: an ontology-based approach**. In the journal of Web Semantics: Science, Services and Agents on the WWW, Special issues on Semantic Search. In Press.

- Uren, V., Sabou, M., Motta, E., Fernandez, M., Lopez, V. and Lei, Y. (2010) **Reflections on five years of evaluating semantic search systems**. In the International Journal of Metadata, Semantics and Ontologies, 5(2).
- D'Aquin, M., Motta, E., Sabou, M., Angeletou, S., Gridinoc, L., Lopez, V. and Guidi, D. (2008) **Towards a New Generation of Semantic Web Applications**. In the special issue of IEEE Intelligent Systems on the Semantic Web, 23(3).
- Uren, V., Lei, Y., Lopez, V., Liu, H., Motta, E. and Giordanino, M. (2007) **The usability of semantic search tools: a review**. Knowledge Engineering Review, 22(4): 361-377.
- Lopez, V., Uren, V., Motta, E. and Pasin, M. (2007) **AquaLog: An ontology-driven question answering system for organizational semantic intranets**. Journal of Web Semantics, 5(2): 72-105, Elsevier.
- Lei, Y., Lopez, V., Motta, E. and Uren, V. (2007) **An infrastructure for Semantic Web Portals** (2007) In a special issue of the Journal of Web Engineering, 6(4): 283-308.

PAPERS – WORKSHOP

- Fernandez, M., Lopez, V., Vallet, D., Castells, P., Motta, E., Sabou, M. and Uren, V. (2009) **Using TREC for cross-comparison between classic IR and ontology-based search models at a Web scale**. In the Semantic Search 2009 Workshop at the International World Wide Web Conference, Madrid, Spain.
- D'Aquin, M., Sabou, M., Motta, E., Angeletou, S., Gridinoc, L., Lopez, V. and Zablith, F. (2008) **What can be done with the Semantic Web? An Overview of Watson-based Applications**. In the 5th Workshop on Semantic Web Applications and Perspectives, SWAP 2008, Rome, Italy.
- Gracia, J., Lopez, V., d'Aquin, M., Sabou, M., Motta, E. and Mena, E. (2007) **Solving Semantic Ambiguity to Improve Semantic Web based Ontology Matching**. In the

Workshop on Ontology Matching at the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference, Korea.

- Sabou, M., Lopez, V., Motta, E. and Uren, V. (2006) **Ontology Selection: Ontology Evaluation on the Real Semantic Web**. In the Workshop on Evaluation of Ontologies for the Web (EON 2006) at 15th International World Wide Web Conference, Edinburgh.
- Lei, Y., Lopez, V. and Motta, E. (2006) **An Infrastructure for Building Semantic Web Portals**. In the International Workshop on Web Information System Modeling (WISM 2006), Luxembourg.
- Lopez, V. and Motta, E. (2005) **PowerAqua: An Ontology Question Answering System for the Semantic Web**, In the Workshop on Ontologias y Web Semantica 2005, held in conjunction with CAEPIA 2005, Spain.

DEMO - POSTERS

- D'Aquin M., Lopez V. and Motta, E. (2008). **FABiT – Finding Answers in a Billion Triples**. In the billion triple challenge hold in the International Semantic Web Conference (ISWC), Karlsruhe, Germany.
- Lopez, V., Fernandez, M., Motta, E., Sabou, M. and Uren, V. (2007) **Question Answering on the Real Semantic Web**. International Semantic Web Conference (ISWC 2007)- Demo session, Korea.
- Lopez, V., Motta, E. and Uren, V. (2006) **AquaLog: An ontology-driven Question Answering System to interface the Semantic Web**, In the HLT-NAACL 2006 - Demo session, New York
- Lopez, V. (2004) **AquaLog technology demonstration**. In the International Conference on Knowledge Engineering and Knowledge Management (EKAW 2004) – Demo session, United Kingdom.

Contents

ABSTRACT.....	V
PREFACE	VI
ACKNOWLEDGEMENTS	VIII
RELATED PUBLICATIONS.....	X
CONTENTS	XIV
PART I: GENERAL OVERVIEW, MOTIVATIONS AND STATE OF THE ART	20
CHAPTER 1 INTRODUCTION AND MOTIVATIONS.....	21
1.1 INTRODUCTION: THE SEMANTIC WEB VISION	21
1.2 MOTIVATION: A NEW GENERATION OF TOOLS FOR THE SEMANTIC WEB	23
1.3 CHALLENGE: QUERYING AND EXPLORING THE SW CONTENT	25
1.4 THEME AND CONTRIBUTIONS OF THIS THESIS	27
1.5 RESEARCH APPROACH: THE MULTI ONTOLOGY CHALLENGE	30
1.6 THESIS ORGANIZATION	33
CHAPTER 2 LITERATURE REVIEW.....	34
2.1 INTRODUCTION AND OUTLINE	34
2.2 GOALS & DIMENSIONS OF QUESTION ANSWERING.....	36
2.3 RELATED WORK ON QUESTION ANSWERING.....	41
2.3.1 <i>NLIDB: Natural Language Interfaces to Databases</i>	41
2.3.2 <i>Open Domain Question Answering over documents</i>	44
2.3.3 <i>Developments on commercial open QA</i>	48
2.4 ONTOLOGY-BASED QUESTION ANSWERING.....	50
2.4.1 <i>The Pioneer: the AquaLog QA system</i>	50
2.4.2 <i>A review of ontology-based QA systems</i>	53
2.4.3 <i>The performance of ontology-based QA systems</i>	62
2.4.4 <i>The competence of ontology-based QA systems</i>	67
2.4.5 <i>Limitations of QA approaches in the context of the SW</i>	71

2.5 RELATED WORK ON USER-FRIENDLY QUERY INTERFACES FOR THE SW	74
2.5.1 <i>Early global-view information systems</i>	75
2.5.2 <i>Evolution of Semantic Search on the Web of Data</i>	76
2.5.3 <i>Large scale Semantic Search and Linked Data interfaces</i>	78
2.6 QA ON THE SW: ACHIEVEMENTS AND RESEARCH GAPS	81
2.6.1 <i>Sources for QA and their effect on scalability</i>	81
2.6.2 <i>Beyond the scope of closed-domain QA</i>	83
2.6.3 <i>Issues associated with performing QA in open and dynamic environments</i>	85
2.6.4 <i>Input and higher expressivity</i>	86
2.7 DIRECTIONS AHEAD: THE CONTRIBUTION OF THIS THESIS	87
CHAPTER 3 CHALLENGES AND REQUIREMENTS	90
3.1 INTRODUCTION: POWERAQUA RESEARCH CHALLENGES	90
3.2 TRADITIONAL INTRINSIC PROBLEMS IN QA: HETEROGENEITY	91
3.2.1 <i>State of the art on ontology mapping</i>	92
3.2.2 <i>State of the art on semantic similarity and WSD</i>	94
3.3 REQUIREMENTS FOR POWERAQUA IN THE CONTEXT OF SW	97
3.4 THE APPROACH OF POWERAQUA AT A GLANCE	99
3.5 SUMMING UP	101
CHAPTER 4 ARCHITECTURE OVERVIEW	103
4.1 INTRODUCTION	103
4.2 ARCHITECTURE: AN ILLUSTRATIVE EXAMPLE	103
4.3 A TRIPLE-BASED APPROACH	112
4.4 UNDERLYING INFRASTRUCTURE	114
4.4.1 <i>PowerMap ontology plugin mechanism</i>	114
4.4.2 <i>Indexing mechanism in PowerMap</i>	115
4.4.3 <i>The Watson Semantic Web gateway</i>	117
PART II: THE POWERAQUA COMPONENTS	118
IN DETAIL	118
CHAPTER 5 LINGUISTIC ANALYSIS (STEP 1)	119

5.1 INTRODUCTION	119
5.2 REALIZATION OF THE LINGUISTIC COMPONENT	120
5.3 CLASSIFICATION OF QUESTIONS	122
5.3.1 <i>Linguistic triples for basic queries</i>	122
5.3.2 <i>Linguistic triples for basic queries with prepositional modifiers</i>	124
5.3.3 <i>Linguistic triples for combination of queries</i>	126
5.4 THE USE OF JAPE GRAMMARS: AN ILLUSTRATIVE EXAMPLE	127
5.5 DISCUSSION	129
5.5.1 <i>Question types and Query-Triple mapping</i>	129
5.5.2 <i>Ambiguity</i>	133
5.5.3 <i>Reasoning mechanism and services</i>	134
5.6 CONCLUSIONS	135
CHAPTER 6 ELEMENT MAPPING (STEP 2)	136
6.1 INTRODUCTION TO POWERMAP MAPPING COMPONENT	136
6.2 PHASE I: ONTOLOGY SELECTION AND DISCOVERY	138
6.3 PHASE II: SEMANTIC ANALYSIS AND FILTERING	140
6.3.1 <i>A WordNet-based algorithm for computing Semantic Similarity</i>	141
6.3.2 <i>Verifying the meaning and semantic validity of mappings</i>	143
6.3.3 <i>Experimental example</i>	145
6.4 POWERMAP EVALUATION OF SEMANTIC CAPABILITIES	147
6.4.1 <i>Evaluation task: Improving term anchoring</i>	147
6.4.2 <i>Applying PowerMap semantic filtering in the anchoring task</i>	150
6.4.3 <i>Analysis of results</i>	151
6.4.4 <i>Discussion and WordNet limitations</i>	151
6.5 SUMMING UP	153
CHAPTER 7 TRIPLE MAPPING (STEP 3)	155
7.1 INTRODUCTION	155
7.2 THE TRIPLE SIMILARITY SERVICE COMPONENT	157
7.2.1 <i>TSS algorithm</i>	157
7.2.2 <i>TSS Illustrative examples</i>	160

7.2.3 <i>The Relation Similarity Service</i>	165
7.3 EFFICIENCY OF THE TSS	170
7.4 KNOWN ISSUES	174
7.5 SUMMING UP	176
CHAPTER 8 MERGING AND RANKING (STEP 4)	178
8.1 INTRODUCTION	178
8.2 ILLUSTRATIVE EXAMPLE.....	179
8.3 MERGING ALGORITHM.....	182
8.3.1 <i>Merging scenarios</i>	182
8.3.2 <i>The co-reference algorithm</i>	185
8.4 RANKING ALGORITHM AND CRITERIA	186
8.4.1 <i>Ranking by semantic similarity</i>	188
8.4.2 <i>Ranking by confidence</i>	189
8.4.3 <i>Ranking by popularity</i>	191
8.4.4 <i>Ranking by combination</i>	191
8.5 INITIAL RESEARCH WORK ON USING TRUST FOR RANKING.....	192
8.6 SUMMING UP	194
PART III: EVALUATION, SCALABILITY, CONCLUSIONS AND OUTLOOK.....	196
CHAPTER 9 EVALUATION OF POWERAQUA WITH RESPECT TO QUESTION ANSWERING ON	
THE SW	197
9.1 INTRODUCTION	197
9.2 EVALUATION 1: DERIVING ANSWERS FROM MULTIPLE ONTOLOGIES	199
9.2.1 <i>Experimental Setup</i>	200
9.2.2 <i>List of queries and results</i>	201
9.2.3 <i>Analysis of Results</i>	205
9.2.4 <i>Conclusions</i>	211
9.3 EVALUATION 2: MERGING AND RANKING	213
9.3.1 <i>Evaluating the fusion algorithm</i>	215
9.3.2 <i>Evaluating the filtering performed by the merging algorithm</i>	216
9.3.3 <i>Evaluating the ranking algorithms</i>	217

9.3.4 Results	217
9.3.5 Discussion of the results.....	222
9.3.6 Conclusions	223
9.4 EVALUATION 3: USABILITY STUDY	225
9.4.1 Evaluation approach: the SEALS evaluation campaign.....	225
9.4.2 Results	227
9.4.3 Discussion and Conclusions	229
 CHAPTER 10 EVALUATION OF POWERAQUA WITH RESPECT TO QUERY EXPANSION ON THE WEB (IR).....	 233
10.1 MOTIVATING SCENARIO	233
10.2 POWERAQUA AS PART OF AN ADVANCED IR SYSTEM: OVERVIEW.	235
10.3 EXPERIMENTAL SETUP	238
10.4 ADAPTING THE TREC QUERIES AND LIST OF QUERY TOPICS.....	240
10.5 RESULTS	244
10.6 DISCUSSION	247
10.7 FEASIBILITY STUDY: INTEGRATION WITH YAHOO SEARCH ENGINE.....	249
10.7.1 Evaluation set up.....	250
10.7.2 Results	251
10.8 CONCLUSIONS	253
 CHAPTER 11 SCALING UP TO LINKED DATA: ISSUES, EXPERIMENTS AND LESSONS LEARNED	 255
11.1 MOTIVATIONS	255
11.2 CURRENT INTERFACES TO LINKED DATA AND LIMITATIONS	257
11.3 BEFORE AND AFTER LINKED DATA: A NEW DIMENSION IN QA	262
11.3.1 Scaling to highly populated and dense ontologies.....	262
11.3.2 Mapping query terms to large generic ontologies.....	263
11.3.3 Fusion across heterogeneous and decentralize ontologies.....	265
11.4 SOLUTIONS TO CHALLENGES AND LESSONS LEARNED.....	265
11.4.1 Large scale data: Shifting focus onto precision for mapping and fusion	265
11.4.2 Higher heterogeneity and duplicated terms: filtering heuristics based on quality and semantics	268

11.4.3 <i>Lack of semantics and incomplete data: light-weight reasoning</i>	269
11.5 EXPERIMENTS WITH DBPEDIA AND DISCUSSION	272
11.6 EVALUATING POWERAQUA’S RESPONSE TIME WHEN USING DIFFERENT SEMANTIC STORAGE PLATFORMS	276
11.6.1 <i>Using Virtuoso as a semantic storage platform</i>	276
11.6.2 <i>Using the Watson SW gateway</i>	277
11.7 SUMMING UP AND CONCLUSIONS	279
CHAPTER 12 CONCLUSIONS AND FUTURE DIRECTIONS	282
12.1 CONCLUSIONS AND CONTRIBUTIONS.....	282
12.1.1 <i>Contributions of semantic QA to the state of the art</i>	283
12.1.2 <i>Our proposal towards Semantic Question Answering</i>	284
12.1.3 <i>Evaluation</i>	286
12.1.4 <i>Scalability towards a Web environment</i>	288
12.2 FUTURE WORK AND EXTENSIONS	290
BIBLIOGRAPHY	293
A.1 EXAMPLE OF NL QUERIES BY TOPICS	301
B.1 POWERAQUA CONFIGURATION.....	302
B.2 POWERAQUA INDEXING MECHANISM	303

PART I: GENERAL OVERVIEW, MOTIVATIONS AND STATE OF THE ART

In part 1 the context, vision, ideas, research challenges and contributions behind this thesis are presented. We also discuss the related work, highlighting in particular the research gap in the current state of the art that this thesis addresses. Finally, we introduce the key design challenges facing the proposed research.

"In the last 30 years mankind has produced more information than in the previous 5000" (Information overload causes stress, 1997, Reuters Magazine)

Chapter 1 Introduction and Motivations

In this chapter, we present the research issues and motivations behind the development of PowerAqua. This work is part of a wider research programme which is described in the IEEE Journal of Intelligent Systems in 2008 (d'Aquin et al., 2008).

The vision and the ideas behind our two pioneering ontology-based applications: AquaLog and PowerAqua, as well as the key challenges facing the proposed research, were first presented and published in the main track of the International Conference on Applications of Natural Language to Information Systems (Lopez and Motta, 2004) and the European Semantic Web Conference (Lopez et al., 2006a), in the years 2004 and 2006 respectively.

1.1 Introduction: the Semantic Web vision

The enormous size of the Web is one of the keys to its success, but it can also make the task of searching for some particular information time-consuming and tedious for users. People can interpret the contents and the output of services on the Web, presented in formats such as HTML, but machines cannot understand these contents, which limits the effectiveness of current search engines. In particular, this is the case when a user's demand cannot be easily expressed by using a few keywords, or the response is unlikely to be available in one document but must be assembled by combining multiple documents (Hallet et al., 2007).

The development of a semantic layer on top of Web contents and services, the Semantic Web (Berners-Lee et al., 2001), has been recognized as the next step in the evolution of the World Wide Web as a distributed knowledge resource. The Semantic Web (SW) brings to the Web the idea of having data formally defined and linked, in a way that it can be used for effective information discovery, integration, reuse and sharing across various applications, and for service automation. The semantic extension of the Web consists of both semantic data and metadata, strictly speaking metadata explicitly describes the content of Web pages in a way that is machine-processable, according to (Finin et al., 2005) typically text documents are enriched by annotations in machine understandable mark-up, alternatively, SW content can be described in separate documents entirely

encoded in a RDF-based mark-up language that reference the content of conventional Web documents. At the same time, semantic data can be generated from external data sources and published without explicitly referring to Web documents. For instance, the goal of the Linked Open Data (LOD) initiative¹ (Bizer et al., 2009a) is to extend the Web by publishing open datasets describing things in the world through RDF triples and by setting RDF links between data items from different sources; so far it has resulted in an openly available Web of data comprising several billions of RDF triples from diverse domains: Wikipedia, government, entertainment, bio-informatics and publications.

In practice, the semantic knowledge extracted from Web resources actually corresponds to a knowledge base encoded using SW technology (Fazzinga and Lukasiewicz, 2010), as stated in (Finin et al., 2005): “While approaches like RDF and OWL are often called markup or metadata, they are typically used not as a markup but rather as stand-alone knowledge representation languages that are not directly tied to text”. Thus, in the content of this thesis we use the terms semantic data and metadata interchangeably.

Often the representation schema for this machine-understandable content on the SW is defined in *ontologies* (Gruber, 1993). Thus, ontologies play a crucial role on the SW by enabling knowledge sharing and exchange: ontologies are reused to characterize knowledge in a particular domain, to provide the conceptual infrastructure supporting semantic interoperability (Adams et al., 2000) and opening up opportunities for automated information processing (Doan et al., 2002). In practical terms, ontologies are commonly handled as hierarchies of concepts with attributes and relations among them, which establish a terminology to define semantic networks of interrelated concepts (i.e., a schema), formally describing domain-specific knowledge, that is often populated and

¹ <http://linkeddata.org/>

instantiated (with instances of the ontological concepts and their relations) in one or various Knowledge Bases (KBs)².

Ontologies are an established research topic in the knowledge representation field (Staab and Studer, 2004), a field which, in turn, has been studied in the artificial intelligence research area long before the emergence of WWW. Formal logic is the foundation of ontologies and the SW. The formal semantics and expressivity of logic statements are well understood and unambiguous. However, because of the SW's distributed nature, data will inevitably be associated with different ontologies and therefore ontologies themselves will introduce heterogeneity. Different ontologies may describe similar domains, but they may use different terminologies, while others may have overlapping domains: i.e., given two ontologies, the same entity can be given different names or simply be defined in different ways. As such, the SW is based on a Web of dynamically growing distributed ontologies and KBs. Semantic technologies that are able to bridge, exploit and benefit from these interconnected and distributed semantic data, contribute to the realization of the SW.

1.2 Motivation: A new generation of tools for the Semantic Web

As argued in (Motta and Sabou, 2006) the processes of building the required infrastructure to produce a large SW and the “smart” applications to exploit such semantic data (or at least, demonstrators) have to go hand in hand. Until recently, only a limited amount of semantic data was available, therefore, Motta and Sabou stated that:

“The early demonstrators produced in the past few years lack many of the key elements that will characterize ‘real’ SW applications that will operate in an open, large-scale, distributed and heterogeneous environment. Consequently they are more akin to traditional KB systems”.

² The distinction between ontologies and KBs is not clear-cut, pragmatically, we use the term ontology to describe both ontologies and KBs.

First generation SW applications (Motta and Sabou, 2006) are usually based on a single ontology, whose role is to support the integration of available data from resources selected at design time. However, as the SW is gaining momentum, we are seeing a dramatic increase in the amount of online distributed semantic data available on the Web³, which is unprecedented in the history of Artificial Intelligence. As discussed at length in (D'Aquin et al., 2008) these may provide the semantic basis for a new generation of intelligent systems, a turning point in the development of applications able to go beyond the brittle traditional knowledge acquisition task to dynamically exploiting and reusing the SW as a large-scale source of distributed knowledge.

The emergence of a large scale SW has re-ignited interest in NL front ends. As discussed in (McGuinness, 2004), the availability of semantic data on the Web opens the way to novel, sophisticated forms of *question answering* (QA), which can not only potentially provide increased precision and recall compared to today's search engines, but are also capable of additional functionalities, such as: i) proactively offering additional information about an answer, ii) providing measures of reliability and trust and iii) explaining how an answer was derived. The goal of QA systems (Hischman et al., 2001) is to allow users to receive concise answers to questions expressed in Natural Language (NL). QA systems have been investigated for many years by several communities (Hischman et al., 2001). Traditionally, approaches have largely focused on retrieving answers from raw text⁴. Therefore, most emphasis in QA has been on the use of ontologies to mark-up and to support query expansion (Mc Guinness, 2004).

Indeed, while semantic information can be used in several different ways to improve QA, an important consequence of the availability of distributed semantic mark-up on the Web is that this can be queried directly. As semantic data becomes ubiquitous, it will become advantageous to use

³ The semantic search engine Sindice reports to have indexed more than 26 million RDF documents.

⁴ Sponsored by the American National Institute (NIST) and the Defence Advanced Research Projects Agency (DARPA), TREC introduced an open-domain QA track in 1999 (TREC-8).

semantic data to be able to interpret queries and obtain formally derived answers, using NL expressions to enhance and go beyond the keyword-based retrieval mechanisms used by the current search engines.

1.3 Challenge: querying and exploring the SW content

With the emergence of initiatives like the schema.org and Linked Open Data, and the current interest of the major commercial search engines, Yahoo! SearchMonkey⁵ and Google Rich Snippets⁶ in the exploitation of SW content, the amount of semantic data available on the Web has significantly increased in the last few years. This semantic data has been generated by creating rich semantic resources such as FreeBase⁷ or DBPedia (Bizer et al., 2009b), by opening up large datasets previously hidden in backend databases, such as the ones released by the data.gov⁸ initiative, or by encouraging publishers to annotate their own Web content using RDFa⁹, or Microformats¹⁰.

Although this data growth opens new opportunities for SW applications that can exploit and reuse these data freely available, the diversity and massive volume currently reached by the publicly available semantic information introduces a **new research question**:

How can we support end users in querying and exploring this novel, massive and heterogeneous, structured information space?

As stated in (Buitelaar et al., 2003) to bridge the gap between the end user and semantic capabilities, language technologies and the SW can mutually benefit from each other. However, the current

⁵ <http://developer.yahoo.com/searchmonkey/>

⁶ <http://google.com/webmasters/tools/richsnippets>

⁷ FreeBase: <http://www.freebase.com/>

⁸ <http://data.gov.uk/>

⁹ <http://http://www.w3.org/TR/xhtml1-rdfa-primer/>

¹⁰ <http://microformats.org/>

approaches that have attempted to address this question suffer from one or more of the following limitations:

- a) They offer limited support for expressing queries, usually at the level of keyword-based search.
- b) They narrow their search scope, assuming that the knowledge is encoded in one, or a subset of, preselected homogeneous Knowledge Bases.
- c) They perform a shallow exploitation of the semantic data, i.e., do not take full advantage of the inherent semantics of the information space to extract the most accurate answers for the users.

As an example of approaches with limited expressivity at the level of query we can highlight ontology search engines and SW gateways such as Swoogle (Ding et al., 2005), Watson (d'Aquin et al., 2007), or Sindice (Oren et al., 2008). Although these systems accept keyword-based queries, which have proved to be user friendly, and work reasonably effectively when asked to find specific information items, such as “Russia” or “rivers”, they cannot be used to answer more complex and specific queries, where explicit relations between search terms are specified, such as “which Russian rivers end in the Black sea?”. For these types of queries, usability studies (Kaufmann and Bernstein, 2007) have demonstrated that casual users prefer the use of Natural Language (NL) queries to keywords when querying the semantic information space.

However, distributed, heterogeneous, large-scale semantic information introduces significant challenges. Consequently, a pervading drawback regarding ontology-based NL approaches is that they focus on a subset of pre-selected domain ontologies (or one single data graph) at a time (Lopez et al., 2007a), (Bernstein et al., 2006), (Cimiano et al., 2007), (Wang et al., 2007), (Tablan et al., 2008), (Linckels and Meinel, 2005).

The latest LOD search approaches (Wang et al., 2008) (Gueret et al., 2009) (Meij et al., 2009) do not restrict their scope to specific domains, or to specific datasets, but instead, attempt to address the whole corpus of publicly available semantic data. However, these approaches, while able to cope with the sheer scale and heterogeneity of the SW, only perform a shallow exploitation of the semantic information, which rely on users to filter out incorrect answers or disambiguate between different interpretations during the search process. These issues will be discussed in more detail in Chapter 2.

1.4 Theme and contributions of this thesis

With the continued growth of online semantic information, the processes of searching and querying large scale and heterogeneous content have become increasingly challenging. For this thesis we have developed a QA system, PowerAqua, designed to exploit the availability of distributed, ontology-based semantic data on the Web to provide answers to questions posed in NL. The system takes as input a NL query and translates it into a set of logical queries, which are then answered by consulting and aggregating information derived from multiple, heterogeneous, semantic sources on the fly. PowerAqua does not assume that the user has any prior information about the semantic resources. The user is not aware of which information sources exist, the details associated with interacting with each source, or the particular vocabulary used by the sources. In contrast to the previously mentioned semantic approaches for querying SW content (Section 1.3), PowerAqua is a system that:

- a) offers a NL query interface that balances usability and expressivity;
- b) is able to answer queries by locating and integrating information, which can be distributed across heterogeneous semantic resources;

- c) performs a deep exploitation of the available semantic information, providing query disambiguation, as well as knowledge fusion and ranking mechanisms to successfully elicit the most accurate answers to user queries.

PowerAqua follows from an earlier system, AquaLog, one of the first ontology-based NL QA systems (Lopez and Motta, 2004) (Lopez et al., 2005) (Lopez et al., 2007a). However, AquaLog was developed in the early days of the SW during a period when little semantic data was available online, and when applications tended to produce and consume their own data, like in traditional knowledge-based systems. As a result it only uses one ontology at a time, even though AquaLog is ontology independent and thus portable from one domain to the other. Nevertheless, the user needs to tell the system which ontology is going to be used to interpret the queries.

Since AquaLog was first implemented, there has been a resurgence of interest in NL front ends and the rise of ontology-based QA as a new paradigm of research, consistent with the crucial role played by ontologies in structuring semantic information on the Web. However, the scope of these systems is also limited to one (or a set) of a priori selected domains (Bernstein et al., 2006), (Cimiano et al., 2007), (Wang et al., 2007), (Tablan et al., 2008), (Linckels and Meinel, 2005). This works well in many scenarios, e.g., in company intranets, where a shared organizational ontology may be used to describe resources homogeneously, and where anything outside the intranet remains out of bounds. However, if we consider the SW in the large, this assumption no longer holds. The SW is heterogeneous in nature and it is not possible to determine in advance which ontologies will be relevant to a particular query. Moreover, it is often the case that queries can only be solved by composing information derived from multiple information sources that are autonomously created and maintained. Hence, to perform effective QA on the SW, a system must be able to locate and aggregate information, without any pre-formulated assumptions about the ontological structure of the relevant information.

PowerAqua has been conceived at a stage when the SW had started to expand, offering a wealth of semantic data that could be used for experimental purposes. Indeed, taking advantage of this expansion, PowerAqua represents a much more mature technology than state of the art ontology-based QA systems: it is not restricted by the single ontology assumption but can, in principle, retrieve answers from many, automatically discovered, semantic sources. PowerAqua's goal is to handle queries in which answers may be assembled in different ways by consulting and combining multiple sources.

Thus, PowerAqua is a multi-ontology based QA system, designed to take advantage of the vast amount of data offered by the SW in order to interpret a query. With PowerAqua we envision open semantic QA as a new research approach that enhances the scope of NL interfaces to databases (NLIDB), which has long been an area of research in the artificial intelligence and database communities (Androutsopoulos et al., 1995), and closed domain models over public, structured data, and presents complementary affordances to open QA over free text, a strong and well-founded research area stimulated since 1999 by the TREC QA track. In particular, a knowledge based QA system can help with answering questions requiring situation-specific answers, where multiple pieces of information (from one or several documents) need to be assembled to infer the answers at run time, rather than reciting a pre-written paragraph of text (Clark et al., 1999) (Hallet et al., 2007). As such, in this scenario, exploiting the SW is essentially about discovering interesting connections between items in a meaningful way. Hence, the main **contribution** of this thesis is twofold:

- To provide the first comprehensive attempt at supporting open domain QA on the semantic data publicly available on the SW.
- To tackle the problem of supporting users in locating and querying information on the SW.

In this thesis, we provide a complete overview of the PowerAqua system including:

- the research challenges that it addresses and the new requirements and implications imposed on traditional methods in ontology selection, matching and semantic similarity measures to balance heterogeneity and sheer scale with the need to provide results in real time;
- its architecture and how each component contributes to addressing different research challenges;
- an analysis of PowerAqua's feasibility through various evaluations, where we also provide a thorough discussion of our experiences with PowerAqua to support users in querying and exploring the SW content.

In addition, the implementation and evaluation of PowerAqua provide concrete insights into the current state and quality of the SW, its strengths (e.g., a powerful source of background knowledge) and its limitations (e.g., domain, instance or relational sparseness, modelling errors), while taking us a step closer to bridging the gap between real world users and the SW capabilities, and the realization of the SW vision.

1.5 Research approach: the multi ontology challenge

In the previous sections, we have sketched our vision for a QA system suitable for the SW, PowerAqua, and we have also explained why AquaLog did not quite fit the bill. In this section we address the problem in more detail, we examine the specific research issues that open domain semantic QA brings up and that needed to be tackled in order to develop PowerAqua.

PowerAqua was first envisioned in 2006 (Lopez et al., 2006) as a shift between the first generation of closed-domain, or domain-specific, semantic systems, and the next generation of open SW applications that aim to exploit the dynamic, heterogeneous nature of the SW, to harvest and reuse the rich ontological knowledge available. Considering a multi-ontology scenario brings several important challenges and two main research questions:

1. Up to what level QA systems are able to exploit the SW in the large and to interpret a query by means of different ontologies that are autonomously created and heterogeneous?.
2. Which new mechanisms that are not needed in traditional KB systems, where the knowledge is selected and integrated manually, are required to locate, disambiguate, integrate and rank heterogeneous semantic data on the fly?.

We focus here on the **research issues** that are specific to PowerAqua's two main research questions:

Resource discovery and information focusing:

PowerAqua aims to support QA in the open, dynamic and heterogeneous SW. In principle, any semantic data associated with any ontology can be potentially relevant to answer the user's information needs, and it is not possible to determine, in advance, which ontologies will be relevant to a particular query. Hence, PowerAqua has to identify automatically, at run-time, the relevant semantic data from a large and heterogeneous SW data space, depending on the content of the user's query. In particular, it needs to efficiently determine the relevance of the ontologies with respect to a query.

Mapping user terminology into ontology terminology:

A key design criterion for PowerAqua is that the user is free to use his / her own terminology when posing a query. This wide expressiveness adds a layer of complexity mainly due to the ambiguous nature of human language. So, while this is an issue for any NL interface, a critical problem for PowerAqua is that of different vocabularies used by different ontologies to describe similar information across domains. In PowerAqua the user terminology has to be translated into several ontology-centric terminologies, as several ontologies may, in principle, provide alternative readings on the same query and thus alternative answers, or parts of a composite answer. Robust disambiguation techniques are required that can potentially lead to a good answer in the context of a

user query. Hence, as user terminology may be translated into terminology distributed across ontologies, mapping and word sense disambiguation techniques need to be applied to avoid incoherent constructions, so that the concepts that are shared by assertions taken from different ontologies have the same sense.

Integrating information from different semantic sources:

Queries posed by end-users may be answered by a single or multiple knowledge sources, which require the integration of relevant *information* from different repositories. Thus, if there is a complete translation into one or more ontologies or if the current partial translation, in conjunction with previously generated partial translations, is equivalent to the original query, the data must be retrieved from the relevant ontologies, and appropriately combined and ranked to give the final, most accurate answers. Among other things, the problem of integrating information from multiple sources requires to be able to identify multiple occurrences of specific individuals in the sources in question.

Scalability:

The recent emergence of Linked Data, characterized by its openness, diversity and most importantly scale, has defined a new turning point in the evolution of the SW and its applications, introducing a new dimension of scale with respect to the availability of very large amounts of heterogeneous and distributed semantic information. Furthermore, in these large scale scenario, it becomes impossible to ensure strict data quality. Thus, coping with the sheer scale, heterogeneity, redundancy, incompleteness or conflicting data is a major challenge for semantic applications.

PowerAqua aims to address the aforementioned challenges, providing a step towards the realization of scalable and effective SW applications, able to deal with the new layers of complexity introduced by the continuous growth of the semantic data.

1.6 Thesis organization

In this chapter, we have presented the context, vision, ideas, challenges and contributions at the basis of this thesis. In the next chapter, Chapter 2, we discuss in detail the related work and state of the art, on the one hand discussing the different dimensions and developments in the broad QA research area, and on the other hand focusing on the current approaches for querying and exploring SW content. In Chapter 3 we discuss the research gap in the current state of the art that this thesis addresses, and the key design challenges facing the proposed research, in particular with respect the requirements for its matching algorithm. This is followed by an overview of the PowerAqua architecture and main components in Chapter 4.

In the second part of this thesis we describe PowerAqua's components in detail to give a comprehensive account of the way the system returns answers from user queries and addresses all the aforementioned challenges and research issues. This second part starts in Chapter 5 describing the linguistic component. It continues in Chapter 6 with the description of the PowerMap matching component. Chapter 7 describes the triple and relation similarity services that match user queries to ontological structures. Chapter 8 describes the work on the ranking and merging component, which integrates and ranks answers obtained from different sources.

Finally, in the third part of this thesis we describe and discuss different evaluations of the system in Chapter 9 and 10. Our latest experiments and lessons learned on scaling up PowerAqua to the recently emerged Linked Data content are presented in Chapter 11. We conclude in Chapter 12 by summarizing the main outcomes of this work and by outlining the main directions of research and development we are pursuing.

Chapter 2 Literature review

The survey presented in this chapter on the different dimensions of Question Answering systems and user-friendly search interfaces for the SW has been submitted for publication to the Semantic Web Journal (Lopez et al., 2011b).

For a further review on the topic of usability of semantic search tools, we refer the reader to our earlier work published in the Knowledge Engineering Review (Uren et al., 2007).

2.1 Introduction and outline

The emerging Semantic Web (SW) (Berners-Lee et al., 2001) offers a wealth of semantic data about a wide range of topics. We are quickly reaching the critical mass required to enable a true vision of a large scale, distributed SW with real-world datasets, leading to new research possibilities that can benefit from exploiting and reusing this vast resource, unprecedented in the history of computer science. Hence, there is now a renewed interest in the search engine market towards the introduction of semantics in order to improve over current keyword search technologies (Fazzinga and Lukasiewicz, 2010) (Hendler, 2010) (Baeza and Raghavan, 2010).

The notion of introducing semantics to search on the Web is not understood in a unique way. According to (Fazzinga and Lukasiewicz, 2010) the two most common uses of SW technology are: (1) to interpret Web queries and Web resources annotated with respect to the background knowledge described by underlying ontologies, and (2) to search in the structured large datasets and Knowledge Bases (KBs) of the SW as an alternative or a complement to the current Web.

Apart from the benefits that can be obtained as more semantic data is published on the Web, the emergence and continued growth of a large scale SW poses some challenges and drawbacks:

- There is a gap between users and the SW: it is difficult for end-users to understand the complexity of the logic-based SW. Solutions that can allow the typical Web user to profit from the expressive power of SW data-models, while hiding the complexity behind them, are of crucial importance.

- The processes of searching and querying content that is massive in scale and highly heterogeneous have become increasingly challenging: current approaches for querying semantic data have difficulties to scale their models successfully to cope with the increasing amount of distributed semantic data available online.

Hence, there is a need for user-friendly interfaces that can scale up to the Web of Data, to support end users in querying and exploring this heterogeneous information space.

Consistent with the role played by ontologies in structuring semantic information on the Web, recent years have witnessed the rise of ontology-based Question Answering (QA) as a new paradigm of research, to exploit the expressive power of ontologies and go beyond the relatively impoverished representation of user information needs in keyword-based queries. QA systems have been investigated by several communities (Hirschman and Gaizauskas, 2001), e.g., Information Retrieval (IR), artificial intelligence and database communities. Traditionally, QA approaches have largely been focused on retrieving answers from raw text, with the emphasis on using ontologies to mark-up Web resources and improve retrieval by using query expansion (McGuinness, 2004). The novelty of this trend of ontology-based QA is to exploit the SW information for making sense of, and answering, user queries.

In this chapter, we present a survey of ontology-based QA systems and other related work. We look at the promise of this novel research area from two perspectives. First, its contributions to the area of QA systems in general; and second, its potential to go beyond the current state of the art in SW interfaces for end-users, thus, helping to bridge the gap between the user and the SW.

We seek a comprehensive perspective on this novel area by analysing the key dimensions in the formulations of the QA problem in Section 2.2. We classify a QA system, or any approach to query the SW content, according to four dimensions based on the type of questions (input), the sources (unstructured data such as documents, or structured data in a semantic or non-semantic space), the scope (domain-specific, open-domain), and the traditional intrinsic problems and research issues

derived from the search environment (discovery, mapping, disambiguation, fusion, ranking, scalability). To start with, we introduce in Section 2.3 the general background and history of the QA research field, from the influential works in the early days of research on architectures for Natural Language Interfaces to Databases (NLIDB) in the 70s (Section 2.3.1), through the approaches to open domain QA over text (Section 2.3.2), to the latest proprietary (commercial) semantic QA systems, based on data that is by and large manually coded and homogeneous (Section 2.3.3). Then, in Section 2.4 we discuss the state of the art in ontology-based QA systems, in particular analysing their drawbacks (restricted domain) when considering the SW in the large. In Section 2.5, we focus on approaches developed in the last decade, that have attempted to support end users in querying and exploring the SW data in the large, from early global-view information systems (Section 2.5.1) and restricted domain semantic search (Section 2.5.2), to the latest works on open domain large scale semantic search and Linked Data (Bizer et al., 2009a) interfaces (Section 2.5.3). In Section 2.6, we argue that this new ontology-based search paradigm based on natural language QA, is a promising direction towards the realization of user-friendly interfaces for all the analysed dimensions, as it allows users to express arbitrarily complex information needs in an intuitive fashion. We conclude in Section 2.7 with an outlook for this research area, in particular, our view on the potential directions ahead to realize its ultimate goal: to retrieve and combine answers from multiple, heterogeneous and automatically discovered semantic sources.

2.2 Goals & Dimensions of Question Answering

The goal of QA systems, as defined by (Hirschman and Gaizauskas, 2001), is to allow users to ask questions in Natural Language (NL), using their own terminology, and receive a concise answer.

In this section we give an overview of the multiple dimensions in the QA process and the context in which this thesis has been developed. These dimensions can be extended beyond NL QA systems to any approach to help users to locate and query structured data on the Web. We discuss in Section 2.6 the state of the art, advances and limitations on this area according to these dimensions.

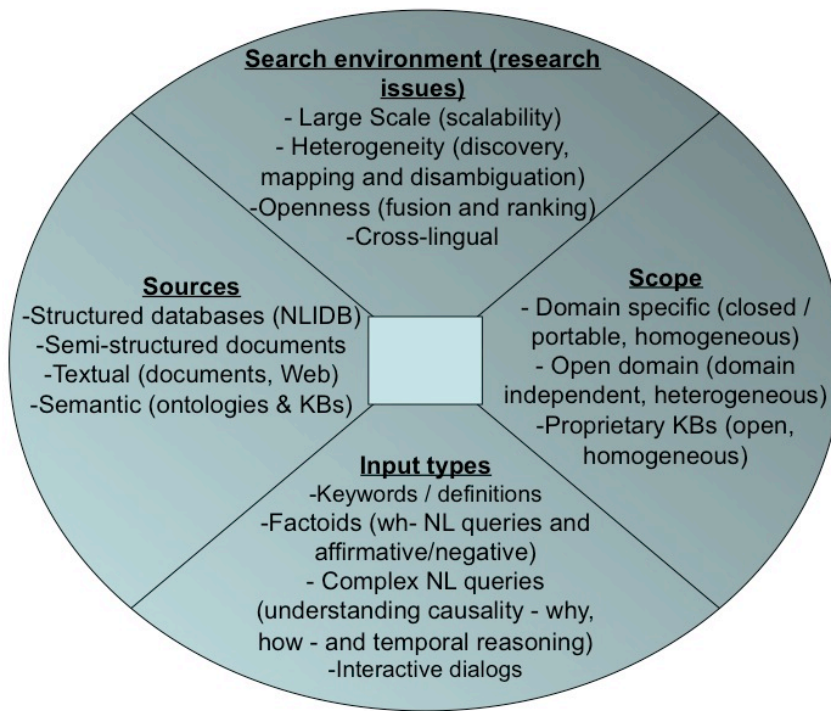


Figure 2.1 The dimensions of Question Answering and query and search interfaces in general.

We classify a QA system, and any semantic approach for searching and querying SW content, according to four interlinked dimensions (see Figure 2.1):

- (1) The input or type of questions it is able to accept (facts, dialogs, etc.).
- (2) The sources from which it can derive the answers (structured vs. unstructured).
- (3) The scope (domain specific vs. open domain).
- (4) How it copes with the traditional intrinsic problems that the search environment imposes in any non-trivial search system (e.g., adaptability and ambiguity).

At the **input** level, the issue is balancing usability and higher expressivity at the level of the query, hiding the complexity of SQL-like query languages, while allowing the user to express her information needs fully. Different kinds of search inputs provide complementary affordances to support the ordinary user in querying the semantic data. The best feature of keyword-based search is its simplicity. Nevertheless, in this simplicity lie its main limitations: the lack of expressivity, e.g., in

expressing relationships between words, and the lack of context to disambiguate between different interpretations of the keywords. An issue for NL-based interfaces is the level of expressiveness and linguistic phenomena they are able to cover (e.g., by using appropriate reasoning services), such as coordinations / disjunctions, quantifiers, elliptical sentences, temporal expressions, etc. Most research in QA focuses on factual QA, where we can distinguish between *Wh-queries* (who, what, how many, etc.), commands (name all, give me, etc.) requiring an element or list of elements as an answer, or affirmation / negation questions. As pointed out in (Hunter, 2000) more difficult kinds of factual questions include those which ask for opinion, like *Why* or *How* questions, which require understanding of causality or instrumental relations, *What* questions which provide little constraint in the answer type, and definition questions. In this survey we focus on factual QA, including open-domain definition questions, i.e., *What-queries* about arbitrary concepts. In the SW context factual QA means that answers are ground facts as typically found in KBs and provides an initial foundation to tackle more ambitious forms of QA.

QA systems can also be classified according to the different **sources** used to generate an answer as follows:

- Natural Language interfaces to structured data on databases (NLIDB traced back to the late sixties (Androutsopoulos et al., 1995)).
- QA over semi-structured data (e.g., health records, yellow pages, wikipedia infoboxes).
- Open QA over free text, fostered by the open-domain QA track introduced by TREC (<http://trec.nist.gov>) in 1999 (TREC-8).
- QA over structured semantic data, where the semantics contained in ontologies provide the context needed to solve ambiguities, interpret and answer the user query.

Another distinction between QA systems is whether they are domain-specific (closed domain) or open domain (thus domain-independent). Ontology-based QA emerged as a combination of ideas of

two different research areas - it enhances the **scope** of closed NLIDB over structured data, by being ontology-independent or portable to the domain of the ontology that it exploits; and also presents complementary affordances to open QA over free text (TREC), the advantage being that it can help with answering questions requiring situation-specific answers, where multiple pieces of information (from one or several sources) need to be assembled to infer the answers at run time. Nonetheless, ontology-based QA systems are akin to NLIDB in the sense that they are able to extract precise answers from structured data in a homogeneous and specific domain scenario, instead of retrieving relevant paragraphs of text in a heterogeneous and open scenario. Domain-specific systems vary on the degree of portability, from approaches that require a lot of configuration efforts to change the domain to fully portable approaches, which are independent of the domain of the source. However, contrary to open domain approaches, they only work in a specific domain at a time selected a priori. Latest proprietary QA systems over structured data, such as TrueKnowledge and Powerset (detailed in Section 2.3.3), are open domain but restricted to their own homogeneous, and by large manually coded, proprietary sources.

A challenge for domain-independent systems comes from the **search environment** that can be characterized by large scale, heterogeneity, openness and multilinguality. The search environment influences to what level semantic systems perform a deep exploitation of the semantic data. In order to take full advantage of the inherent characteristics of the semantic information space to extract the most accurate answers for the users, QA systems need to tackle various **traditional research issues** derived from the search environment, such as:

- Mapping the terminology and information needs of the user into the terminology used by the sources, in such a form that: (1) it finds the sources relevant to the user query on the fly, (2) it does not affect portability or adaptability of the systems to new domains, and (3) it leads to the correct answer, due to the approximate nature of the mappings between the user query

and the ontology terms in a highly heterogeneous and open scenario, there may be more than a single correct answer to a given query across ontologies.

- Disambiguating between all possible interpretations of a user query. Independently of the type of query, any non-trivial NL QA system has to deal with ambiguity. Furthermore, in an open scenario, ambiguity cannot be solved by means of an internal unambiguous knowledge representation, as in domain-restricted scenarios. In open-domain scenarios, systems face the problem of disambiguating polysemous words, whose senses are unknown or which may have different meanings according to different domains across ontologies (similarity / dissimilarity of different ontological concepts).
- Knowledge fusion and ranking measures should be applied to select the better sources, combine partial answers, fuse similar answers together, and rank alternative answers across sources. This is because answers may come from different sources and the mappings across sources are approximate and have varying levels of quality and trust.
- With regards to scalability, in general terms, there is a trade-off between the complexity of the querying process and the amount of data systems can use in response to a user demand in a reasonable time.

Multilinguality issues, the ability to answer a question posed in one language using an answer corpus in another language, fostered by the Multilingual Question Answering Track at the cross language evaluation forum, CLEF (<http://clef.isti.cnr.it>), since 2002 (Forner et al., 2010), are not covered in this thesis.

NL interfaces are an often-proposed solution in the literature for casual users (Kauffman and Bernstein, 2007), being particularly appropriate in domains for which there are authoritative and comprehensive databases or resources (Mollá and Vicedo, 2007). However, their success has been typically overshadowed by both the brittleness and *habitability* problems (Thompson et al., 2005), defined as the mismatch between the user expectations and the capabilities of the system with

respect to its NL understanding and what it knows about (users do not know what it is possible to ask). As stated in (Uren et al., 2007) iterative and exploratory search modes are important to the **usability** of all search systems, to support the user in understanding what is the knowledge of the system and what subset of NL is possible to ask about. Systems also should be able to provide justifications for an answer in an intuitive way (NL generation), suggest the presence of unrequested but related information, and actively help the user by recommending searches or proposing alternate paths of exploration. For example, view based search and forms can help the user to explore the search space better than keyword-based or NL querying systems, but they become tedious to use in large spaces and impossible in heterogeneous ones.

Usability of NL interfaces is not covered in this review so for additional information we refer the reader to (Uren et al., 2007) and (Kauffman and Bernstein, 2007).

2.3 Related work on Question Answering

Here we present a short survey of related work on QA targeted to different types of sources: structured databases, unstructured free text and precompiled fact-based KBs.

2.3.1 NLIDB: Natural Language Interfaces to Databases

The use of NL to access relational databases can be traced back to the late sixties and early seventies (Androutsopoulos et al., 1995). The first QA systems were developed in the sixties and they were basically NL interfaces to expert systems, tailored to specific domains, the most famous ones being **BASEBALL** (Green et al., 1961) and **LUNAR** (Woods, 1973). Both systems were domain specific, the former answered questions about the US baseball league over the period of one year, the later answered questions about the geological analysis of rocks returned by the Apollo missions. LUNAR was able to answer 90% of the questions in its domain when posed by untrained geologists. In (Androutsopoulos et al., 1995) a detailed overview of the state of the art for these early systems can be found.

Some of the early NLIDB approaches relied on pattern-matching techniques. In the example described by (Androutsopoulos et al., 1995), a rule says that if a user's request contains the word "capital" followed by a country name, the system should print the capital which corresponds to the country name, so the same rule will handle "what is the capital of Italy?", "print the capital of Italy", "could you please tell me the capital of Italy". This shallowness of the pattern-matching would often lead to failures but it has also been an unexpectedly effective technique for exploiting domain-specific data sources.

The main drawback of these early NLIDB systems is that they were built having a particular database in mind, thus they could not be easily modified to be used with different databases and were difficult to port to different application domains. Configuration phases were tedious and required a long time, because of domain-specific grammars, hard-wired knowledge or hand-written mapping rules that had to be developed by domain experts.

The next generation of NLIDBs used an intermediate representation language, which expressed the meaning of the user's question in terms of high-level concepts, independently of the database's structure (Androutsopoulos et al., 1995). Thus, separating the (domain-independent) linguistic process from the (domain-dependent) mapping process into the database, to improve the portability of the front end (Martin et al., 1985).

The **formal semantics approach** presented in (De Roeck et al., 1991) follows this paradigm and clearly separates the NL front ends, which have a very high degree of portability, from the back end. The front end provides a mapping between sentences of English and expressions of a formal semantic theory, and the back end maps these into expressions, which are meaningful with respect to the domain in question. Adapting a developed system to a new application domain requires altering the domain specific back end alone.

MASQUE/SQL (Androutsopoulos et al., 1993) is a portable NL front end to SQL databases. It first translates the NL query into an intermediate logic representation, and then translates the logic

query into SQL. The semi-automatic configuration procedure uses a built-in domain editor, which helps the user to describe the entity types to which the database refers, using an is-a hierarchy, and then declaring the words expected to appear in the NL questions and defining their meaning in terms of a logic predicate that is linked to a database table/view.

Relevant advances in this area (2003) can be found in **PRECISE** (Popescu et al., 2003). PRECISE maps questions to the corresponding SQL query by identifying classes of questions that are understood in a well defined sense: the paper defines a formal notion of *semantically tractable* questions. Questions are translated into sets of attribute/value pairs and a relation token corresponds to either an attribute token or a value token. Each attribute in the database is associated with a wh-value (what, where, etc.). Also, a lexicon is used to find synonyms. The database elements selected by the matcher are assembled into a SQL query, if more than one possible query is found, the user is asked to choose between the possible interpretations. However, in PRECISE the problem of finding a mapping from the tokenization to the database requires all tokens to be distinct; questions with unknown words are not semantically tractable and cannot be handled. As a consequence, PRECISE will not answer a question that contains words absent from its lexicon. Using the example suggested in (Popescu et al., 2003), the question “what are some of the neighbourhoods of Chicago?” cannot be handled by PRECISE because the word “neighbourhood” is unknown. When tested on several hundred questions, 80% of them were semantically tractable questions, which PRECISE answered correctly, and the other 20% were not handled.

NLI have attracted considerable interest in the Health Care area. In the approach presented in (Hallet et al., 2007) users can pose complex NL queries to a large medical repository, question formulation is facilitated by means of *Conceptual Authoring*. A logical representation is constructed using a query editing NL interface, where, instead of typing in text, all editing operations are defined directly on an underlying logical representation governed by a predefined ontology ensuring that no problem of interpretation arises.

However, all these approaches still need an intensive configuration procedure. To reduce the formal complexity of creating underlying grammars and representations for different domains, (Minock et al., 2008), and most recently **C-PHRASE** (Minock et al., 2010) present a state-of-the-art authoring system for NLIDB. The author builds the semantic grammar through a series of naming, tailoring and defining operations within a Web-based GUI, as such the NLI can be configured by non-specialized, Web based technical teams. In that system queries are represented as expressions in an extended version of Codd's Tuple Calculus, which may be directly mapped to SQL queries or first-order logic expressions. Higher-order predicates are also used to support ranking and superlatives.

2.3.2 Open Domain Question Answering over documents

2.3.2.1 Document-based Question Answering

Most current work on QA, which has been rekindled largely by the TREC Text Retrieval Conference (sponsored by the American National Institute, NIST, and the Defence Advanced Research Projects Agency, DARPA) and by the cross-lingual QA Track at CLEF, is somewhat different in nature from querying structured data. These evaluation campaigns foster research in QA from the IR perspective, where the task consists in finding the text that contains the answer to the question and extracting the answer. The ARDA's Advanced Question Answering for Intelligence funded the AQUAINT program, a multi-project effort to improve the performance of QA systems over free large heterogeneous collections of structured and unstructured text or media. Given the large, uncontrolled text files and the very weak world knowledge available from lexical libraries, such as WordNet (Miller, 1995), and gazetteers, these systems have performed well. For example, the **LCC system** (Moldovan et al., 2002) that uses a deep linguistic analysis and an iterative strategy correctly answered 415 questions out of 500 in TREC-11 (2002).

There are linguistic problems common in most kinds of NL understanding systems. A high-level overview on the state of the art techniques for open QA can be found in (Pasca, 2003). Some of the methods use shallow keyword-based expansion techniques to locate interesting sentences from the

retrieved documents, based on the presence of words that refer to entities of the same type of the expected answer type. Ranking is based on syntactic features such as word order or similarity to the query. Templates can be used to find answers that are just reformulations of the question. Most of the systems classify the query based on the type of the answer expected: e.g., a name (i.e. person, organization), a quantity (monetary value, distance, length, size) or a date. Classes of questions are arranged hierarchically in taxonomies and different types of questions require different strategies. These systems often utilize world knowledge that can be found in large lexical resources such as WordNet, or ontologies such as the Suggested Upper Merged Ontology (SUMO) (Pease et al., 2002) to pinpoint question types and match entities to the expected answer type. More sophisticated syntactic, semantic and contextual processing to construct an answer might include: named-entity (NE) recognition, relation extraction, co-reference resolution, syntactic alternations, word sense disambiguation (WSD), logical inferences and temporal-spatial reasoning.

Going into more details, QA applications for text typically involve two steps, as pointed out by (Hirschman and Gaizauskas, 2001): (1) “identifying the semantic type of the entity sought by the question”; and (2) “determining additional constraints on the answer entity”. Constraints can include, for example, keywords (that may be expanded using synonyms or morphological variants) to be used in the matching of candidate answers, and syntactic or semantic relations between a candidate answer entity and other entities in the question. Various systems have, therefore built hierarchies of question types based on the types of answers sought (Moldovan et al., 1999) (Hovy et al., 2000) (Wu et al., 2003) (Srihari et al., 2004). NE recognition and information extraction (IE) are powerful tools in free text QA. The study presented in (Srihari et al., 2004) showed that over 80% of questions asked for a named entity as a response.

For instance, in **LASSO** (Moldovan et al., 1999) a question type hierarchy was constructed from the analysis of the TREC-8 training data, and a score of 55.5% for short answers and 64.5% for long answers was achieved. Given a question, LASSO can find automatically (a) the type of the question

(what, why, who, how, where), (b) the type of the answer (person, location, etc.), (c) the focus of the question, defined as the “main information required by the interrogation” (useful for “what” questions, which usually leave implicit the type of the answer which is sought), (d) the relevant keywords from the question. Occasionally, some words of the question do not occur in the answer (for example, the focus “day of the week” is very unlikely to appear in the answer). Therefore, LASSO implements NE recognition heuristics for locating the possible answers.

The best results of the TREC-9 (De Boni, M., 2001) competition were obtained by the **FALCON** system described in (Harabagiu et al., 2000), with a score of 58% for short answers and 76% for long answers. In FALCON the semantic categories of the answers are mapped into categories covered by a NE Recognizer. When the answer type is identified, it is mapped into an answer taxonomy, where the top categories are connected to several word classes from WordNet. In an example presented in (Harabagiu et al., 2000), FALCON identifies the expected answer type of the question “what do penguins eat?” as food because “it is the most widely used concept in the glosses of the sub-hierarchy of the noun synset {eating, feeding}”. All nouns (and lexical alterations), immediately related to the concept that determines the answer type, are considered among the other query keywords. Also, FALCON gives a cached answer if the similar question has already been asked before; a similarity measure is calculated to see if the given question is a reformulation of a previous one.

The system described in Litkowski (Litkowski, 2001), called **DIMAP**, extracts “semantic relation triples” after a document is parsed, converting a document into triples. The DIMAP triples are stored in a database in order to be used to answer the question. The semantic relation triple described consists of a discourse entity, a semantic relation that characterizes the entity’s role in the sentence and a governing word to which the entity stands in the semantic relation. The parsing process generates an average of 9.8 triples per sentence in a document. The same analysis was done for each question, generating on average 3.3 triples per sentence, with one triple for each question containing

an unbound variable, corresponding to the type of question (the system categorized questions in six types: time, location, who, what, size and number questions).

2.3.2.2 Question Answering on the Web

QA systems over the Web have the same three main components as QA systems designed to extract answers to factual questions by consulting a repository of documents (TREC): (1) a query formulation mechanism that translates the NL queries into the required IR queries, (2) a search engine over the Web, instead of an IR engine searching the documents, and (3) the answer extraction module that extracts answers from the retrieved documents. A technique commonly shared in Web and TREC-systems, is to use WordNet or NE tagging to classify the type of the answer.

For instance, **Mulder** (Kwok et al., 2001) is a QA system for factual questions over the Web, which relies on multiple queries sent to the search engine Google. To form the right queries for the search engine, the query is classified using WordNet to determine the type of the object of the verb in the question (numerical, nominal, temporal), then a reformulation module converts a question into a set of keyword queries by using different strategies: extracting the most important keywords, quoting partial sentences (detecting noun phrases), conjugating the verb, or performing query expansion with WordNet. In Mulder, an answer is extracted from the snippets or summaries returned by Google, which is less expensive than extracting answers directly from a Web page. Then, to reduce the noise or incorrect information typically found on the Web and improve accuracy, Mulder clusters similar answers together and picks the best answer with a voting procedure. Mulder takes advantage of Google ranking algorithms base on PageRank, the proximity or frequency of the words, and the wider coverage provided by Google: “with a large collection there is a higher probability of finding target sentences”. An evaluation using the TREC-8 questions, based on the Web, instead of the TREC document collection, showed that Mulder’s recall is more than a factor of three higher than AskJeeves.

The search engine **AskJeeves** (www.ask.co.uk) looks up the user's question in its database and returns a list of matching questions that it knows how to answer, the user selects the most appropriate entry in the list, and he is taken to the Web pages where the answer can be found. AskJeeves relies on human editors to match question templates with authoritative sites.

Other approaches are based on statistical or semantic similarities. For example, **FAQ Finder** (Burke et al., 1997) is a NL QA system that uses files of FAQs as its KB; it uses two metrics to match questions to answers: statistical similarity and semantic similarity. For shorter answers over limited structured data, NLP-based systems have generally performed better than statistical based ones, which need a lot of domain specific training and long documents with large quantities of data containing enough words for statistical comparisons to be considered meaningful. Semantic similarity scores rely on finding connections through WordNet between the user's question and the answer. The main problem here is the inability to cope with words that are not explicitly found in the KB. **Gurevych's** (Gurevych et al., 2009) **approach** tries to identify semantically equivalent questions, which are paraphrases of user queries, already answered in social Q&A sites, such as Yahoo!Answers.

Finally, Google itself is also evolving into a NL search engine, providing precise answers to some specific factual queries, together with the Web pages from which the answers have been obtained. However, it does not yet distinguish between queries such as "where Barack Obama was born" or "when Barack Obama was born" (as per May 2011).

2.3.3 Developments on commercial open QA

As we have seen in the previous subsections, work on large-scale, open-domain QA has been stimulated in the last decade (since 1999) by the TREC QA track evaluations. The current trend is to introduce semantics to search for Web pages based on the meaning of the words in the query, rather than just matching keywords and ranking pages by popularity. Within this context, there are also approaches that focus on directly obtaining structured answers to user queries from pre-compiled

semantic information, which is used to understand and disambiguate the intended meaning and relationships of the words in the query.

This class of systems includes START, which came online in 1993 as the first QA system available on the Web, and several industrial startups such as Powerset, Wolfram Alpha and True Knowledge¹¹, among others. These systems use a well-established approach, which consists of semi-automatically building their own homogeneous, comprehensive factual KB about the world, similarly to OpenCyc and Freebase¹².

START (Katz et al., 2002) answers questions about geography and the MIT infolab, with a performance of 67% over 326 thousand queries. It uses highly edited KBs to retrieve tuples and adopts a triple-base data model called “object-property-value”. Using the example presented in (Katz et al., 2002): “what languages are spoken in Guernsey?”, START considers “languages” as the property linking the object “Guernsey” and the value “French”. START compares the user query against the annotations derived from the KB. However, START suffers from the knowledge acquisition bottleneck, as only trained individuals can add knowledge and expand the system’s coverage (by integrating new Web sources).

Commercial systems include PowerSet, which tries to match the meaning of a query with the meaning of a sentence in Wikipedia. Powerset not only works on the query side of the search (converting the NL queries into database understandable queries, and then highlighting the relevant passage of the document), but it also reads every word of every (Wikipedia) page to extract the semantic meaning. It does so by compiling *factzs* - similar to triples, from pages across Wikipedia, together with the Wikipedia page locations and sentences that support each factz and using Freebase and its semantic resources to annotate them. The Wolfram Alpha knowledge inference engine builds

¹¹ <http://www.powerset.com/>, <http://www.wolframalpha.com/index.html>, and <http://www.trueknowledge.com/>.

¹² www.openencyc.org and <http://www.freebase.com>

a broad trusted KB about the world by ingesting massive amounts of information (storing approximately 10TBs, still a tiny fraction of the Web), while True Knowledge relies on users to add and curate information.

2.4 Ontology-based Question Answering

In this section we look at ontology-based semantic QA systems (also referred in this thesis as semantic QA systems), which take queries expressed in NL and an ontology as input, and return answers drawn from one or more KBs that subscribe to the ontology. Therefore, they do not require the user to learn the vocabulary or structure of the ontology to be queried. We will begin our discussion with AquaLog, one of the first ontology-based QA prototypes in the literature to query Semantic Web data (Lopez and Motta, 2004), and the basis for the later work on PowerAqua.

2.4.1 The Pioneer: the AquaLog QA system

AquaLog (Lopez and Motta, 2004), (Lopez, et al., 2005) (Lopez et al., 2007a) allows the user to choose an ontology and then ask NL queries with respect to the universe of discourse covered by the ontology. AquaLog is ontology independent because the configuration time required to customize the system for a particular ontology is negligible. The reason for this is that the architecture of the system and the reasoning methods are completely domain-independent, relying on an understanding of general-purpose knowledge representation languages, such as RDF or OWL¹³, and the use of generic lexical resources, such as WordNet.

The major task of the system is to bridge the gap between the terminology used by the user and the concepts used by the underlying ontology. In AquaLog different strategies are combined together to make sense of user queries with respect to the target KB. AquaLog uses a sequential process model (see Figure 2.2), in which a NL input is first translated into a set of intermediate

¹³ The *World Wide Web Consortium (W3C)* has published a number of standards and specific recommendations for the SW: the RDF, RDFS and OWL general purpose languages and the SPARQL query language.

representations. In a first step, the Linguistic Component uses the GATE infrastructure and resources (Cunningham et al., 2002) to obtain a set of linguistic annotations associated with the input query. The set of annotations is extended by the use of JAPE grammars¹⁴ to identify terms, relations, question indicators (who, what, etc.), features (voice and tense) and to classify the query into a category. Knowing the category and GATE annotations for the query, the Linguistic Component creates the linguistic triples or Query-Triples. Then, these Query-Triples are further processed and interpreted by the Relation Similarity Service, which maps the Query-Triples to ontology-compliant Onto-Triples, from which an answer is derived. AquaLog identifies ontology mappings for all the terms and relations in the Query-Triples by means of string based comparison methods and WordNet. In addition, AquaLog's interactive relation similarity service uses the ontology taxonomy and relationships to disambiguate between the alternative representations of the user query. When the ambiguity cannot be resolved by domain knowledge the user is asked to choose between the alternative readings.

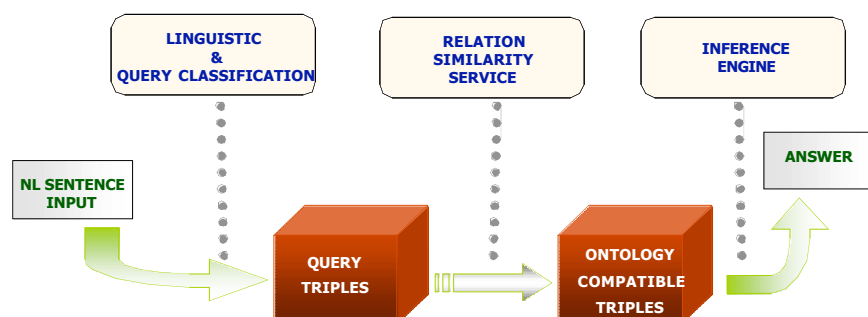


Figure 2.2. The AquaLog Data Model

Moreover, AquaLog includes a learning component to automatically obtain domain-dependent knowledge by creating a lexicon, which ensures that the performance of the system improves over time, in response to the particular community jargon (vocabulary) used by end users. Thus, users can

¹⁴ JAPE is a language for creating regular expressions applied to linguistic annotations in a text corpus

easily correct mistakes and disambiguate between different interpretations of a query. AquaLog uses generalization rules to learn novel associations between the relations used by the users and the ontology structure. Once the question is entirely mapped to the underlying ontological structure the corresponding instances are obtained as an answer.

To briefly illustrate the QA process, imagine that the system is asked the question: “Which projects are related to researchers working with ontologies?”¹⁵. In a first stage, by using linguistic techniques, the system interprets the NL question and translates it to triple-like data structures. Then, these triples are compared to the underlying ontology using a set of string comparison methods and WordNet. As such, the term *projects* is matched to the ontology concept *Project*, *researchers* to the ontology concept *Researcher*, and *ontologies* is assumed equivalent to the *ontologies* instance of the *Research-Area* concept. After the modifier attachment is resolved (to disambiguate how the terms link to each other) by using domain knowledge, two triples are identified: (projects, related to, researchers) and (researchers, working, ontologies). The relations of the triples are also mapped to the ontology. For example, for the second triple, there is only one known relation in the ontology between a *Researcher* and a *Research-area*, namely *has-research-interest*. This relation is assumed to be the relevant one for the question. However, when disambiguating the relation that is referred to by *related to*, the system cannot find any syntactically similar relation between a *Project* and a *Researcher* (or between all more generic and more specific classes of the two concepts). Nevertheless, there are four, alternative relations between these two concepts: *has-contact-person*, *has-project-member*, *has-project-leader*, *uses-resource*. The user is asked to choose the relation that is closest to his interest. Over time, the system can learn a particular user’s “jargon” from these interventions. Once a choice is made, the question is entirely mapped to the underlying ontological structure and the corresponding instances can be retrieved as an answer.

¹⁵ Using the KMi populated version of the AKT ontology: <http://kmi.open.ac.uk/projcest/akt/ref-onto>

2.4.2 A review of ontology-based QA systems

Since the steady growth of the SW and the emergence of large-scale semantics the necessity of NLI to ontology-based repositories has become more acute, re-igniting interest in NL front ends. This trend has also been supported by usability studies (Kaufmann and Bernstein, 2007), which show that casual users, typically overwhelmed by the formal logic of the SW, prefer to use a NL interface to query an ontology. Hence, in the past few years there has been much interest in ontology based QA systems, where the power of ontologies as a model of knowledge is directly exploited for query analysis and translation, thus providing a new twist on the old issues of NLIDB, by focusing on portability and performance, and replacing the costly domain specific NLP techniques with shallow but domain-independent ones. A wide range of off-the-shelf components, including triple stores (e.g., Sesame¹⁶), text retrieval engines (e.g., Lucene¹⁷), domain-independent linguistic resources, such as WordNet and FrameNet¹⁸, and NLP Parsers, such as Stanford Parser (Klein and Manning, 2002), support the evolution of these new NLI.

Ontology-based QA systems vary on two main aspects:

(1) The degree of domain customization they require, which correlates with their retrieval performance.

(2) The subset of NL they are able to understand (full grammar-based NL, controlled or guided NL, pattern based), in order to reduce both complexity and the habitability problem, pointed out as the main issue that hampers the successful use of NLI (Kaufmann and Bernstein, 2007).

At one end of the spectrum, systems are tailored to a domain and most of the customization has to be performed or supervised by domain experts. For instance QACID (Fernandez, O. et al., 2009) is

¹⁶ <http://www.openrdf.org/>

¹⁷ <http://lucene.apache.org/>

¹⁸ <http://wordnet.princeton.edu> , <http://framenet.icsi.berkeley.edu>

based on a collection of queries from a given domain that are analysed and grouped into clusters, where each cluster, containing alternative formulations of the same query, is manually associated with SPARQL queries. In the middle of the spectrum, a system such as ORAKEL (Cimiano et al., 2007) requires a significant domain-specific lexicon customization process, while for systems like the e-librarian (Linckels and Meinel, 2005) performance is dependent on the manual creation of a domain dependent lexicon and dictionary. At the other end of the spectrum, in systems like AquaLog (detailed in Section 2.4.1), the customization is done on the fly while the system is being used, by using interactivity to learn the jargon of the user over time. GINSENG (Bernstein et al., 2006) guides the user through menus to specify NL queries, while systems such as PANTO (Wang, 2007), NLP-Reduce, Querix (Kaufmann et al., 2006) and QuestIO (Tablan et al., 2008), generate lexicons, or ontology annotations (FREya by Damjanovic et al.), on demand when a KB is loaded. In what follows, we look into these systems in detail and present a comparison in Table 2.1.

QACID (Fernandez, O. et al., 2009) relies on the ontology, a collection of user queries, and an entailment engine that associates new queries to a cluster of existing queries. Each query is considered as a bag of words, the mapping between words in NL queries to instances in a KB is done through string distance metrics (Cohen et al., 2003) and an ontological lexicon. Prior to launching the corresponding SPARQL query for the cluster, the SPARQL generator replaces the ontology concepts with the instances mapped for the original NL query. This system is at the domain-specific end of the spectrum because the performance depends on the variety of questions collected in the domain, the process is domain-dependent, costly and can only be applied to domains with limited coverage.

ORAKEL (Cimiano et al., 2007) is a NL interface that translates factual *wh-queries* into F-logic or SPARQL and evaluates them with respect to a given KB. The main feature is that it makes use of a compositional semantic construction approach thus being able to handle questions involving quantification, conjunction and negation. In order to translate factual *wh-queries* it uses an

underlying syntactic theory built on a variant of a *Lexicalized Tree Adjoining Grammar* (LTAG), extended to include ontological information. The parser makes use of two different lexicons: the general lexicon and the domain lexicon. The general or domain independent lexicon includes closed-class words such as determiners, i.e., a, the, every, etc., as well as question pronouns, i.e., who, which, etc. The domain lexicon, in which natural expressions, verbs, adjectives and relational nouns, are mapped to corresponding relations specified in the domain ontology, varies from application to application and, for each application, this lexicon has to be partially generated by a domain expert. The semantic representation of the words in the domain independent lexicon makes reference to domain independent categories, as given for example by a foundational ontology such as DOLCE. This assumes that the domain ontology is somehow aligned to the foundational categories provided by the foundational ontology. Therefore, the domain expert is only involved in the creation of the domain specific lexicon, which is actually the most important lexicon as it is the one containing the mapping of linguistic expressions to domain-specific predicates. The domain expert has to instantiate *subcategorization frames*, which represent linguistic structures (e.g., verbs with their arguments), and maps these to domain-specific relations in the ontology. WordNet is used with the purpose to suggest synonyms (in the most frequent sense of the word) for the verb or noun currently edited. The approach is independent of the target language, which only requires a declarative description in Prolog of the transformation from the logical form to the target language.

The **e-Librarian** (Linckels and Meinel, 2005) understands the sense of the user query to retrieve multimedia resources from a KB. First, the NL query is pre-processed into its linguistic classes, in the form of triples, and translated into an unambiguous logical form, by mapping the query to an ontology to solve ambiguities. If a query is composed of several linguistic clauses, each one is translated separately and the logical concatenation depends on the conjunction words used in the question. The system relies on simple, string-based comparison methods (e.g., *edit distance metrics*) and a domain dictionary to look up lexically related words (synonyms) because general-purpose dictionaries like WordNet are often not appropriate for specific domains. Regarding portability, the

creation of this dictionary is costly, as it has to be created for each domain, but the strong advantage of this is that it provides very high performance, which is difficult to obtain with general-purpose dictionaries (from 229 user queries, 97% were correctly answered in the evaluation). The e-librarian does not return the answer to the user's question, but it retrieves the most pertinent document(s) in which the user finds the answer to her question.

Moving into the systems that do not necessitate any customization effort or previous pre-processing, (Kaufmann and Bernstein, 2007) presented four different ontology independent query interfaces with the purpose of studying the usability of NLI for casual end-users. These four systems lie at different positions of what they call the *Formality Continuum*, where the freedom of a full NL and the structuredness of a formal query language are at opposite ends of the continuum. The first two interfaces, *NLP-Reduce* and *Querix* allow users to pose questions in full or slightly controlled English. The third interface *Ginseng* offers query formulation in a controlled language akin to English. Therefore, the first three interfaces lie on the NL end of the Formality Continuum towards its middle. As such, they analyse a user query, match it to the content of a KB, and translate these matches into statements of a formal query language (i.e., SPARQL) in order to execute it. The last interface, *Semantic Crystal*, belongs to the formal approaches, as it exhibits a graphical query language. The guided and controlled entry overcomes the habitability problem of NL systems (providing a trade-off between structuredness and freedom) and ensuring all queries make sense in the context of the loaded KB. However, as stated in this usability study “users favour query languages that impose some structure but do not overly restrict them”, thus, from the four systems, Querix was the interface preferred by the users, which query language (full English) was perceived as a natural, not formal, guiding structure.

The interface that has the least restrictive and most natural query language, **NLP-Reduce** (Kaufmann, Bernstein and Fischer, 2007), allows almost any NL input (from ungrammatical inputs, like keywords and sentence fragments, to full English sentences). It processes NL queries as bags of

words, employing only two basic NLP techniques: stemming and synonym expansion. Essentially, it attempts to match the parsed question words to the synonym-enhanced triples stored in the lexicon (the lexicon is generated from a KB and expanded with WordNet synonyms), and generates SPARQL statements for those matches. It retrieves all those triples for which at least one of the question words occur as an object property or literal, favouring triples which cover most words and with best matches, and joins the resultant triples to cover the query.

The second interface **Querix** (Kaufmann et al., 2006) is also a pattern matching NLI, however, the input is narrowed to full English (grammatically correct) questions, restricted only with regard to sentence beginnings (i.e., only questions starting with “which”, “what”, “how many”, “how much”, “give me” or “does”) and a mandatory question mark or full stop. In contrast with NLP-Reduce, Querix makes use of the syntactic structure of input questions to find better matches in the KB. Querix uses the Stanford parser to analyse the input query, then, from the parser’s syntax tree, extended with WordNet synonyms, it identifies triple patterns for the query. These triple patterns are matched in the synonym-enhanced KB by applying pattern matching algorithms. When a KB is chosen, the RDF triples are loaded into a Jena model, using the Pellet reasoner¹⁹ to infer all implicitly defined triples and WordNet to produce synonym-enhanced triples. Pattern matching is done by searching for triples that include one of the nouns or verbs in the query. Querix does not try to resolve NL ambiguities, but asks the user for clarifications in a pop-up dialog menu window to disambiguate. Several triples can be retrieved for the nouns, verbs and their synonyms. Those that match the query triples are selected, and from these, a SPARQL query is generated to be executed in the Jena’s SPARQL engine.

In the middle of the formality continuum, **Ginseng** (Bernstein et al., 2006) controls a user’s input via a fixed vocabulary and predefined sentence structures through menu-based options, as such it

¹⁹ <http://pellet.owldl.com>

falls into the category of *guided input NL* interfaces, similar to **LingoLogic** (Thompson et al., 2005). These systems do not try to understand NL queries but they use menus to specify NL queries in small and specific domains. GINSENG uses a small static grammar that is dynamically extended with elements from the loaded ontologies, without using any predefined lexicon beyond the vocabulary that is defined in the static sentence grammar and provided by the loaded ontologies. When the user enters a sentence, an incremental parser relies on the grammar to constantly (1) propose possible continuations to the sentence, and (2) prevent entries that would not be grammatically interpretable.

PANTO (Wang et al., 2007) is a portable NLI that takes a NL question as input and executes a corresponding SPARQL query on a given ontology model. It relies on the statistical Stanford parser to create a parse tree of the query from which triples are generated. These triples are mapped to the triples in the lexicon. The lexicon is created when a KB is loaded into the system, by extracting all entities enhanced with WordNet synonyms. Following the AquaLog model, it uses two intermediate representations: the Query-Triples, which rely solely on the linguistic analysis of the query, and the Onto-Triples that match the query triples and are extracted using the lexicon, string distance metrics and WordNet. PANTO can handle conjunctions / disjunctions, negation, comparatives and superlatives (those that can be interpreted with *Order by* and *Limit* on *datatype*, superlatives that require the functionality *count* are not supported).

Similarly, in **QuestIO** (Tablan et al., 2008) NL queries are translated into formal queries but the system is reliant on the use of gazetteers initialized for the domain ontology. In QuestIO users can enter queries of any length and form. QuestIO works by recognizing concepts inside the query through the gazetteers, without relying on other words in the query. It analyses potential relations between concept pairs and ranks them according to string similarity measures, the specificity of the property or distance between terms. QuestIO supports conjunction and disjunction.

FRyA (Damljanovic et al., 2010) is the successor to QuestIO, providing improvements with

respect to a deeper understanding of a question's semantic meaning, to better handle ambiguities when ontologies are spanning diverse domains. FREyA allows users to enter queries in any form. Therefore, to identify the answer type of the question and present a concise answer to the user a syntax tree is generated using the Stanford parser. In addition, FREyA assists the user to formulate a query through the generation of clarification dialogs; the user's selections are saved and used for training the system in order to improve its performance over time for all users. Similar to AquaLog's learning mechanism, FREyA uses ontology reasoning to learn more generic rules, which could then be reused for the questions with similar context (e.g., for the superclasses of the involved classes). Given a user query, the process starts with finding ontology-based annotations in the query, if there are ambiguous annotations that cannot be solved by reasoning over the context of the query (e.g., “Mississippi” can be a river or a state) the user is engaged in a dialog. The quality of the annotations depends on the ontology-based gazetteer OntoRoot, which is the component responsible for creating the annotations. The suggestions presented to the user in the clarification dialogs have an initial ranking based on synonym detection and string similarity. Each time a suggestion is selected by the user, the system learns to place the correct suggestions at the top for any similar question. These dialogs also allow translating any additional semantics into the relevant operations (such is the case with superlatives, which cannot be automatically understood without additional processing, i.e., applying a maximum or minimum function to a datatype property value). Triples are generated from the ontological mappings taking into account the domain and range of the properties. The last step is generating a SPARQL query by combining the set of triples.

Table 2.1. Ontology-based QA approaches classified by the subset of NL and degree of customization

Ontology-based QA systems	Subset of NL			Customization		Ontology-independent		
	Guided NL	Bag of words	Full / shallow grammar	Domain grammar/ collection	Domain lexicons	User learning	Relation (Triple) based	Pattern-matching (structured lexicon)
QACID		+		+	+			
ORAKEL			+	+	+			
e-Librarian			+		+			
GINSENG	+							+

NLPReduce		+						+
Querix			+				+	+
AquaLog			+			+	+	(entity lexicon)
PANTO			+				+	+
QuestIO		+					+	+ (Gazetteers)
FreyA			+			+	+	+ (Gazetteers)

We have selected a representative selection of state-of-the-art NL QA systems over ontologies to understand the advances and limitations in this area. However, this study is not exhaustive²⁰, and other similar systems to structured knowledge sources exist, such as **ONLI** (Mithun et al., 2007), a QA system used as front-end to the RACER reasoner. ONLI transform the user NL queries into a nRQL query format that supports the <argument, predicate, argument> triple format. It accepts queries with quantifiers and number restrictions. However, from (Mithun et al., 2007) it is not clear how much effort is needed to customize the system for different domains. (Dittenbach et al., 2003) also developed a NL interface for a Web-based **tourism platform**. The system uses an ontology that describes the domain, the linguistic relationships between the domain concepts, and parameterised SQL fragments used to build the SQL statements representing the NL query. A lightweight grammar analyses the question to combine the SQL statements accordingly. The system was online for ten days and collected 1425 queries (57.05% full input queries and the rest were keywords and question fragments). Interestingly, this study shows that the complexity of the NL questions collected was relatively low (syntactically simple queries combining an average of 3.41 concepts) and they can be parsed with shallow grammars.

Another approach with elaborated syntactic and semantic mechanisms that allows the user to input full NL to query KBs was developed by (Frank et al., 2006). **Frank et al. system** applies deep linguistic analysis to a question and transforms it into an ontology-independent internal representation based on conceptual and semantic characteristics. From the linguistic representation,

²⁰ For example, the EU funded project QALL-ME (Question Answering Learning technologies in a multiLingual and Multimodal Environment): <http://qallme.fbk.eu/>

they extract the so-called *proto queries*, which provide partial constraints for answer extraction from the underlying knowledge sources. Customization is achieved through hand-written rewriting rules transforming FrameNet like structures to domain-specific structures as provided by the domain ontology. A prototype was implemented for two application domains: the Nobel prize winners and the language technology domains, and was tested with a variety of question types (wh-, yes-no, imperative, definition, and quantificational questions), achieving precision rates of 74.1%.

To cope with the slower pace of increase in new knowledge in semantic repositories, in comparison with non-semantic Web repositories, **SemanticQA** (Tartir and Arpinar, 2010) makes it possible to complete partial answers from a given ontology with Web documents. SemanticQA assists the users in constructing an input question as they type, by presenting valid suggestions in the universe of discourse of the selected ontology, whose content has been previously indexed with Lucene. The matching of the question to the ontology is performed by exhaustively matching all word combinations in the question to ontology entities. If a match is not found, WordNet is also used. Then all generated ontological triples are combined into a single SPARQL query. If the SPARQL query fails, indicating that some triples have no answers in the ontology, the system attempts to answer the query by searching in the snippets returned by Google. The collection of keywords passed to Google is gathered from the labels of the ontological entities in the triples plus WordNet. The answers are ranked using a semantic answer score, based on the expected type (extracted from the ontology) and the distance between all terms in the keyword set. To avoid ambiguity it allows restricting the document search to a single domain (e.g., PubMed if the user is looking for bio-chemical information). A small-scale ad-hoc test was performed with only eight samples of simple factoid questions using the Lehigh University Benchmark ontology²¹ (63%

²¹ <http://swat.cse.lehigh.edu/projects/lubm/>

precision), and six sample queries using the SwetoDblp ontology (83% precision) (Aleman-Meza et al., 2007).

One can conclude that the techniques used to solve the lexical gap between the users and the structured knowledge are largely comparable across all systems: off-the-shelf parsers and shallow parsing are used to create a triple-based representation of the user query, while string distance metrics, WordNet, and heuristics rules are used to match and rank the possible ontological representations.

2.4.3 The performance of ontology-based QA systems

We examine the performance of the ontology-based QA systems presented in the previous section by looking at the evaluation results presented in the literature. Systematic and standard evaluation benchmarks to support independent datasets and comparisons between systems are not yet in place for semantic QA tools, therefore evaluations are generally small scale with ad-hoc tasks that represent the user needs and the system functionality to be evaluated (Uren et al., 2010), (Sure and Iosif, 2002), (McCool et al., 2005). Although the different evaluation set-ups and techniques undermine the value of direct comparisons, nevertheless they are still useful to do an approximate assessment of the strengths and weaknesses of the different systems. We hereby briefly describe the different evaluation methods and performance results. These are presented in Table 2.2.

Evaluations performed in the early days of the SW had to cope with the sparseness and limited access to high quality and representative public semantic data. As a result, to test the AquaLog system (Lopez et al., 2007a) two (manually built) ontologies were used and the query sets were gathered from 10 users. This approach fell short of providing a statistical sample over a wide range of users, but it gave a good insight about the effectiveness of the system and the extent to which AquaLog satisfied user expectations about the range of queries it is able to answer across two different domains. In order for an answer to be correct, AquaLog had to correctly align the vocabularies of both the asking query and the answering ontology. The test showed a 63.5% success,

a promising result considering that almost no linguistic restrictions were imposed on the questions. In accordance with the sequential nature of the AquaLog architecture, failures were classified according to which component caused the system to fail. The major limitations were due to lack of appropriate reasoning services defined over the ontology (e.g., temporal reasoning, quantifier scoping, negations -“not”, “other than”, “except”), comparatives and superlatives, a limited linguistic coverage (e.g., queries that are too long and need to be translated into more than two triples), and lack of semantic mechanisms to interpret a query given the constraints imposed by the ontology structures (e.g., AquaLog could not properly handle anaphoras²², compound nouns, non-atomic semantic relations, or reasoning with literals).

Alternatively, the evaluations presented in (Kaufmann, 2009) for NLP Reduce, Querix and Ginseng were measured by adapting standard IR performance metrics: precision and recall. Failures are categorized according to whether they are due to: 1) “no semantically tractable queries” (Tang and Mooney, 2001) (Popescu et al., 2003), i.e., questions that were not accepted by the query languages of the interfaces or 2) irrelevant SPARQL translations. Recall was defined as the number of questions from the total set that were correctly answered (% success), while precision is the number of queries that were correctly matched to a SPARQL query with respect to the number of semantically tractable questions (see Figure 2.3). Thus, the average recall values are lower than the precision values, a logical consequence of the fact that recall is based on the number of semantically tractable questions (those that the system can transform into one or more SPARQL queries, independently of whether the query produced is appropriate or not). For instance Ginseng has the highest precision but the lowest recall and semantic tractability due to its limited query language (some of the full NL test queries could not be entered into the system). Also, the use of comparative and superlative adjectives in many of the questions decreased the semantic tractability rate in NLP –

²² A linguistic phenomenon in which pronouns (e.g. “she”, “they”), and possessive determiners (e.g. “his”, “theirs”) are used to implicitly denote entities mentioned in an extended discourse (www.freepatentsonline.com/6999963.html).

Reduce, which cannot process them. To enable a comparison, these NLIs were benchmarked with the same three externally sourced test sets with which other NLI systems (PANTO by Wang et al. and the NLIDBs PRECISE by Popescu et al.) had already been evaluated. These three datasets are based on the *Mooney NL Learning Data* provided by Ray Mooney and his group from the University of Texas at Austin (Tang and Mooney, 2001) and translated to OWL for the purposes of the evaluation in (Kaufmann, 2009). Each dataset supplies a KB and set of English questions, belonging to one of the following domains: geography (9 classes, 28 properties and 697 instances), jobs (8 classes, 20 properties, 4141 instance) and restaurants (4 classes, 13 properties and 9749 instances).

$$\text{recall} = \frac{\text{number of correct SPARQL queries produced}}{\text{total number of questions}}$$

$$\text{precision} = \frac{\text{number of correct SPARQL queries produced}}{\text{number of semantically tractable questions}}$$

Figure 2.3. Definition of precision and recall by (Kaufmann, 2009)

PANTO assesses the rate of how many of the translated queries correctly represent the semantics of the original NL queries by comparing its output with the manually generated SPARQL queries. The metrics used are precision and recall, defined in (Wang et al., 2007) as “precision means the percentage of correctly translated queries in the queries where PANTO produced an output; recall refers to the percentage of queries where PANTO produced an output in the total testing query set”. Note that these definitions make the notion of correctness somewhat subjective, even between apparently similar evaluations. Recall is defined differently in PANTO and the approaches in (Kaufmann, 2009). For (Kaufmann, 2009) recall is the number of questions from the total correctly answered, which is defined as a *%success* in AquaLog, while for PANTO is the number of questions from the total that produce an output, independently of whether the output is valid or not. Thus, to measure success (how many NL question the system successfully transformed in SPARQL queries) in PANTO we need to multiply precision by recall and divide it by 100; the results are in Table 2.2.

There are also some discrepancies in the number of queries in the Mooney datasets between (Kauffman, 2009) and (Wang et al, 2007).

QuestIO was tested on a locally produced ontology, generated from annotated postings in the GATE mailing list, with 22 real user queries that could be answered in the ontology and a Travel Guides Ontology with an unreported number of queries, to demonstrate portability. The initialization time of QuestIO with the Travel Guides ontology (containing 3194 resources in total) was reported to be 10 times longer, which raises some concerns in terms of scalability. A query is considered correctly answered if the appropriate SeRQL query is generated (71.8% success).

FREyA was also evaluated using 250 questions from the Mooney geography dataset. Correctness is evaluated in terms of precision and recall, defined in the same way as in (Kaufmann, 2009). The ranking and learning mechanism was also evaluated, they report an improvement of 6% in the initial ranking based on 103 questions from the Mooney dataset. Recall and precision values are very high, both reaching 92.4%.

The system that reports the highest performance is the e-Librarian; in an evaluation with 229 user queries 97% were correctly answered, and in nearly half of the questions only one answer, the best one, was retrieved. Two prototypes were used: a computer history expert system and a mathematics expert system. The higher precision performance of e-Librarian with respect to a system like PANTO reflects the difficulty with precision performance on completely portable systems.

QACID has been tested with an OWL ontology in the cinema domain, where 50 users were asked to generate 500 queries in total for the given ontologies. From these queries, 348 were automatically annotated by an Entity Annotator and queries with the same ontological concepts were grouped together, generating 54 clusters that were manually associated to SPARQL queries. The results reported in an on-field evaluation, where 10 users were asked to formulate spontaneous queries about the cinema domain (a total of 100 queries), show an 80% precision.

As already mentioned, the different evaluation set-ups and techniques undermine the validity of direct comparisons, even for similar evaluations, like the ones between PANTO and the systems in (Kaufmann, 2009), because of the different sizes of the selected query samples and the different notions of evaluating correctness.

These performance evaluations share in common the pattern of being ad-hoc, user-driven and using unambiguous, relatively small and good quality semantic data. Although they test the feasibility of developing portable NLIs with high retrieval performance, these evaluations also highlight that the NLIs with better performance usually tend to require a degree of expensive customization or training. As already pointed out in (Damjanovic et al., 2008), to bridge the gap between the two extremes, domain independency and performance, the quality of the semantic data has to be very high, to ensure a good lexicalization of the ontology and KBs and a good coverage of the vocabulary. Nonetheless, as previously reported in AquaLog, and recently evaluated in FREyA, the inclusion of a learning mechanism offers a good trade-off between user interaction and performance, ensuring an increase in performance over time by closing the lexical gap between users and ontologies, without compromising portability.

Table 2.2: Performance of the semantic QA systems evaluated in the literature

	Datasets	N° queries	%Success (S) %Precision (P) %Recall (R)	Average (%S)	Domain independent
AquaLog	KMi portal ²³	69	58% (S)	63.5%(S)	Yes (NL queries)
	Wine and food ²⁴	68	69.11%(S)		
NLP Reduce	Geography	887	95.34%(P) 55.98% (R)(S)	55.3%(S)	Yes (NL and keyword queries)
	Restaurants	251	80.08%(P) 97.10% (R)(S)		
	Jobs	620	81.14%(P) 29.84% (R)(S)		
Querix	Geography (USA)	887	91.38%(P) 72.52% (R)(S)	54.4%(S)	Yes (NL wh-queries)
	Restaurants	251	94.31%(P)		

²³ The akt ontology: <http://kmi.open.ac.uk/projects/akt/ref-onto/>

²⁴ W3C, OWL Web Ontology Language Guide: <http://www.w3.org/TR/2003/CR-owl-guide-0030818/>

			59.36% (R)(S)		
	Jobs	620	80.25(P) 31.45% (R)(S)		
Ginseng	Geography (USA)	887	98.86%(P) 39.57% (R)(S)	48.6%(S)	Yes (guided interface)
	Restaurants	251	100%(P) 78.09% (R)(S)		
	Jobs	620	97.77%(P) 28.23% (R)(S)		
PANTO	Geography (USA)	877 out of 880	88.05%(P) 85.86%(R) 75.6% (S)	80%(S)	Yes (NL queries)
	Restaurants	238 out of 250	90.87%(P) 96.64%(R) 87.8% (S)		
	Jobs	517 out of 641	86.12%(P)/ 89.17%(R) 76.8% (S)		
ORAKEL	Geography (Germany)	454	93% (S)	93%(S)	Domain-dependent grammar (NL queries)
QuestIO	GATE ontology	22	71.88% (S)	71.8%(S)	Yes (NL queries)
	Travel guides	Not reported			
e-Librarian	Computer history and mathematics	229	97% (S)	97%(S)	Domain-dependent dictionary (NL queries)
QACID	Cinema	100	80% (S)	80%(S)	Domain-dependent collection NL queries
FREyA	Geography (USA)	250	92.4%(P) 92.4% (R)(S)	92.4%(S)	Yes

Finally, large ontologies pose additional challenges with respect to usability, as well as performance. The ontologies used in the evaluations are relatively small; allowing to carry out all processing operations in memory, thus, scalability is not evaluated.

2.4.4 The competence of ontology-based QA systems

The main clear advantage of the use of NL query tools is the easy interaction for non-expert users. As the SW is gaining momentum, it provides the basis for QA applications to exploit and reuse the structured knowledge available on the SW. Beyond the commonalities between all forms of QA (in particular for question analysis), in this section, we analyse the advantages of using an ontology to perform QA with respect to other forms of QA.

2.4.4.1 Ontology-based QA with respect to NLIDB

Since the development of the first QA systems (Androutsopoulos et al., 1995), there have been major improvements in the availability of lexical resources, such as WordNet; string distance metrics for name-matching tasks (Cohen et al., 2003); shallow, modular and robust NLP systems,

such as GATE (Cunningham et al., 2002); and NLP Parsers, such as the Stanford parser. In comparison with the latest work on NLIDB, the benefits of ontology-based QA are:

- **Ontology independence:** Since the 80s (Martin et al., 1985) (Copestake and Jones, 1990) NLIDB systems use intermediate representations to have a portable front end with general purpose grammars, while the back end is dependent on a particular database. As a result, long configuration times are normally required to port the system to a new domain. Ontology-based QA systems have successfully solved the portability problem, as the knowledge encoded in the ontology, together with (often shallow) domain-independent syntactic parsing, are the primary sources for understanding the user query, without the need to encode specific domain-dependent rules. Hence, these systems are practically ontology independent, less costly to produce, and require little effort to bring in new sources (AquaLog, PANTO, Querix, QuestIO, FREyA). Optionally, on these systems manual configuration or automatic learning mechanisms based on user feedback can optimize performance.
- **Able to handle unknown vocabulary in the user query:** NLIDB systems, such as PRECISE (Popescu et al., 2003), require all the tokens in a query to be distinct and questions with unknown words are not semantically tractable. In ontology-based QA if a query term is lexically dissimilar from the vocabulary used by the ontology, and it does not appear in any manually or automatically created lexicon, studying the ontology “neighbourhood” of the other terms in the query may lead to the value of the term or relation we are looking for. In many cases this would be all the information needed to interpret a query.
- **Deal with ambiguities:** When ontologies are directly used to give meaning to the queries expressed by the user and retrieve answers, the main advantage is the possibility to link words to obtain their meaning based on the ontological taxonomy and inherit relationships, and thus, to deal with ambiguities more efficiently.

Summing up, the main benefits of ontology-based QA systems are that they make use of the semantic information to interpret and provide precise answers to questions posed in NL and are able to cope with ambiguities in a way that makes the system highly portable.

2.4.4.2 Ontology-based QA with respect to open QA on text

Although most of the state-of-the-art of ontology-based QA still presumes that the knowledge needed is encoded in one ontology in a closed domain scenario, we envision ontology-based QA to move towards an open SW scenario (see Section 2.4.5), to become complementary to free-text open QA. While the first targets the open, structured SW to give precise answers, the second targets unstructured documents on the Web. Under such a perspective, a document search space is replaced by a semantic search space composed of a set of ontologies and KBs. Although linguistic and ambiguity problems are common in most kinds of NL understanding systems, building a QA system over the SW has the following advantages:

- **Balancing relatively easy design and accuracy:** As seen in Section 2.3.2 the current state of the art open systems to query documents on the Web require sophisticated syntactic, semantic and contextual processing to construct an answer, including NE recognition (Harabaigiu et al., 2000). These open QA systems classify queries using hierarchies of question types based on the types of answers sought (e.g., person, location, date, etc.) and filter small text fragments that contain strings with the same type as the expected answers (Moldovan et al., 1999) (Srihari et al., 2004). In ontology-based QA there is no need to build complex hierarchies, to manually map specific answer types to WordNet conceptual hierarchies or to build heuristics to recognize named entities, as the semantic information needed to determine the type of an answer is in the publicly available ontology (ies). As argued in (Mollá and Vicedo, 2007) a major difference between open-domain QA and ontology-based QA is the existence of domain-dependent information that can be used to improve the accuracy of the system.

- **Exploiting relationships for query translation:** NE recognition and IE are powerful tools for free-text QA (Section 3.2.1), but discovering relationships between entities is a crucial problem (Srihari et al., 2004) and one cannot always rely on WordNet coverage to determine the answer type or the type of the object of the verb in the question (Pasca, 2003). On the contrary, QA systems over semantic data can benefit from exploiting the ontological relationships and the explicit semantics of the ontology schema (type, subclassOf, domain and range), to understand and disambiguate a query. WordNet is only used for query expansion, to bridge the gap between the vocabulary of the user and the ontology terminology through lexically related words (such as synonyms).
- **Handling queries in which the answer type is unknown:** *What* queries, in which the type of the expected answer is unknown, are harder than other types of queries when querying free text (Hunter, 2000). However, the ontology simplifies handling what-is queries because the possible answer types are constrained by the types of the possible relations in the ontology.
- **Structured answers are constructed from ontological facts:** Arbitrary query concepts are mapped to existing ontology entities, answers are then obtained by extracting the list of semantic entities that comply with the facts, or fulfil the ontological triples (or SPARQL queries). The approach to answer extraction in text-based QA requires first identifying entities matching the expected answer in text, e.g., using the WordNet mapping approach. Second, the answers within these relevant passages are selected using a set of proximity-based heuristics, whose weights are set by a machine-learning algorithm (Pasca, 2003). Although IR methods scale well, valid answers in documents that do not follow the syntactic patterns expected by the QA system can be easily disregarded.
- **Combine multiple pieces of information:** Ontological semantic systems can exploit the power of ontologies as a model of knowledge to give precise, focused answers, where multiple pieces of information (that may come from different sources) can be inferred and combined

together. In contrast, QA systems over free text cannot do so, as they retrieve pre-written paragraphs of text or answer strings (typically NPs or named entities) extracted verbatim from text (Pasca, 2003).

2.4.4.3 Ontology-based QA with respect to commercial QA

It is costly to produce the large amounts of domain background knowledge, which are required by the proprietary open domain approaches described in Section 2.3.3. Although based on semantics, these systems do not reuse or take fully advantage of the freely available structured information on the SW. This is a key difference as they impose an internal structure on their knowledge and claim ownership of a trusted and curated homogeneous KB, rather than supporting the user in exploring the increasing number of distributed knowledge sources available on the Web.

2.4.5 Limitations of QA approaches in the context of the SW

Most of the semantic QA systems reviewed in this thesis are portable or agnostic to the domain of the ontology, even though, in practice they differ considerably in the degree of domain customization they require. Regardless of the various fine-grained differences between them, current ontology-aware systems suffer from the following main limitation when applied to a Web environment: *they are restricted to a limited set of domains*. Such domain restriction may be identified by the use of just one, or a set of, ontology(ies) covering one specific domain at a time, or the use of one large ontology which covers a limited set of domains. The user still needs to tell these systems which ontology is going to be used. For instance, in AquaLog the user can select one of the pre-loaded ontologies or load a new ontology into the system (to be queried the ontology is temporarily stored in a Sesame store in memory). Like in NLIDB, the key limitation of all the aforementioned systems is the one already pointed out in (Hirschman and Gaizauskas, 2001): these systems presume that the knowledge the system needs to answer a question is limited to the knowledge encoded in one, or a set of homogeneous ontologies at a time. Therefore, they are essentially designed to support QA in corporate databases or *semantic intranets*, where a shared organizational ontology (or a set of them) is typically used to annotate resources. In such a scenario

ontology-driven interfaces have been shown to effectively support the user in formulating complex queries, without resorting to formal query languages. However, these systems remain brittle, and any information that is either outside the semantic coverage, or simply not integrated with the corporate ontology remains out of bounds.

As a result, it is difficult to predict the feasibility of these models to scale up to open and heterogeneous environments, where an unlimited set of topics is covered. Nonetheless, we detail next the intrinsic characteristics of these systems, which in principle impair their suitability to scale up to the open SW in the large:

- **Domain-specific grammar-based systems:** In these systems grammars are used to syntactically analyse the structure of a NL query and interpret, if there are no linguistic ambiguities, how the terms in a query link to each other. According to (Copestake and Jones, 1990) it is difficult to devise grammars that are sufficiently expressive. Often, they are quite limited with regard to the syntactic structures they are able to understand or are domain dependent. Nevertheless, according to (Linckels and Meinel, 2006) users tend to use a limited language when interacting with a system interface, so grammars do not need to be complete. Systems like ORAKEL that involve the user in the difficult task of providing a domain-specific grammar are not a suitable solution in a multi-ontology open scenario.
- **Pattern-matching or bag-of-words approaches:** These systems search for the presence of constituents of a given pattern in the user query. As stated in (Kaufmann and Bernstein, 2007) “the more flexible and less controlled a query language is, the more complex a system’s question analysing component needs to be to compensate for the freedom of the query language”. However, naïve and flexible pattern-matching systems work well in closed scenarios, like the NLP-Reduce system, in which complexity is reduced to a minimum by only employing two basic NLP techniques: stemming and synonym expansion. Their best feature is that they are ontology independent and even ungrammatical and ill-formed questions can be

processed. Nevertheless, the lack of sense disambiguation mechanisms hampers their ability to scale to a large open scenario. In a non-trivial scenario, pattern-matching or bag-of-words approaches (QACID, QuestIO), together with the almost unlimited freedom of the NL query language, result in too many possible interpretations of how the words relate together. Thus, increasing the risk of not finding correct SPARQL-like translations and suffering from the habitability problem (Kauffman, 2009). As stated in an analysis of semantic search systems in (Hildebrand et al., 2007): “Naïve approaches to semantic search are computationally too expensive and increase the number of results dramatically, systems thus need to find a way to reduce the search space”.

- **Guided interfaces:** Guided and controlled interfaces, like GINO, which generates a dynamic grammar rule for every class, property and instance and presents pop-up boxes to the user to offer all the possible completions to the user’s query, are not feasible solutions in a large multi-ontology scenario. As stated in (Kaufmann, 2009) when describing GINO “It is important to note that the vocabulary grows with every additional loaded KB, though users have signalled that they prefer to load only one KB at a time”.
- **Disambiguation by dialogs and user interaction:** Dialogs are a popular and convenient feature (Kaufmann and Bernstein, 2007) to resolve ambiguous queries, for the cases in which the context and semantics of the ontology is not enough to choose an interpretation. However, to ask the user for assistance every time an ambiguity arises (AquaLog, Querix) can make the system not usable in a multi-domain scenario where many ontologies participate in the QA processes. In FREyA, the suggestions presented on the dialogs are ranked using a combination of string similarity and synonym detection with WordNet and Cyc²⁵. However, as stated in (Damjanovic et al., 2010): “the task of creating and ranking the suggestions before showing

²⁵ <http://sw.openencyc.org>

them to the user is quite complex, and this complexity arises [sic] as the queried knowledge source grows”.

- **Domain dependent lexicons and dictionaries:** High performance can be obtained with the use of domain dependent dictionaries at the expense of portability (as in the e-librarian system). However it is not feasible to manually build, or rely on the existence of domain dictionaries in an environment with a potentially unlimited number of domains.
- **Lexicons generated on demand when a KB is loaded:** The efficiency of automatically generating triple pattern lexicons when loading an ontology (PANTO, NLP-Reduce, QuestIO, FREyA), including inferred triples formed applying inference rules and WordNet lexically related words independently of their sense, decreases with the size of the ontology and is itself a challenging issue if multiple large-scale ontologies are to be queried simultaneously. In contrast with the structured indexes used by PANTO or NLP-Reduce, entity indexes can benefit from less challenging constraints in terms of index space, creation time and indexing maintenance. However, ignoring the remaining context provided by the query terms can ultimately lead to an increase in query execution time to find the adequate mappings.

2.5 Related work on user-friendly query interfaces for the SW

In the previous sections, we have seen that QA systems have proven to be ontology independent or easily adaptable to new domains, while keeping their efficiency and retrieval performance even when shallow NLP techniques are used. Despite the potential advantages of ontology-based QA applied to the SW scenario, in order to enhance and complement traditional forms of QA, these systems should overcome their main limitation, which is in essence the single-ontology assumption. In this section we broaden our scope and look at user-friendly semantic search systems and Linked Data querying interfaces, in search for models, beyond NL QA systems, that can in principle scale enough to open up, and even integrate, heterogeneous data sources on the Web of Data.

Many approaches exist to translate user queries into formal queries. Semantic search, a broader area than semantic QA, faces similar challenges to those tackled by QA systems when dealing with heterogeneous data sources on the SW. Here, we look at the solutions proposed in the literature for semantic search from early information systems to the latest approaches to searching the SW. We further discuss how all QA approaches presented till now and the SW user-friendly querying models presented in this section are compared according to the criteria presented in Section 2.2, and how both research directions can converge into large scale open ontology-based QA for the SW, to solve the bottlenecks and limitations of both.

2.5.1 Early global-view information systems

The idea of presenting a conceptually unified view of the information space to the user, the “world-view”, has been studied in (Levy et al, 1995). In early global information systems with well-defined boundaries, the solutions for interfacing and integrating heterogeneous knowledge sources, in order to answer queries that the original sources alone were unable to handle, are based on two approaches (Mollá and Vicedo, 2007): either all the information from multiple sources is extracted to create a unified database, or the set of databases can be seen as a federated database system with a common API, as in (Basili et al., 2004). However, this type of centralized solution that forces users and systems to subscribe to a single ontology or shared model are not transferable to the open-world scenario, where the distributed sources are constantly growing and changing. The manual effort needed to maintain any kind of centralized, global shared approach for semantic mapping is not only very costly, in terms of maintaining the mapping rules in a highly dynamic environment (Mena et al., 2000), but it also has the added difficulty of “negotiating” a shared model, or API, that suits the needs of all the parties involved (Bouquet et al., 2003).

Lessons and remaining open issues: Interestingly, the problems faced by these early information systems are still present nowadays. Linked Data assumes re-use of identifiers and the explicit specification of strong inter-dataset linkage in an open distributed fashion, without forcing users to

commit to an ontology. However, on the SW the heterogeneity problem can hardly be addressed only by the specification of mapping rules. As stated in (Polleres et al., 2010), “although RDF theoretically offers excellent prospects for automatic data integration assuming re-use of identifiers and strong inter-dataset linkage, such an assumption currently only weakly holds”. Therefore, open semantic applications need to handle heterogeneity and mappings on the fly, in the context of a specific task.

2.5.2 Evolution of Semantic Search on the Web of Data

Aiming to overcome the limitations of keyword-based search, semantic search has been present in the IR field since the eighties (Croft, 1986), through the use of domain knowledge and linguistic approaches (thesaurus and taxonomies) to expand user queries. Ontologies were soon envisaged as key elements to represent and share knowledge (Gruber, 1993) and enable a move beyond the capabilities of current search technologies (Guarino et al., 1999). As stated by (Fernandez et al., 2011) “the most common way in which semantic search has been addressed is through the development of search engines that execute a user query in the KB, and return tuples of ontology values which satisfy the user request”.

In the same line TAP (Guha et al., 2003), one of the first keyword-based semantic search system, presented a view of the search space where documents and concepts are seen as nodes in a semantic network. In TAP the first step is to map the search term to one or more nodes of the SW. A term is searched by using its `rdfs:label`, or one of the other properties indexed by the search interface. In ambiguous cases it chooses a search term based on the popularity of the term (frequency of occurrence in a text corpus), the user profile, the search context, or by letting the user pick the right denotation. The nodes that express the selected denotation of the search term provide a starting point to collect and cluster all triples in their vicinity (the intuition being that proximity in the graph reflects mutual relevance between nodes).

In 2004 the annual SW Challenge was launched, whose first winner was CS Aktive Space (Schraefel et al., 2004). This application gathers and combines a wide range of heterogeneous and distributed Computer Science resources to build an interactive portal. The top two ranked entries of the 2005 challenge, Flink (Mika, 2005) and Museum Finland (Hyvonen et al., 2005), are similar to CS Aktive Space as they combine heterogeneous and distributed resources to derive and visualize social networks and to expose cultural information gathered from several museums respectively. However, there is no semantic heterogeneity and “openness” in them: these tools simply extract information, scraped from various relevant sites, to populate a single, pre-defined ontology. A partial exception to this rule is Flink, which makes use of some existing semantic data, by aggregating online *FOAF* files.

Later semantic systems adopted interesting approaches to query interpretation, where keyword queries are mapped and translated into a ranked list of formal queries. These include SemSearch (Lei et al., 2006), XXPloreKnow! (Tran et al., 2007) and QUICK (Zenz et al., 2009). For instance, SemSearch supports the search for semantic relations between two terms in a given semantic source, e.g., the query ‘news:PhD students’ results in all instances of the class news that are related to PhD students. SemSearch and XXPloreKnow! construct several formal queries for each semantic relation or combination of keywords’ matches, where ranking is used to identify the most relevant meanings of keywords, and to limit the number of different combinations. To go beyond the expressivity of keywords and translate a keyword query into a set of semantic queries that are most likely to be the ones intended by the user, QUICK computes all possible semantic queries among the keywords for the user to select one. With each selection the space of semantic interpretations is reduced, and the query is incrementally constructed by the user.

The approach in (Fazzinga and Lukasiewicz, 2010) combines standard Web search queries with ontological search queries. It assumes that Web pages are enriched with annotations that have unique identifiers and are relative to an underlying ontology. Web queries are then interpreted based

on the underlying ontology, allowing the formulation of precise complex ontological conjunctive queries as SW search queries. Then these complex ontology queries are translated into sequences of standard Web queries answered by standard Web search. Basically, they introduce an offline ontological inference step to compute the completion of all semantic annotations, augmented with axioms deduced from the semantic annotations and the background ontologies, as well as an online step that converts the formal conjunctive ontological queries into semantic restrictions before sending them to the search engine.

Lessons and remaining open issues: As argued in (Motta and Sabou, 2006), the major challenge faced by early semantic applications was the lack of online semantic information. Therefore, in order to demonstrate their methods, they had to produce their own semantic metadata. As a result, the focus of these tools is on a single, well-defined domain, and they do not scale to open environments. The latest semantic applications, set out to integrate distributed and heterogeneous resources, even though these resources end up centralized in a semantic repository aligned under a single ontology. Therefore, these approaches follow the paradigm of smart KB-centred applications, rather than truly exploring the dynamic heterogeneous nature of the SW (Motta and Sabou, 2006). Furthermore, as discussed in (Fazzing et al., 2010), pressing research issues on approaches to semantic search on the Web are on the one hand, the ability to translate NL queries into formal ontological queries, and on the other hand, how to automatically add semantic annotations to Web content, or alternatively, extract knowledge from Web content.

2.5.3 Large scale Semantic Search and Linked Data interfaces

New technologies have been developed to manipulate large sets of semantic metadata available online. Search engines for the SW collect and index large amounts of semantic data to provide an efficient keyword-based access point and a gateway for other applications to access and exploit the growing SW. Falcons (Cheng et al., 2008) allows concept (classes and properties) and object (instance) search. The system recommends ontologies on the basis of a combination of the TF-IDF

technique and popularity for concept search, or the type of objects the user is likely to be interested in for object search. Falcons indexes 7 million of well-formed RDF documents and 4,400 ontologies (Cheng et al., 2008). Swoogle (Ding et al., 2005) indexes over 10,000 ontologies, Swoogle claims to adopt a Web view on the SW by using a modified version of the PageRank popularity algorithm, and by and large ignoring the semantic particularities of the data that it indexes. Later search engines such as Sindice (Oren et al., 2008) index large amounts of semantic data, over 10 billion pieces of RDF, but it only provides a *look-up* service that allows applications and users to locate semantic documents. Watson (D'Aquin et al., 2007) collects the available semantic content from the Web, indexing over 8,300 ontologies, and also offers an API to query and discover semantic associations in ontologies at run time, e.g., searching for relationships in specific ontological entities. Indeed out of these four ontology search engines, only Watson allows the user to exploit the reasoning capabilities of the semantic data without the need to process these documents locally, and therefore it can support systems which aim to exploit online ontologies in a dynamic way (D'Aquin, Motta et al., 2008).

Other notable exceptions to this limited-domain approach include search applications demonstrated in the Semantic Web Challenge competitions, and more recently the Billion Triples Challenge (btc)²⁶, aimed at stimulating the creation of novel demonstrators that have the capability to scale and deal with heterogeneous data crawled from the Web. Examples include SearchWebDB (Wang et al., 2008), the second prize-winner of the btc in 2008, which offers a keyword-based interface to integrated data sources available in the btc datasets. However, as keywords express the user needs imprecisely, the user needs to be asked to select among all possible interpretations. In this system the mappings between any pairs of data sources at the schema or data levels are computed a priori and stored in several indexes: the *keyword index*, the *structure index* and the *mapping index*.

²⁶ <http://challenge.semanticweb.org/>

The disadvantage being that, in a highly dynamic environment, static mappings and complex indexes are difficult to maintain, and the data quickly becomes outdated.

The eRDF infrastructure (Gueret et al., 2009) explores the Web of Data by querying distributed datasets in live SPARQL endpoints. The potential of the infrastructure was shown through a prototype Web application. Given a keyword, it retrieves the first result in Sindice to launch a set of SPARQL queries in all SPARQL end points, by applying an evolutionary anytime query algorithm, based on substitutions of possible candidate variables for these SPARQL queries. As such, it retrieves all entities related to the original entity (because they have the same type or a shared relationships to the same entity, for example Wendy Hall and Tim Berners Lee both hold a professorship at the university of Southampton).

Faceted views have been widely adopted for many RDF datasets, including large Linked Data datasets such as DBpedia, by using the Neofonie²⁷ search technology. Faceted views, over domain-dependent data or homogenous sources, improve usability and expressivity over lookups and keyword searches, although, the user can only navigate through the relations explicitly represented in the dataset. Faceted views are also available over large-scale Linked Data in Virtuoso (Erling and Mikhailov, 2009), however scalability is a major concern, given that faceted interfaces become difficult to use as the number of possible choices grows. The ranking of predicates to identify important facets is obtained from text and entity frequency, while semantics associated with the links is not explored. Similarly, Query Builder interfaces allow users to create complex queries to a KB by means of multiple triple patterns that follow the terminology and structure of the ontology (Auer and Lehmann, 2007).

Mash-ups (Tummarello et al., 2010) are able to aggregate data coming from heterogeneous repositories and semantic search engines, such as Sindice, however these systems do not

²⁷ <http://www.neofonie.de/index.jsp>

differentiate among different interpretations of the query terms, and disambiguation has to be done manually by the user.

Lessons and remaining open issues: these systems have the capability to deal with the heterogeneous data crawled from the Web. However, they have limited reasoning capabilities on the fly: mappings are either found and stored a priori (SearchWebdB), or disambiguation between different interpretations is not performed (eRDF). The scale and diversity of the data put forward many challenges, imposing a trade-off between the complexity of the querying and reasoning process and the amount of data that can be used. Expressivity is also limited compared to the one obtained by using query languages, which hinders the widespread exploitation of the data Web for non-expert users. Finally, in both facets and mash-ups, the burden to formulate queries is shifted from the system to the user. Furthermore, they do not perform a semantic fusion or ranking of answers across sources.

2.6 QA on the SW: achievements and research gaps

An overview of related work shows a wide range of approaches that have attempted to support end users in querying and exploring the publicly available SW information. It is not our intention to exhaustively cover all existing approaches, but to look at the state of the art and applications to figure out the capabilities of the different approaches, considering each of the querying dimensions presented in Section 2 (sources, scope, search environment and input), to identify promising directions towards overcoming their limitations and filling the research gaps.

2.6.1 Sources for QA and their effect on scalability.

We have shown through this paper that ontologies are a powerful source to provide semantics and background knowledge about a wide range of domains, providing a new important context for QA systems. We look at the capabilities of the different semantic query interfaces, to potentially scale to the Web of Data in its entirety and to the Web itself (see Table 2.3):

- Traditionally, the major drawbacks of intelligent NLIDB systems are that to perform both complex semantic interpretations and achieve high performance, these systems tend to use computationally intensive algorithms for NL processing (NLP) and presuppose large amounts of domain dependent background knowledge and hand-crafted customizations, thus being not easily adaptable or portable to new domains.
- Open QA systems over free text require complicated designs and extensive implementation efforts, due to the high linguistic variability and ambiguity they have to deal with to extract answers from very large open-ended collections of unstructured text. The pitfalls of these systems arise when a correct answer is unlikely to be available in one document but must be assembled by aggregating answers from multiple ones.
- Ontology-specific QA systems, although ontology-independent, are still limited by the single ontology assumption and they have not been evaluated with large-scale datasets.
- Commercial QA systems, although they scale to open and large scenarios in a potentially unlimited number of domains, cannot be considered as interfaces to the SW, as they use their own encoding of the sources. Nonetheless, they are a good example of open systems that integrate structured and non-structured sources, although, currently they are limited to Wikipedia (Powerset, TrueKnowledge) or a set of annotated documents linked to the KB (START).
- Although not all keyword-based and semantic search interfaces (including mash-ups and facets) scale to multiple sources in the SW, we are starting to see more and more applications that can scale, by accessing search engines, large collections of datasets (i.e., provided by the billion triple challenge), SPARQL endpoints, or various distributed online repositories (previously indexed). We have also seen an example of semantic search approaches (Fazzinga and Lukasiewicz, 2010) that can use semantic search to achieve high precision and recall on the Web. However this approach is limited by the single-ontology assumption and it is based

on the assumption that documents in the Web are annotated. In (Fazzinga and Lukasiewicz, 2010) conjunctive semantic search queries are not formulated yet in NL and logical queries need to be created according to the underlying ontology, thus making the approach inaccessible for the typical Web user. DBpedia has also been used as a source for a query completion component in normal Web queries on the mainstream Yahoo search engine (Meij et al., 2009). However, the results of a large scale evaluation, based on query logs, suggested that the most common queries were not specific enough to be answered by factual data and therefore factual information may only address a relatively small portion of the user information needs. The current implementation based on a large but single dataset suffer from the knowledge incompleteness and sparseness problems.

Notwithstanding, we believe that the development of open semantic ontology-based QA systems can potentially fill the gap between closed domain QA over structured sources (NLIDB) and domain independent QA over free text (Web), as an attempt to enhance the limitations of currently ontology-specific QA and search interfaces and perform QA in open domain environments by assembling and aggregating answers from multiple sources. The trade-off is that in a highly formalized and domain-specific scenario answers can be derived with a high degree of accuracy, while systems that aim to perform at Web scale and interface highly heterogeneous retrieve the most likely answers, ranking heterogeneous results in case of ambiguity (Madhavan et al., 2007).

2.6.2 Beyond the scope of closed-domain QA

One main dimension over which these approaches can be classified is their scope. On a first level we can distinguish the **closed domain approaches**, whose scope is limited to one (or a set of) a-priori selected domain(s) at a time. As we have seen, ontology-based QA systems, which give meaning to the user queries with respect to the domain of the underlying ontology, although portable or ontology-independent they are still limited to a domain specific scenario at a time.

On a second level, and enhancing the scope embraced by closed domain models, we can distinguish those **approaches restricted to their own semantic resources**. While successful NL search interfaces to structured knowledge in an open domain scenario exist (popular examples are Powerset, Wolfram Alpha, or TrueKnowledge), they are restricted to the use of their own semi-automatically built and comprehensive factual knowledge bases. This is the most expensive scenario as they are typically based on data that are by and large manually coded and homogeneous.

On a third level, we can highlight the latest **open semantic search approaches**. These systems are not limited by closed-domain or homogeneous scenarios, neither by their own resources, but provide a much wider scope, attempting to cover and reuse the majority of publicly available semantic knowledge. We have seen in Section 2.5 examples of these different approaches: a) using Linked Data sources, i.e., DBpedia, for a query completion component on the Yahoo search engine, b) keyword-based query interfaces to data sources available in the billion triple challenge datasets and live SPARQL endpoints, c) mash-ups able to aggregate heterogeneous data obtained from the search engine Sindice from a given keyword and d) facets, which allow the user to filter objects according to properties or range of values.

We can see that there is a continuous tendency to move towards applications that take advantage of the vast amount of heterogeneous semantic data and get free of the burden of engineering their own semantic data (D'Aquin, Motta et al., 2008). Nevertheless, the major drawback of current ontology-based QA systems is that although portable, the scope of these systems is limited to the amount of knowledge encoded in one ontology. As such, the next key step towards the realization of QA on the SW is to move beyond domain specific QA to robust open domain QA over structured and distributed semantic data. In addition, given that it is often the case that queries can only be solved by composing information derived from multiple and autonomous information sources, portability alone is not enough and openness is required.

2.6.3 Issues associated with performing QA in open and dynamic environments

A new layer of complexity arises when moving from a classic KB system to an open and dynamic search environment. If an application wishes to use data from multiple sources the integration effort is non-trivial.

While the latest open Linked Data and semantic search applications shown in 2.5.3 present a much wider scope, scaling to the large amounts of available semantic data, they perform a shallow exploitation of this information: 1) they do not perform semantic disambiguation, but need users to select among possible query interpretations, 2) they do not generally provide knowledge fusion and ranking mechanisms to improve the accuracy of the information retrieved, and 3) they do not discover mappings between data sources on the fly, but need to pre-compute them beforehand.

Automatic disambiguation (point 1) can only be performed if the user query is expressive enough to grasp the conceptualizations and content meanings involved in the query. In other words, the context of the query is used to choose the correct interpretation. If the query is not expressive enough, the only alternative is to call the user to disambiguate, or to rank the different meanings based on the popularity of the answers.

With regards to fusion (point 2) only mash-ups aggregate answers across sources. However, so far, mash-ups do not attempt to disambiguate between different interpretations of a user keyword. Ranking techniques are crucial to scale to large-scale sources or multiple sources.

With regards to on the fly mappings (point 3), only the very latest semantic search systems analysed here perform mappings on the fly given a user task, and some of them are able to select the relevant sources on the fly. There are three different mechanisms which are employed: (1) through search engines (mash-ups); (2) by accessing various distributed online SPARQL end-points providing full text search capabilities (Linked Data facets); (3) by indexing multiple online repositories (semantic search).

Although ontology-based QA can use the context of the query to disambiguate the user query, it still faces difficulties to scale up to large-scale and heterogeneous environments. The complexity arises because of its “openness”, where systems face the problem of polysemous words, which are usually unambiguous in restricted domains. At the same time, open-domain QA can benefit from the size of the corpus: as the size increases it becomes more likely that the answer to a specific question can be found without requiring a complex language model. As such, in a large-scale open scenario the complexity of the tools will be a function of their ability to make sense of the heterogeneity of the data to perform a deep exploitation beyond simple lookup and mash-up services.

In Table 2.3 we compare how the different approaches to query the SW, tackle these traditional research issues derived from the openness of the search environment (automatic disambiguation of user needs, ranking, portability, heterogeneity and fusion across sources)

Table 2.3. Querying approaches classified according to their intrinsic research problems and search criteria

Criteria	Input		Scope			Search environment (research issues)				Sources	
	NL Expressivity	Reasoning services	Portability	Open Domain	Heterogeneity	Ranking	Disambiguation	Fusion	Sources on-the-fly	Scale SW	Scale Web
NLIDB	√	√	∅	∅	∅	∅	√	∅	∅	∅	∅
QA-Text	√	∅	√	√	√	√	√	∅	√	∅	√
Ontology-QA	√	√	√	∅	∅	+/-	√	∅	∅	+/-	∅
Proprietary QA	√	√	√	√	∅	√	√	∅	∅	∅	+/-
Keyword-search	+/-	∅	√	√	√	√	+/-	∅	√	√	+/-
Mashups	∅	∅	√	√	√	+/-	∅	√	√	√	∅
Facets	√	∅	√	√	√	√	∅	∅	∅	√	∅

2.6.4 Input and higher expressivity

Finally, the expressivity of the user query is defined by the **input** the system is able to understand. As shown in Table 2.3, keyword-based systems lack the expressivity to precisely describe the user’s intent, as a result ranking can at best put the query intentions of the majority on top. Most approaches look at expressivity at the level of relationships (factoids), however, different systems provide different support for complex queries, from including reasoning services to understand

comparisons, quantifications and negations, to the most complex systems (out of the scope of this review) that go beyond factoids and are able to understand anaphora resolution and dialogs (Basili et al., 2007). Ontologies are a powerful tool to provide semantics, and in particular, they can be used to move beyond single facts to enable answers built from multiple sources. However, regarding the input, ontologies have limited capability to reason about temporal and spatial queries and do not typically store time dependent information. Hence, there is a serious research challenge in determining how to handle temporal data and causality across ontologies. In a search system for the open SW we cannot expect complex reasoning over very expressive ontologies, because this requires detailed knowledge of ontology structure. Complex ontology-dependant reasoning is substituted by the ability to deal and find connections across large amounts of heterogeneous data.

2.7 Directions ahead: the contribution of this thesis

Through efforts such as the Linked Open Data initiative, the Web of Data is becoming a reality, growing and covering a broader range of topics. Novel approaches that can help the typical Web user to access the open, distributed, heterogeneous character of the SW and Linked Data are needed to support an effective use of this resource.

Scalability is the major open issue and a study presented in (Lee and Goodwin, 2005) about the potential size of the SW reveals that the SW mirrors the growth of the Web in its early stages. Therefore, semantic systems should be able to support large-scale ontologies and repositories both in terms of ontology size and the number of them. Semantic search technologies that have been proven to work well in specific domains still have to confront many challenges to scale up to the Web in its entirety. The latest approaches to exploit the massive amount of distributed SW data represent a considerable advance with respect to previous systems, which restrict their scope to a fraction of the publicly available SW content or rely on their own semantic resources. These approaches are ultimately directed by the potential capabilities of the SW to provide accurate responses to NL user queries.

The aim behind this thesis is to go beyond the state of the art of current QA systems and user-friendly semantic querying approaches that either restrict their scope to homogenous and domain-specific content, or perform a shallow exploitation of it, by developing a new open QA system that integrates ideas from traditional QA research into scalable SW tools to support users in querying and exploring the heterogeneous SW content. In particular we aim:

- a) To bridge the gap between the end-user and the SW by providing a NL interface that can scale up to the Web of Data.
- b) To take advantage of the structured information distributed on the SW to retrieve aggregated answers to factual queries that extend beyond the coverage of single datasets and are built across multiple ontological statements obtained from different sources. Consequently, we aim to smooth the habitability and brittleness problems intrinsic to closed domain systems.

In this thesis, our interest lies on factual QA (such as in TREC) over semantic data distributed across multiple sources. The major challenge is the combination of scale with the considerable heterogeneity and noise intrinsic to the SW. As stated in (Motta and Sabou, 2006) when operating at scale on a large and distributed SW, then it becomes much more difficult, if not impossible, ensuring strict data quality.

In these factual systems the lack of very complex reasoning is substituted by the ability to deal and find connections in large amounts of heterogeneous data, to provide coherent answers within a specific context/ task. As a consequence, for QA, exploiting the SW is by and large about discovering interesting connections between items. In any case this is unlikely to provide a major limitation given that, most of the large datasets published in Linked Data are light-weight.

Furthermore, besides scaling up to the SW in its entirety, we still have to bridge the gap between the semantic data and unstructured textual information available on the Web. Ultimately, complementing the structured answers from the SW with Web pages will enhance the expressivity and performance of traditional search engines with semantic information.

Research on open QA over semantic sources builds on top of the results obtained from other research areas, like ontology selection and ranking (which sources better satisfy a user queries), ontology mapping (e.g. “car” maps to “auto” in one ontology and to “vehicle” in another), word sense disambiguation (“squash” can be a sport or a vegetable), and co-reference of instances, applied to a dynamic task context given by the user needs. In the next chapter we will introduce the requirements and research challenges we aim to address the limitations of current approaches.

Chapter 3 Challenges and Requirements

The challenges and requirements that PowerAqua imposes on ontology mapping and related research areas are presented in this chapter, parts of which have been presented at the International Semantic Web Conference in 2006 (Lopez et al., 2006b).

3.1 Introduction: PowerAqua research challenges

As the amount of semantic data available online increases, ontology mapping plays an increasingly important role in bridging the semantic gap between distributed and heterogeneous data sources. The importance of mapping for the Semantic Web (SW) has been widely recognized (Shvaiko and Euzenat, 2005) and a range of techniques and tools has been developed. However, the predominant view of mapping is that it will be performed during the development of the application, e.g., when deciding on mapping rules between a set of ontologies (Bouquet et al., 2003). This was a plausible assumption because, until recently, only a limited amount of semantic data was available; therefore, there was little need for run time integration. Indeed, one of the main characteristics of SW based applications built so far is that they tackle the data heterogeneity problem in the context of a given domain or application, by integrating a few, a-priori determined sources (Hyvonen et al., 2005) (Mika, 2005). However, we are now moving away from the early applications characterized by limited heterogeneity to applications that explore the dynamic and heterogeneous nature of the SW, depending on their current information need (Motta and Sabou, 2006).

Among this new generation of tools, PowerAqua adopts an open QA strategy by consulting and aggregating information derived from multiple heterogeneous ontologies on the web to answer a user query. To extend the ontology-based query capabilities to several sources of information, PowerAqua needs in essence to solve the following challenges:

- *Finding the relevant ontologies to answer the user's query.* In an open domain scenario it is not possible to determine in advance which ontologies will be relevant to answer the user's information needs.

- *Identifying semantically sound mappings.* User queries can be mapped over several ontologies. In the case of ambiguity, the correct interpretation of the given term in the context of the user query should be returned.
- *Composing heterogeneous information on the fly.* Answering queries may require aligning (merging) and fusing information from multiple sources. Composite translations and partial answers from different ontologies need to be combined and ranked to fully translate a user query and retrieve accurate results. Answers distributed across multiple sources, which require fusion techniques that combine partial answers from different sources, should be coherent (Burger et al., 2002).
- *Giving answers in real time.* This requirement must be satisfied regardless of the complexity and ambiguity of the question, or the size and multitude of the data sources.

Prior to illustrating the design of the PowerAqua architecture in Chapter 4, here, we first discuss the state of the art in ontology mapping and related research areas, such as Word Sense Disambiguation (WSD) and semantic similarity measures (Section 3.2), which are relevant to addressing the PowerAqua challenges. Second, we present a novel perspective about the general requirements that tools like PowerAqua impose on mapping and related areas (Section 3.3) and we then describe our proposed approach for a run time mapping algorithm that complies with the given requirements (Section 3.4). Thus, this chapter describes the context in which our mapping algorithm, PowerMap (described in Chapter 5), which is a core component of PowerAqua, was conceived. Scalability issues, which are emerging with the appearance of open, large-scale content in Linked Data, are further discussed in Chapter 11.

3.2 Traditional intrinsic problems in QA: heterogeneity

Here, we review the state of the art in the research areas relevant to ontology-based QA.

3.2.1 State of the art on ontology mapping

For SW applications to take advantage of the vast amount of heterogeneous semantic data made available by the growth of the SW, they need to concentrate on meaningfully finding the relevant ontologies and mappings. Robust mechanisms for ontology selection and matching are crucial to support knowledge reuse in this large-scale, open environment. Ontology selection algorithms in the literature rely on ranking measures such as connectedness and popularity (ontology imports), or structure-based metrics such as compactness and density. More details are given in the study presented in (Sabou et al., 2006b). An analysis of mapping systems is presented in (Shvaiko and Euzénat, 2005). Basically, current approaches to ontology mapping combine a range of:

- Lexical non-semantic techniques that exploit string similarity between meaningful labels.
- Structural techniques that rely on the structure of the mapped ontologies.
- Instance-based techniques that map concepts on the basis of shared instances.
- Background knowledge approaches that use external sources as an oracle (WordNet, high level, or domain dependent ontologies) to identify mappings which cannot be found with the above techniques.

The majority of approaches to ontology mapping use a combination of lexical and structural methods, where lexical overlap is used to produce an initial mapping that is subsequently improved by the structure of the source and the target. For instance, Falcon-AO (Jian et al., 2005) outperformed all other ontology matchers in the 2005 Ontology Alignment Contest (OAC) (Euzénat et al., 2005). Falcon-AO regards ontologies as graph-like structures to produce mappings by using both linguistic and structural similarity. In many cases string and structural similarities can imply meaningful mappings but also, as observed to some extent in the OAC-05, traditional methods fail when there is little overlap between the labels of the ontology entities, or when the ontologies have weak or dissimilar structures.

A different approach is to use background knowledge sources, which can help to identify mappings between lexically dissimilar items and to disambiguate polysemous terms (e.g., Turkey may be related to food in a domain about food but not in a domain about countries). For example, in SMatch (Giunchiglia et al., 2004) predefined sources like WordNet and reference domain ontologies have been used as an oracle for background knowledge. SMatch translates ontology labels into a logical formula between their constituents and maps those into corresponding WordNet senses, where a *SAT* solver is then used to derive semantic mappings. Other approaches (Aleksovski et al., 2006) (Stuckenschmidt et al., 2004) have considered the use of external background knowledge and rely on a reference domain ontology as a way to obtain semantic mappings between syntactically dissimilar ontologies. In (Aleksovski et al., 2006) terms from two vocabularies are first mapped to so called anchor terms in the reference domain ontology (DICE), and then their mapping is deduced based on the semantic relation between the anchor terms.

However, as stated in (Sabou et al., 2006a) obtaining the right background knowledge is problematic, because it requires axiomatized domain ontologies (Aleksovski et al., 2006) that: (1) do not exist in all domains, (2) are unlikely to cover all intended mappings between the input ontologies, and (3) have to be selected a priori, not in real time. In (Van Hage et al., 2005) online textual resources (i.e., Google) are used as a source of background knowledge, so there is no need for a manual and domain dependent ontology selection task prior to mapping, but the drawback is that the required knowledge extraction techniques lead to considerable noise and human validation is needed.

A more suitable approach to overcome the limitation of syntactic approaches and obtain semantic relations between dissimilar ontologies is used in Scarlet (Sabou et al., 2008), where ontology mapping exploits the heterogeneity of the SW and uses it as a background knowledge source. Using semantic data as sources of background knowledge is likely to be less noisy than deriving it from textual sources and therefore leads to the discovery of better mappings. Scarlet takes two candidate words as input, then the Watson search engine is used to find ontologies containing concepts with

the same names as the candidate words, and to derive mappings from the taxonomical relationships between them in the selected ontologies. If there is not a single ontology that relates them together then Scarlet attempts to identify relations across ontologies. The main limitations of this approach are that the terms have to be found as classes (or properties) in the pool of ontologies in the first place, by using string matching techniques, and second, that it can suffer from the *knowledge sparseness phenomenon*: some domains are well covered by existing ontologies (e.g., academic research and medicine), while others are not covered at all.

With the exception of dynamic approaches, such as the one presented in (Sabou et al., 2008), the predominant view of mapping is that it will be performed offline, in the context of domain-specific applications characterized by a limited degree of heterogeneity. Approaches that analyse all the ontology concepts in order to obtaining mappings for each pair of concepts belonging to different ontologies are not scalable. (Madhavan et al., 2001) analyses the factors that affect the effectiveness of algorithms for automatic semantic reconciliations, stating that the level of effort is at least linear in the number of matches to be performed. Run time approaches targeted to integrate information from multiple heterogeneous large sources should be driven by the task, without requiring the construction, a priori, of a shared ontology that defines mappings between ontologies of different origin. The requirements on mapping and further advances imposed by this open setting are analysed in Section 3.3.

3.2.2 State of the art on semantic similarity and WSD

Because similarly spelled words (labels) may have not precisely matched meanings, relationships between word senses, not words, are needed. WSD is measured through the notion of *similarity* (Resnik, 1995). Similarity is a more specialized notion than *association or relatedness*. Similar entities are semantically related by virtue of their similarity (bank-trust company). Dissimilar entities may also be semantically related by lexical relationships such as meronym (*car-wheel*) and antonymy (*hot-cold*), or just by any kind of functional relationship or frequent association (*doctor-hospital*, *penguin-Antarctica*) (Budanitsky and Hirst, 2006).

Many similarity measures and strategies exist in the literature for WSD (Ide and Veronis, 1998). The majority of proposals for semantic distance and similarity measures are based on the following underlying approaches (Bernstein et al., 2005): (1) ontology-based measures which reflect the closeness of two objects in a taxonomy, (2) information theory entropy measures, which require a probabilistic model of the application domain, and (3) vector space models, typically the cosine measure, and string distance metrics, such as Levenshtein and Jaccard (Cohen et al., 2003).

Here, we look at ontology-based approaches. The most intuitive similarity measure between concepts in an ontology is their distance within the ontology following the number of IS_A relations between them, so that the shorter the path between two terms the more similar they are (Resnik, 1995). However, a widely acknowledged problem is that the approach relies on the notion that links in the taxonomy represent uniform distances, but as (Budanitsky and Hirst, 2006) stated typically this is not true and there is a wide variability in the “distance” covered by a single taxonomic link. Resnik (Resnik, 1995) established a second criterion of similarity between two concepts that is the extent to which they share information in common, which, in an IS-A taxonomy, can be determined by inspecting the relative position of the most-specific concept that subsumes them both (*lowest subsume object -lso*). The number of links (depth) is still important to distinguish between any two pairs of concepts having the same lso. One of the variations of this edge-counting method is the conceptual similarity introduced by Wu & Palmer (Wu and Palmer, 2004):

$$Similarity(C_1, C_2) = \frac{2 \times N}{(N_1 + N_2 + (2 \times N))}$$

Where N_i = length of the path from C_i to C ; N = length of path from C to root; C = lso (C_1, C_2)

Researchers in the NLP domain commonly measure both similarity and relatedness between two concepts by using WordNet as the reference ontology and source of lexical and domain knowledge (Budanitsky and Hirst, 2006). In WordNet nouns, verbs, adjectives, and adverbs are each organized into networks of synonyms sets (*synsets*). There are nine types of semantic relations defined on the noun subnetwork: hyponymy (IS-A) relation, and its inverse hypernymy; six meronymic (PART-

OF) relations – COMPONENT-OF, MEMBER-OF, SUBSTANCE-OF and their inverses; and the COMPLEMENT-OF relation.

WSD is also applied to disambiguate terms in a sentence. Most approaches assume that words that appear together in a sentence can be disambiguated by assigning to them the senses that are most closely related to their neighbouring words (Banerjee and Pedersen, 2003). Pedersen and his colleagues have made available a Perl implementation of six WordNet measures evaluated in (Budanitsky and Hirst, 2006)²⁸, plus their own semantic relatedness measure based on the number of shared words in the WordNet glosses (the definitions associate to each synset) to assign a meaning to every content word in a text. However, as argued in (Budanitsky and Hirst, 2006), WSD methods based on WordNet are useful computational methods only for quantifying semantic distances for similarity relationships. But, relatedness includes not just the WordNet relationships but also *associative* and *ad hoc* relationships. These can include just about any kind of functional relation or frequent association in the world (e.g., bed-sleep), sometimes constructed in the context, and cannot always be determined purely from *a priori* lexical resources such as WordNet.

In addition to that, frequently cited WordNet limitations are (Gracia et al., 2006): (1) very limited coverage and (2) rather static set of senses, e.g., “apple” as a “computer” does not appear in WordNet 3.0. To overcome WordNet’s limitations, the approach introduced by (Gracia et al., 2006) consists of using not only WordNet but also the whole SW as background knowledge, where different ontologies represent different views of the world and therefore they can be exploited jointly to gather a much larger set of senses. The authors propose a multi-ontology based method to disambiguate the senses of keywords used in a search engine, e.g., in “astronomy, star, planet”, “star” is used in its sense of celestial body. In Gracia et al.’s approach online ontologies are dynamically identified as sources for candidate word senses through Swoogle, and then semantic

²⁸ Basically, these measures look for a path connecting a synset associated with each word.

measures between these senses are computed, by employing algorithms that combine information available both on the selected ontologies and the web, to complete their disambiguation.

To summarize, the general approach to run time WSD in the SW has been to syntactically anchor the terms onto a single shared hierarchy, such as WordNet. Then, semantic similarity is determined as a function of the path distance between terms in the hierarchy of the underlying single ontology. In a closed domain scenario, where it is expected that most polysemous terms have a well-defined meaning, ambiguity in anchoring is a minor problem (because matched and background ontologies share the same domain). On the contrary, in the case of using online ontologies in an open and heterogeneous scenario, ambiguity becomes relevant, as discussed in (Gracia et al., 2006). Further work is needed to extrapolate these techniques to extract the similarity across concepts in different ontologies in an open scenario and at run time.

3.3 Requirements for PowerAqua in the context of the SW

Core to the information integration tasks that can be supported by SW technology are algorithms that allow matching between the elements of several, distributed ontologies. As we have seen throughout this chapter, the problem of ontology schema mapping has been investigated by many research groups, which have proposed a large variety of approaches (Shvaiko and Euzenat, 2005) (Rahm and Bernstein, 2001). While all this research has produced increasingly complex algorithms, the setting in which the mapping problem was tackled was almost always the same: given two ontologies, find all the possible mappings between their entities attaching a confidence level to the mappings that are returned. One of the challenges in the field of ontology mapping now is not so much perfecting these algorithms, but rather trying to adapt them to novel scenarios, which require SW applications to automatically select and integrate semantic data available online. The setting in which the mapping would take place is quite different from the “traditional” ontology mapping scenario. Unlike the case of earlier tools, where mapping has been performed at design time, PowerAqua requires mapping techniques that can be performed at run time. Indeed, the focus is not

on mapping complete ontologies but rather small snippets that are relevant for a given query. These new settings impose a number of requirements:

1) Access to multiple distributed ontologies – when integrating data from online ontologies it is often necessary to map between several online ontologies. This is different from the traditional scenario where only two ontologies are considered.

2) Increased heterogeneity – traditional mapping techniques often assume that the ontologies to be matched will be similar in structure and describe more or less the same topic domain. For example, S-Match (Giunchiglia et al., 2004) is targeted towards matching classification hierarchies, whereas due to its structure based techniques, Anchor-PROMPT (Noy and Musen, 2001) works best if the matched ontologies have structures of similar complexity. Such similarity assumptions fail on the SW: we cannot predict whether relevant information will be provided by a simple FOAF file or by WordNet, or top level ontologies, or combined from these different sources. Mapping techniques should function without any pre-formulated assumptions about the ontological structure.

3) Time performance is important - As already pointed out in (Ehrig and Staab, 2004), the majority of mapping approaches focus on the effectiveness (i.e., quality) of the mapping rather than on its efficiency (i.e., speed). In contrast with these design-time scenarios, the speed of the response is a crucial factor in a QA scenario, where mappings have to be established at run time.

4) Consider relation and instance mappings – much of the work in ontology mapping has focused on matching the concepts in two schemas, while other ontology entities, such as relations and instances, have largely been ignored so far (although relations and instances are taken into account as evidence to support the matching process in some approaches). However, SW tools are often used to find out information about specific entities (traditionally modelled as ontology instances), as well as the relations between entities. Therefore, mapping techniques should be developed to efficiently match these kinds of entities too.

5) Cross-ontology mapping filtering - several approaches adopt the model of first generating all possible mappings and then filtering the relevant ones. However, in these approaches mappings are typically created between two ontologies describing the same domain. When performing mappings on the SW, we are also likely to discover several mappings but this time the mapping candidates might be drawn from different ontologies in different domains. Therefore domain-independent methods able to reason about ontologies that may only have very few concepts in common are needed. This requires mechanisms to assess whether or not such ‘sparse concepts’ are related.

6) Results should be ranked with respect to their relevance within a specific task – for systems that aim to perform at Web scale and perform QA on highly heterogeneous data, which can potentially yield a large number of results, ranking is necessary to retrieve the most relevant answers for the user task in real time, and in case of ambiguity to rank the answers according to their degree of relevance. Context can be used to clarify a question and resolve ambiguities.

7) Distributed results should be coherent - answers distributed across multiple sources, which require fusion techniques that detect similar answers and combine partial answers from different sources, should be coherent (Burger et al., 2002).

8) Leverage noise and incomplete semantic data – in an open scenario, systems should be able to handle semantic data that is inconsistent, noisy, contains modelling errors or it is incomplete (i.e., lack of schema information). The breadth and variety of data on the Web of Data makes inconceivable any manual effort to create a clean and complete model of the data or to rely on building a model of the data (mediated schema) ahead of time.

3.4 The approach of PowerAqua at a glance

These requirements imposed by the SW applications that need to harvest the rich ontological knowledge on the web, are exemplified in the context of PowerAqua. Unlike traditional mapping algorithms, PowerMap is focused towards dealing with several, heterogeneous ontologies which are

not given a priori but rather discovered depending on the context of the user's query (thus it fulfils **requirements 1 and 2**), and without pre-formulated assumptions about the structure or domain of the relevant ontologies. Thus, in PowerAqua the translation process is driven by the task that has to be performed, more concretely by the query asked by the user. Indeed, this is novel in comparison with traditional approaches where mappings are done prior to the ontology being used for a specific task.

We envision a scenario where a user may need to interact with thousands of KBs structured according to hundreds of ontologies. However, good performance could be obtained also at such scale, as requested by our **requirement 3**, by:

- Using an infrastructure based on indexing mechanisms and storage technologies developed by the SW community for processing SW information, in order to scale an enormous amount of data and provide support to respond to queries quickly.
- Avoiding a global interpretation of the mapped ontologies, in which the level of effort is at least linear in the number of matches to be performed (Giunchiglia et al., 2004). Only concepts relevant to the user's query should be analysed, retrieving the most relevant answers ranked according to the confidence and quality on the mappings in this open scenario, where data quality cannot be assured due to noise, inconsistency and modelling errors (**requirement 6 and 8**).

An efficient algorithm should employ steps that are increasingly complex (so that the most-time consuming tasks are the last ones to run when the system has already filtered out a small set of ontologies), starting with syntactic mappings that take into account entity labels and lexically related words to find potentially useful ontologies (considering concepts, relations, instances and literals, following the criteria defined by **requirements 4**). Then, once a collection, usually rather large, of potential candidates is found, it relies, e.g., on WordNet or ontological information to extract the meaning of the proposed mappings, and to verify that the mappings are semantically sound

according to the user query (**requirement 5**). After that, in the cases where an answer may require integrating a number of statements, answers should be fused and ranked across ontologies (**requirement 7**).

3.5 Summing up

QA considered, until now, two main branches, depending on the sources used to generate an answer: closed domain QA over structured data (databases, KBs, etc.) and open domain QA over free text. As seen in the literature review in Chapter 2 recent work on closed-domain QA over structured data has addressed successfully key issues such as portability. QA over free text has developed sophisticated, syntactic, semantic and contextual processing to answer queries in an open-domain scenario. However, the pitfalls of QA over free text arise when a correct answer is unlikely to be extracted verbatim from relevant text passages available in one document (Pasca, 2003), but must be assembled by aggregating answers from the same or multiple documents (Hallet et al., 2007). With PowerAqua, we develop a new branch: open QA over structured data, allowing the system to benefit from knowledge combined from the wide range of ontologies autonomously created and maintained on the SW.

As stated in (D'Aquin et al., 2008c) “millions of RDF documents are now published online, describing hundreds of millions of entities through billions of statements. The SW is now the biggest, most distributed, and most heterogeneous KB that ever existed, and is very quickly evolving”. As such, with PowerAqua, we aim to go one step forward towards addressing the knowledge-accessing problem: supporting users in re-using and querying this novel, massively heterogeneous information space.

While a lot of interesting research has been carried out in the areas of ontology mapping and WSD as seen in our review in Section 3.2, the scenario of interest to us is different from the traditional applications in which these techniques have been applied. Therefore existing approaches

do not necessarily provide the kind of support and requirements (Section 3.3) required by PowerAqua to work in an open domain scenario. An important contribution of this chapter is to recognize and analyse the requirements that have to be addressed by novel mapping, disambiguation, ranking and fusion techniques. In particular, such techniques need to leverage the heterogeneity and large scale of online available semantic data to be used at run-time.

Chapter 4 Architecture Overview

The initial vision on the general architecture of the PowerAqua system was presented at the European Semantic Web Conference in 2006 (Lopez et al., 2006a).

The architecture of the system has evolved since that original vision. Here we present a general overview of the current architecture of the system and its main components. The system was demonstrated for the first time in the demo & poster session at the International Semantic Web Conference in 2007 (Lopez et al., 2007b).

4.1 Introduction

Being restricted to a predefined set of ontologies, and consequently being limited to specific domain environments, is a pervading drawback in ontology-based QA. By contrast, PowerAqua is designed to exploit multiple distributed ontologies and KBs to answer queries in multi-domain environments. PowerAqua interprets the user's query expressed in NL using the relevant available semantic information, automatically discovered on the SW, and translates the user terminology into ontology terminology, retrieving and combining accurate semantic entity values in response to the user's request.

In this chapter we give an overview of the PowerAqua architecture and the functionality of each component by means of an illustrative example. Each component provides its own API and can be independently used and integrated in other platforms.

4.2 Architecture: an illustrative example

To support users in querying and exploring SW content, PowerAqua accepts users' queries expressed in NL and retrieves precise answers by dynamically selecting and combining information distributed across highly heterogeneous semantic resources. To do so, at a coarse-grained level of abstraction, PowerAqua follows the pipeline architecture shown in Figure 4.1.

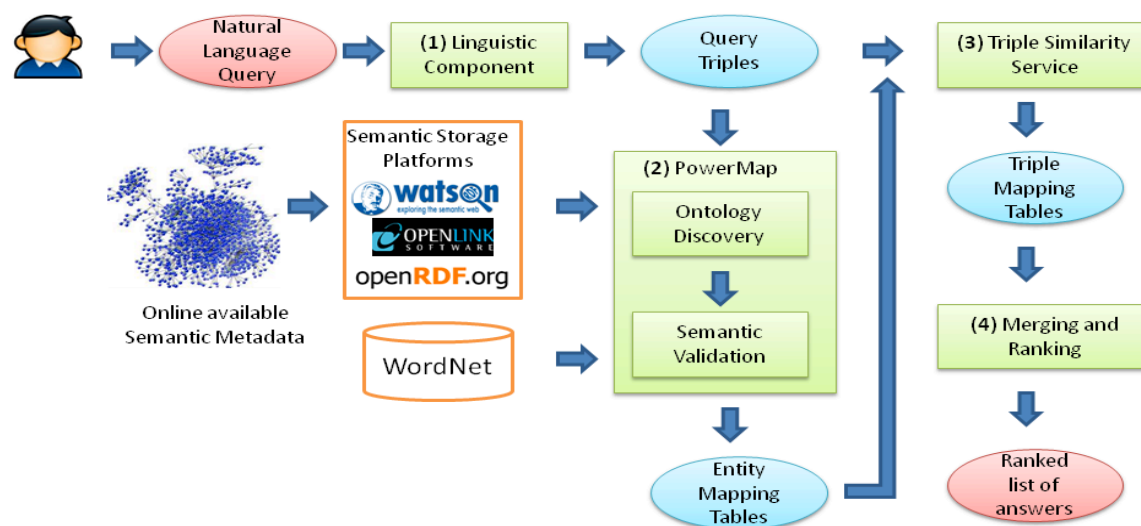


Figure 4.1 PowerAqua components

At setup, PowerAqua is coupled with several semantic storage platforms that collect, and provide fast access to, online semantic data. The mapping component, PowerMap, uses the Watson semantic search engine as the entry point to the SW. Watson crawls, analyses, indexes and stores online SW documents and provides an efficient access point through its API. Watson is used as PowerAqua's gateway to the dynamically evolving entire SW because its combination mechanisms for searching semantic documents (keyword search) and querying their content provides all necessary elements to select and exploit online semantic sources, without having to download the corresponding ontologies or to identify them at design time. In addition, PowerAqua can also query its own repositories and offers the capability to add and index (if the ontology platform does not provide full-text search on RDF literals) new online ontologies²⁹. To this purpose PowerMap provides a plug-in specification mechanism that supports a common API to query ontologies residing within

²⁹ Searching for text matches using SPARQL-like queries is very inefficient. PowerMap uses Lucene (lucene.apache.org) for the offline creation of inverted indexes to provide efficient keyword searches.

different repositories. Currently plug-ins are available for Sesame 1 and 2³⁰, Virtuoso³¹ and Watson; however, it is relatively easy to create plug-ins also for other semantic storage platforms. This infrastructure is further detailed in Section 4.4. In what follows, the set of components and the overall retrieval process are briefly summarized step by step with an illustrative example concerning the following query: “who plays in the rock group Nirvana?”.

The *Linguistic Component* is invoked in the first step; it analyses a NL query, and translates it into a set of linguistic triples, called *Query-Triples* (QTs), by identifying associations that relate the query terms together and mimic the structure of triples in an ontology, while using the NL terms in the query. The example query “who plays in the rock group nirvana?” is translated into the Query-Triple output: *<person/organization?, plays, rock group nirvana>*. The role of the Query-Triples is simply to provide an easy way to manipulate the input, as better results are expected by considering the triples rather than isolated words. This component is based on GATE (www.gate.ac.uk), and it is described in Chapter 5.

In the second step, the Query-Triples produced by the Linguistic Component are passed on to *the element mapping component, PowerMap*, which is responsible for identifying suitable semantic sources that are likely to provide the information requested by the user and answer the given query, thus producing initial element mappings between Query-Triple terms and entities in these sources. This component includes two subcomponents: *Ontology Discovery* and *Semantic Validation*. PowerMap is presented in Chapter 6, and therefore here we only summarize the key aspects of its behaviour, in the context of the illustrative example.

³⁰ <http://www.openrdf.org>

³¹ <http://virtuoso.openlinksw.com>

Initially, PowerMap's Ontology Discovery sub-module identifies the semantic sources, from all the sources available in the semantic storage platforms previously coupled with PowerAqua, that are likely to describe Query-Triple terms, and therefore to provide the information requested by the user. In this phase, match is performed by means of syntactic techniques. PowerMap maximizes recall in order to broaden the search space and bridge the gap between the terminology of the user and that of the various ontologies. This is achieved by searching for *approximate mappings* (lexical overlap) as well as *exact mappings* (lexical equality). These are jointly referred to as *equivalent mappings*. PowerMap uses both WordNet and the SW itself as sources of background knowledge to perform query expansion and to find lexically dissimilar (but semantically similar) matches – including synonyms, hypernyms and hyponyms.

The output is a set of *Entity Mapping Tables* (EMTs), where each table associates each Query-Triple term with a set of entities found on the SW (Table 4.1). For instance, the fifth row in Table 4.1 shows a mapping between Person (a term in the Query-Triple) and Musician (a concept in the Music Ontology) discovered using the hyponymy relation between Person and Musician, suggested by the TAP Ontology. For the example query, PowerMap is able to find a large number of candidate mappings in several ontologies. The first row in Table 4.1 indicates that no mappings were found for the compound term in the query, “rock group nirvana”, although they were found for the individual components. Consequently, if a query is formed by compound terms for which there are no ontological mappings in any ontology PowerAqua knows of, like for “rock group nirvana”, the compound is then split in its individual elements and the Query-Triples representations³² are modified accordingly. In this example, the triple *<person/organization?, plays, rock group*

³² In the case of ambiguity, i.e., more than one query term that can fulfil the same role, the ambiguous role is represented in its QT, splitting the candidate query terms by the symbol “/”. If the query term of a specific role is unknown, this situation is specified in the QT by means of the symbol, “?”, e.g., *<subject, ?, object>*.

nirvana> is then split into the set of triples <*person/organization?*, *plays*, *nirvana*> <*rock*, ?, *nirvana*> and <*group*, ?, *nirvana*>. Then, as shown in the Entity Mapping Table, the individual term “nirvana” has an approximate equivalent match with the instance of “researcher”, labelled “Nirvana Meratnia” in the SWETO ontology, and an exact match with the instance of “group” labelled “nirvana” in the music ontology, among others. The term “rock” has two exact matches in the ATO ontology (parent “substance”) and the music ontology (parent “specific-genre”), and partial matches in other ontologies, like “rock star” in the film festival ontology. For the sake of simplicity, we omit from the table the term “organization” which produces a large number of mappings.

Table 4.1 Partial view of the EMTs for QT <person/org?, plays, rock group nirvana>

Rock group Nirvana, rock group, group nirvana: \emptyset	
Nirvana	<i>Music ontology</i> : Nirvana (as an instance of group); <i>TAP</i> : MusicianNirvana (as a type of person); <i>SWETO</i> : Nirvana Meratnia (instance of researcher); <i>KIM</i> : Eden (as a synonym); <i>Spiritual stages ontology</i> : Nirvana; <i>Magnatune</i> : Passion of Nirvana (as an instance of “track”)
Rock	<i>Music ontology</i> : rock (as a type of genre); <i>SWETO</i> : Michael_Rock, Sibyl Rock, etc <i>ATO ontology</i> : rock (as a type of substance), Ayers_Rock (as a place); <i>film festival</i> : rock_star
Group	<i>Music ontology</i> : group, etc ...
Person	<i>Music ontology</i> : musicians (as a hyponym of person according to the TAP ontology), <i>TAP</i> : person, <i>KIM</i> : person, <i>Magnatune</i> : musicArtist (hyponym), etc ..
Play	<i>KIM ontology</i> : sport (as synonym of “play”).

The previous syntactic phase generates, in many cases, several possible candidates, which may provide alternative interpretations for a query term. Then, using the Semantic Validation component, PowerMap filters out the least promising mappings within an ontology by using a number of heuristics (equivalent mappings are preferred over hyper/hyponyms, redundant mappings within the same taxonomy are removed, etc.). In addition, this component also attempts to generate WordNet

synsets for all classes and individuals included in the EMTs – e.g., synsets would be generated for all the various entities listed in row 3 of Table 4.1, which are associated with the query term “rock”. The semantic validation component builds on techniques developed in the Word Sense Disambiguation community. This component exploits the background knowledge provided by WordNet and the ontological context surrounding candidate entities to disambiguate between different possible interpretations of the same query term within an ontology or across several ones. In our example the system fails to find a valid synset for Nirvana, as the intended meaning cannot be found in WordNet. It does, however, produce valid synsets for “rock”, interpreted as a “genre of popular music” in both the music and *TAP* ontologies, and as a “stone (material consisting of aggregate minerals)” in the *ATO* ontology. While obviously only one of these interpretations is correct, at this stage the system is unable to choose between the two. Nevertheless, other interpretations can already be ruled out at this stage. For instance, the association between the query term “rock” and class “stone”, interpreted as a “measure of weight”, can be discarded, because there is no intersection in WordNet between the intended synset and its synset in the ontology (therefore it does not appear in Table 4.1).

PowerAqua’s third step, the *Triple Similarity Service* (TSS), detailed in Chapter 7, takes as input the EMTs generated by PowerMap and the Query-Triples, and returns a set of *Triple Mapping Tables* (TMTs), which define a set of complete mappings between a Query-Triple, and the appropriate *Ontology Triples* (OTs) that better match, partially or completely, with the set of Query-Triples, as shown in Table 4.2. To determine the most likely interpretations for a user query, the TSS chooses if possible the ontologies that better cover the user query and domain. For instance it chooses ontologies related to music (which have mappings for “nirvana” and “person”) instead of those talking about spiritual stages (which have only mappings for “nirvana”). Then, it extracts, by

analyzing the ontology relationships, a small set of ontologies that jointly cover the user query and contain enough information to generate an answer to the question.

Finally, a major challenge faced by PowerAqua is that answers to a query may need to be derived by integrating different ontological facts, recovered in the previous step, and even different semantic sources and domains. Often, multiple, redundant information needs to be combined (or *merged*) to obtain a reduced number of answers. Then, because different semantic sources have varying levels of quality, when multiple answers are derived from alternative interpretations to a query it is important to be able to *rank* them, in terms of their relevance to the query at hand. In this fourth step, the *Merging and Ranking Component*, detailed in Chapter 8, generates the final answers from the returned Ontology Triples. In our example, one set of answers is produced by merging the semantically equivalent Ontology Triples obtained from the *music* and the *TAP* ontologies. The instances on which both ontologies agree (“Dave Grohl”, “Kurt Cobain”, “Chad Channing” and “Chris Novoselic”) are ranked higher. Then, the answer is augmented with additional instances from the *music* ontology (“Dan Peters”, “Dave Foster”, “Jason Everman”, “Pat Smear”, “Dale Crover”, and “Aaron Burckhard”, all former members of the band). The *music* ontology also produces additional “informative” mappings for the rest of the compound terms: *<nirvana, has-genre, rock>* *<nirvana, is-a, group>*. Once the different facts are merged and the answers are composed, this component applies a set of ranking criteria to sort the list of results. The answers from the *SWETO* and *magnatune* ontologies are ranked last.

Table 4.2 The TMT for OTs in online ontologies that match the QTs

<person / organization?, play, Nirvana>	
<i>SWETO</i>	<Nirvana Meratnia (equivalent), IS_A, person (equivalent)>
<i>Magnatune</i>	<MusicArtist (hyponym), maker (ad-hoc), Passion of Nirvana (equivalent)>
<i>Music</i>	<Musician (hyponym), has_members (ad-hoc), Nirvana (equivalent)>
<i>TAP</i>	<Person (equivalent), hasMember (ad-hoc), MusicianNirvana (equivalent)>

<rock, ?, nirvana>	
Music	<Nirvana, has_genre, rock>
<group, ?, nirvana>	
Music	<nirvana, is-a, group>

Each of the aforementioned components can be considered a research contribution on its own. In the next chapters we describe each component in detail and associate it with the research challenges that it aims to address.

A screenshot with the ontological answers from the TAP, the music ontologies and the merged answers can be seen in Figures 4.2, 4.3, 4.4 and 4.5.

The screenshot displays the PowerAqua Question Answering web interface. At the top left is the 'PowerAqua QUESTION ANSWERING' logo. The interface is divided into several sections:

- EXAMPLES:** A link to 'View a list of example queries and topics.'
- ASK ANOTHER QUESTION:** A text input field containing 'who play in the rock group nirvana' and an 'Ask' button. Below the input is a 'Make use of WATSON' checkbox.
- SOURCES:** A list of four sources with their respective fact counts and URLs:
 - 1 tapfull: "organization" 3 facts | "person" 8 facts | "nirvana" 3 facts | "play" 5 facts | "group" 2 facts | "rock" 3 facts | <http://kmi-web07.open.ac.uk:8080/sesame/tapfull>
 - 2 KIM: "organization" 1 facts | "person" 3 facts | "nirvana" 1 facts | "group" 1 facts | "rock" 11 facts | <http://kmi-web07.open.ac.uk:8080/sesame/KIM>
 - 3 ATO: "organization" 2 facts | "person" 7 facts | "play" 1 facts | "group" 2 facts | "rock" 1 facts | <http://kmi-web07.open.ac.uk:8080/sesame/ATO>
 - 4 SWETO: "organization" 1 facts | "person" 1 facts | "nirvana" 1 facts | "group" 1 facts | "rock" 14 facts | <http://kmi-web07.open.ac.uk:8080/sesame/SWETO>
- LINGUISTIC TRIPLES:** A section showing the query triples: <subject, relation, object>. The query is: <person / organization, play, nirvana> <rock, ?, nirvana> <group, ?, nirvana>, Category: WH_GENERICTERM.
- Answers:** Two tabs are visible: 'Individual Answers' and 'Merged Answers'. The 'Merged Answers' tab is selected, showing:
 - Ontologies found with answers for <[person, organization], play, nirvana> [3]
 - 1. mappings in <http://kmi-web07.open.ac.uk:8080/sesame/tapfull> [1]
 - "Dave Grohl"@en, "Chad Channing"@en, "Kurt Cobain"@en, "Krist Novoselic"@en, ...
 - Mapping 1 (rank @ 1): <Person (Person equivalentMatching), hasMember (hasMember ontology_ad_hoc), MusicianNirvana (Nirvana equivalentMatching)>
 - 4 answer(s):
 - MusicianGrohl, Dave ("Dave Grohl"@en)
 - MusicianChanning, Chad ("Chad Channing"@en)
 - MusicianCobain, Kurt ("Kurt Cobain"@en)
 - MusicianNovoselic, Chris ("Krist Novoselic"@en)

Figure 4.2 Answers for the example query in the TAP ontology

The query interface: aiming to balance usability and expressivity, PowerAqua accepts as input NL queries. An example of PowerAqua's web interface can be seen in Figure 4.2. In the top right part of the figure, under the title "ask another question" we can see the text box where the user introduces

her query. To support users in their initial interaction with the system, PowerAqua provides in the top left part of the interface a set of NL query examples. Once the query has been posed to the system, PowerAqua retrieves on the left hand side the list of semantic resources that are relevant to the user's query. On the right hand side of the interface, PowerAqua displays the set of linguistic triples to which the query has been translated, and below that the interface contains mechanisms to allow the user to see the answers before and after the merging (in the figure, Individual Answers / Merged Answers). The individual or partial answers (before the merging) are obtained for each ontology as a translation (in the form of ontological triples) for each Query-Triple (i.e., the *Triple Mapping Tables*). Thus, in the "Individual answers" tab the various ontological facts used to provide the answers are presented, showing how the user query has been interpreted, while in the "Merged Answers" the final ranked list of aggregated answers obtained for the query is presented, as well as different ranking criteria, (in the figure Alphabet/ Confidence/ Popularity, etc.) to sort the responses.

• mappings in <http://kmi-web07.open.ac.uk:8080/sesame/music> [1]

Dan_peters, Dave_grohl, Dave_foster, Jason_everman, Chad_channing, ...

The screenshot displays a web interface for querying a music ontology. At the top, it shows a mapping: "Mapping 1 (rank @ 3): < [Musician](#) (Musician hyponym) , [has_members](#) (has_members ontology_ad_hoc) , [Nirvana](#) (Nirvana equivalentMatching) >". Below this, it indicates "10 answer(s)" and lists several names with their corresponding ontology identifiers: [Dan_peters](#) (Dan_peters), [Dave_grohl](#) (Dave_grohl), [Dave_foster](#) (Dave_foster), [Jason_everman](#) (Jason_everman), and [Chad_channing](#) (Chad_channing). A link "View all Answers >>" is provided. At the bottom, there is a "Rate trust values" button with a star icon and a "Find these answers in Yahoo" button.

Figure 4.3 Answers for the example query in the *music* ontology for <person / organization?, play, Nirvana>

Ontologies found with answers for < [rock], null, nirvana > [1]

1. mappings in <http://kmi-web07.open.ac.uk:8080/sesame/music> [1]

Nirvana, ...

Mapping 1 (rank @ 1): < [Nirvana](#) (Nirvana equivalentMatching) , [has_genre](#) (has_genre ontology_ad_hoc) , [rock](#) (rock equivalentMatching) >

1 answer(s)

[Nirvana](#) (Nirvana)

⊖ ★ ★ ★ ★ ★ [Rate trust values](#) [Find these answers in Yahoo](#)

Ontologies found with answers for < [group], null, nirvana > [1]

1. mappings in <http://kmi-web07.open.ac.uk:8080/sesame/music> [2]

Nirvana, ...

Mapping 1 (rank @ 1): < [Nirvana](#) (Nirvana equivalentMatching) , [IS_A](#) , [Group](#) (Group equivalentMatching) >

1 answer(s)

[Nirvana](#) (Nirvana)

⊖ ★ ★ ★ ★ ★ [Rate trust values](#) [Find these answers in Yahoo](#)

Figure 4.4 Answers for the example query in the *music* ontology for the rest of Query-Triples

Individual Answers Merged Answers

Sort by: [Alphabet](#) / [Confidence](#) / [Popularity](#) / [WordNet Synset](#) / [Combined](#)

We found 12 answers in total from 3 ontologies

MusicianChanning, Chad ("Chad Channing" @en) tapfull	⊕ Explain	score: 2
Chad_channing (Chad_channing) music		
MusicianGrohl, Dave ("Dave Grohl" @en) tapfull	⊕ Explain	score: 2
Dave_grohl (Dave_grohl) music		
MusicianNovoselic, Chris ("Krist Novoselic" @en) tapfull	⊕ Explain	score: 2
Krist_novoselic (Krist_novoselic) music		
MusicianCobain, Kurt ("Kurt Cobain" @en) tapfull	⊕ Explain	score: 2
Kurt_cobain (Kurt_cobain) music		

Figure 4.5 Merged answers rank by popularity across ontologies

4.3 A triple-based approach

Core to the overall architecture is the triple-based data representation approach. First of all, the NL query gets translated by the Linguistic Component into a set of intermediate triple-based representations, which are referred to as the Query-Triples or linguistic triples. These triples are then further processed to match them to ontology compliant triples, or Onto-Triples, from which an

answer can be derived by applying merging and ranking techniques. Thus, the data model is triple based, namely it takes the form of a binary relation model and expresses statements (subject, predicate, object) in the same form as the RDF-based knowledge representation formalisms for the SW, such as RDF or OWL. Also, as pointed out by (Katz et al., 2002), although not all possible queries can be represented in the binary relational model, in practice these exceptions occur very infrequently. Hence, it makes sense for a query system targeted at the SW to adopt a triple-based model that shares the same format as the triples on the SW.

A query can be translated into one or more linguistic triples, and then each linguistic triple can be translated into one or more ontology-compliant triples. Each linguistic triple can also have additional lexical features in order to facilitate reasoning about the answer, such as the voice and tense of the relation. Another key feature of the triple is its *category*, these categories identify different basic structures of the NL query and their equivalent representation in the triples. Depending on the category, the triple tells us how to deal with its elements, what inference process is required and what kind of answer can be expected (e.g., affirmation / negation, description or a list of answers obtained as a conjunction or disjunction of all triples involved). In particular, different queries may be represented by triples of the same category, since, in NL, there can be different ways of asking the same question, i.e. “who works in the akt project?” (QT: <*person / organization?*, *works*, *akt project*>) and “Show me all researchers involved in the akt project” (QT: <*researchers?*, *involved*, *akt project*>). The number and classification of the linguistic triples may be modified during their life cycle in compliance with the target ontologies they subscribe to (e.g., as in the illustrative example by splitting the compound “rock group nirvana” into new QTs). The different linguistic categories are further explained in Chapter 5 of this thesis.

4.4 Underlying infrastructure

4.4.1 PowerMap ontology plugin mechanism

A key feature in the PowerAqua infrastructure is the use of a plug-in mechanism that allows PowerAqua to be configured for different Knowledge Representation (KR) languages and SW platforms. The functionality of the plug-in is implemented through SeRQL queries³³ in the case of Sesame 1 and 2 and SPARQL queries in the case of a Virtuoso SPARQL end-point. In the case of Watson the plug-in functionality is implemented on top of calls to the Watson API. All the plug-ins have a common *OntologyPlugin* API with all the functionality required to query the ontologies independently of their language, type, storage or location, thus providing a common access mechanism to multiple distributed ontologies. Plug-ins are loaded on demand, each ontology in use by PowerAqua is dynamically associated to an instantiation of a plug-in containing all the connection information needed to access the online ontology (ontology identifier, language, its corresponding framework and its location). Technical details on how this information is specified are given in Appendix B of this thesis. To access several ontologies at a time at run time, an additional API encapsulates a cache of ontologyPlugins in use. Internally this cache is managed as a HashTable of ontologyPlugins, where each plugin is associated to the ontology identifier that it is instantiating.

PowerAqua can use the Watson search engine to access and query online semantic data previously crawled by Watson and it can be configured to be used with multiple online semantic repositories, e.g., in Sesame or Virtuoso, to effectively store and access selected datasets that are not currently available in Watson, e.g., some of the datasets offered by the Linked Data community³⁴. Unlike Watson or Virtuoso, which implement their own indexing mechanism, in Sesame the plugin

³³ The seRQL query language: <http://www.openrdf.org/doc/sesame/users/ch06.html>

³⁴ <http://esw.w3.org/TaskForces/CommunityProjects/LinkingOpenData/DataSets>

is extended with an ontology independent module to create inverted indexes based on Lucene, as detailed in Section 4.4.1, in order to achieve real time performance for full-text searches.

4.4.2 Indexing mechanism in PowerMap

In our scenario, the user may need to interact with thousands of semantic documents structured according to hundreds of ontologies. To successfully manage such amounts of information in real time, if the ontological platform does not provide full-text index searches (e.g., Sesame), the ontologies and semantic sources are previously analysed and stored into one or several inverted indexes using Lucene.

Indexes can be generated or updated at any time, each inverted index indexes one or many ontologies. The semantic entities are indexed based on a mapping between each entity and a set of keywords that represent its meaning. These keywords are extracted by default from the entity local name and its `rdfs:label` meta property and optionally from any other ontology property. These mappings allow the generation of an inverted index where each keyword may be associated to several semantic entities from different ontologies. The semantic entities are uniquely identified within the systems considering: the identifier of the ontology they belong to, their URI and their type (class, property, individual or literal). Ontological classes and properties (schema) are indexed in a separate index from ontological instances and literals. To search the semantic information stored in the indexes we make use of the advantages that Lucene provides for standard and approximate searches combined with the capabilities of WordNet. A second index level is also generated with taxonomical information about each semantic entity using a MySQL database. PowerAqua makes use of both levels of indexing to increase the speed of mapping query terms to entities, managing the distributed semantic information in real time. The configuration of the indexes and the location of the ontology servers is specified through the use of XML files, as further explained in Appendix B of this thesis.

These indexes are useful to identify, in a first step, the set of potential semantic entities that can be associated to a set of terms describing a user query. The search capabilities provided by Lucene allow the user to perform three different kind of searches:

- Exact search: the index must contain the exact searched term to retrieve an answer.
- Fuzzy search: the keywords stories in the index must be “similar” to the searched term. The similarity is computed using the Levenshtein distance (Levenshtein, 1966) and considering an established prefix that represents the number of letters that must be equal at the beginning of both words.
- Spell search: the searched term may contain some mistakes, therefore Lucene provides suggestions of additional terms.

The second (optional) level of indexing stores taxonomical information to associate a retrieved ontology entity with its superclasses and subclasses. This taxonomical information can be used by PowerAqua to analyse and select only valid mappings at a rapid speed, without accessing and querying the ontologies.

Currently, the PowerMap indexes contain a large collection of ontologies and semantic data (about 2GBs) saved into online Sesame repositories. Our collection of ontologies includes high level ontologies, like ATO, TAP, SUMO, DOLCE, and large ontologies like SWETO_DBLP and SWETO with around 800,000 entities and 1,600,000 relationships. Although, very large ontologies such as DBpedia, which describes over 2.6 million entities, can also be stored in a Sesame repository and indexed with Lucene, for performance reasons they are stored in a Virtuoso repository, which is more efficient when querying large-scale repositories.

4.4.3 The Watson Semantic Web gateway

PowerAqua is also able to retrieve information if this has been crawled and indexed by Watson. Watson is described in (D'Aquin, 2008) as a gateway to the SW that provides a single access point to semantic resources. Watson collects, analyses, and indexes online resources in order to provide efficient services to support their exploitation. In principle such a tool plays the same role as a search engine for the standard web.

As seen in the literature review presented in Chapter 2 there have been several research efforts concerning providing a efficient access to the SW. The most influential example of this class of systems is probably Swoogle, a search engine that crawls and indexes online SW documents. Swoogle claims to adopt a web view on the SW and indeed, most of the techniques on which it relies are inspired by traditional web search engines. While relying on these well-studied techniques offers a range of advantages, it also constitutes a major limitation: by largely ignoring the semantic particularities of the data that it indexes, Swoogle falls short of offering the functionalities required by a SW gateway. Other SW search engines (e.g., Sindice and Falcon-S³⁵) adopt a similar viewpoint as Swoogle. Indeed out of these four ontology search engines, only Watson allows the user to search for specific semantic entities, such as <Class “Researcher”> or <Individual “Enrico Motta”>. The other engines support keyword search but fail to exploit the semantic nature of the content they store and therefore, are still rather limited in the ways they can support dynamic access to information at run time.

³⁵ <http://iws.seu.edu.cn/services/falcons/objectsearch/index.jsp>

PART II: THE POWERAQUA COMPONENTS IN DETAIL

In Part 2 of this thesis PowerAqua's architecture and components are described in detail, to give a comprehensive account of the way the system returns answers from user queries and addresses the research challenges discussed in Part 1.

"An object by itself is intensely uninteresting" (Grady Booch)

Chapter 5 Linguistic Analysis (step 1)

This chapter on the linguistic analysis of a Natural Language user query was first partially presented as part of a paper in the Natural Language Interfaces to Databases conference in 2004 (Lopez and Motta, 2004), and later it has been published as a section in the Journal of Web Semantics in 2007 (Lopez et al., 2007a).

5.1 Introduction

The Linguistic Component of PowerAqua is an evolution from the one used in AquaLog. The goal of this component is to create an *intermediate representation* that relates the terms from the Natural Language (NL) query. The output of this component, the *Query-Triples*, is used as the input for the rest of the PowerAqua components.

PowerAqua uses the GATE infrastructure and resources (Cunningham et al., 2002), including the *ANNIE* information extraction system³⁶, as part of the Linguistic Component to pre-process the query. Communication between PowerAqua and GATE takes place through the standard GATE API. As explained in this chapter, this pre-processing step helps towards the accurate classification of the query and the correct linkage of the query terms in the linguistic triples. This classification identifies the type of the question and hence the kind of answer required.

For the intermediate representation, we use a triple-based data model, to express relationships between two entities, rather than a logic-based formal representation, mainly because at this stage of the QA process we do not have to worry about getting the representation fully disambiguated. The role of the intermediate representation is simply to provide an easy way to manipulate input for the rest of the PowerAqua components, independently of the domain or representational choices of the sources to be queried. An alternative would be to use more complex linguistic parsing, using tools

³⁶ <http://gate.ac.uk/sale/tao/splitch6.html#chap:annie>

such the Stanford NLP parser (Klein et al., 2002) or the Open NLP parser ³⁷; however, as discussed by (Katz and Lin, 2003), it is often time consuming to generate parse trees, which makes them less suitable for performing an efficient linguistic analysis in PowerAqua. Furthermore, studies (Dittenbach et al., 2003) have shown that users often do not build complete sentences and when they do, they often introduce errors and ambiguities.

5.2 Realization of the Linguistic Component

After the execution of the GATE processing resources, a set of syntactic annotations associated with the input query are returned. In particular, PowerAqua makes use of the following pipeline of ANNIE processing resources included in GATE: *document reset*, *English tokeniser*, *sentence splitter*, *part-of-speech (POS) tagger*, *gazetteers* and *VP (verb group) chunker*. The annotations returned after the sequential execution of these resources include information about sentences, tokens, nouns and verbs. For example, we get voice (active / passive) and tense for the verbs and categories for the nouns, such as determinant, conjunction, possessive, determiner, preposition, existential, wh-determiner and count (singular / plural). These *features*, returned for each annotation, are important to create the triples.

For instance, verb annotations, together with the voice and sense, include information that indicates whether a verb is the main verb of the query, i.e., the one that separates the subject and object. This is referred to as the *lexical verb*. This information is used by the Linguistic Component to establish the proper attachment of prepositional phrases prior to the creation of the Query-Triples. Let's clarify this point with an example; let's consider the two queries:

- 1) "Which academics work in the akt project sponsored by epsrc?"
-

³⁷ Open NLP project: <http://opennlp.sourceforge.net>

2) “Which academics working in the akt project are sponsored by epsrc?”.

In the first example the main verb is “work”, which separates the nominal group “which academics” and the predicate “akt project sponsored by epsrc”. Hence, in this sentence “sponsored by epsrc” is related to “akt project”. In the second example, the predicate “are sponsored” defines the main verb, whose nominal group is “which academics working in akt”, and whose object is “epsrc”. Consequently, “sponsored by epsrc” is related to the subject of the previous clause, which is “academics”.

The PowerAqua Linguistic Component extends the set of annotations returned by GATE to improve its ability to identify noun terms (such as “academics”, “akt project” and “epsrc” in the earlier example), relations³⁸ (e.g., “work in” and “sponsored by”), question indicators (e.g., which/who/what, etc.) and types of questions (see Section 5.3). This is achieved by using a set of *JAPE grammars* that we originally built for AquaLog and extended for PowerAqua. JAPE is an expressive, finite state transducer based rule language offered by GATE. These *JAPE grammars* consist of a set of *phases*, that run sequentially, and each phase is defined as a set of pattern rules written to recognize regular expressions using annotations in documents. Thus, the *JAPE grammars*’ power lies in their ability to regard the data stored in the GATE annotation graphs as simple sequences, which can be matched deterministically by using regular expressions. Examples of the annotations obtained after the use of our JAPE grammars can be seen in Section 5.3 (JAPE grammars are further detailed in Section 5.4).

Currently, the Linguistic Component, through the number, position and kind of annotations identified by the *JAPE grammars* in the sentence (e.g., question type, nouns, relations, etc.),

³⁸ In order to identify relations we exploit the fact that NL commonly uses a preposition to express a relationship.

dynamically identifies around 14 different question types. Differently from QA approaches on unstructured text, where queries are classified according to the type of answer expected based on a hierarchy of name entity answer types (see Chapter 2.3.2), the classification of a query not only tells us the kind of answer that needs to be returned (e.g., an affirmation-negation or a list of entities), but also gives an indication of the most likely problems that the Linguistic Component and Triple Similarity Services will need to deal with to understand the NL query in question. Independently of how the query is formulated in NL, queries belonging to the same category have similar Query-Triple representations. Hence, the question type guides the process of creating the intermediate representation or Query-Triples using the annotated elements in the query (question indicators, noun terms and relations), as explained in the next section.

5.3 Classification of questions

In this section we present the classification of questions used by PowerAqua and discuss the rationale behind distinguishing between these 14 question types. According to the level of complexity of the linguistic query, question types are placed into three main groups. These three groups of factual queries are based on the number of triples needed to generate an equivalent representation of the query: basic queries (Section 5.3.1), basic queries with prepositional modifiers (Section 5.3.2) and combinations of queries (Section 5.3.3).

5.3.1 Linguistic triples for basic queries

Basic queries are those for which the Linguistic Component creates just one triple that represents a (explicit or implicit) relationship between a (explicit or implicit) generic term (the subject) and a second term (the object). There are five types of basic queries: the affirmative/negative query type, which are those queries that require an affirmation or negation as an answer (consisting on two noun terms and one relation), i.e. “is John Domingue a PhD student?” (<John Domingue, is-a, PhD

student>?), “is Motta working at IBM?” (<Motta, working IBM>?); and the set of type of queries constituted by a *wh-question*. These include both those starting with: *what*, *who*, *when*, *which*, *where*, *are there any*, *does anybody/anyone*, *how long* and *how many*, as well as imperative commands, such as *list*, *give*, *tell*, *name*, *show*, *describe*.

Wh-queries are further categorized into four query types depending on their equivalent intermediate representation. For each query type (or category) there is a set of patterns defined in the JAPE grammars. For example, in the query “who are the researchers involved in the Semantic Web?”, the generic triple has the form <*wh-query term?*, relation, second term>, where the relationship and both terms in the query (subject and object) are explicitly defined, thus generating the Query-Triple: <researchers?, involved, Semantic Web>. A different linguistic query with an equivalent triple representation is “which technologies has KMi produced?”, for which the associated triple is: <technologies?, has produced, KMi>. Both example patterns match constructions containing two explicit nominal groups, one of them accompanied by a wh-preposition (the focus of the query), and one relation. Thus, they belong to the same category. However, a query such as “are there any PhD students in X-Media?” belongs to a different category, where the relation is implicit or unknown <PhD students?, ?, X-Media> . Other queries may provide little or no information about the type of the expected answer, e.g., “what is the job title of Enrico Motta?” (<?, job title, Enrico Motta>, where “?” denotes the value or term is unknown), or they can define a generic enquiry about someone or something, e.g., “who is Peter Scott?” (<person, is-a, Peter Scott?>), “what is an ontology?” (<*what-is*, ?, *ontology?*>). The expected answers for the latter case are not the list of instances that satisfy the type of the wh-query term obtained after resolving each triple, but, the list of triples that describe the given term (either as subject or object) in the case of an enquire about an instance, or the list of instances (or subclasses, if there are not instances) in the case of an enquire about a class.

Figure 5.1 shows a screenshot of GATE annotations for the query terms and relations in the example queries used for each category, and the Query-Triples generated from these annotations by the Linguistic Component. In the figure, nominal groups, formed by nouns and certain adjectives and pronouns, are tagged as *NP* by the POS tagger; the nominal group accompanied by the *wh*-preposition is also tagged as *QU*; while relations are tagged as *REL*.

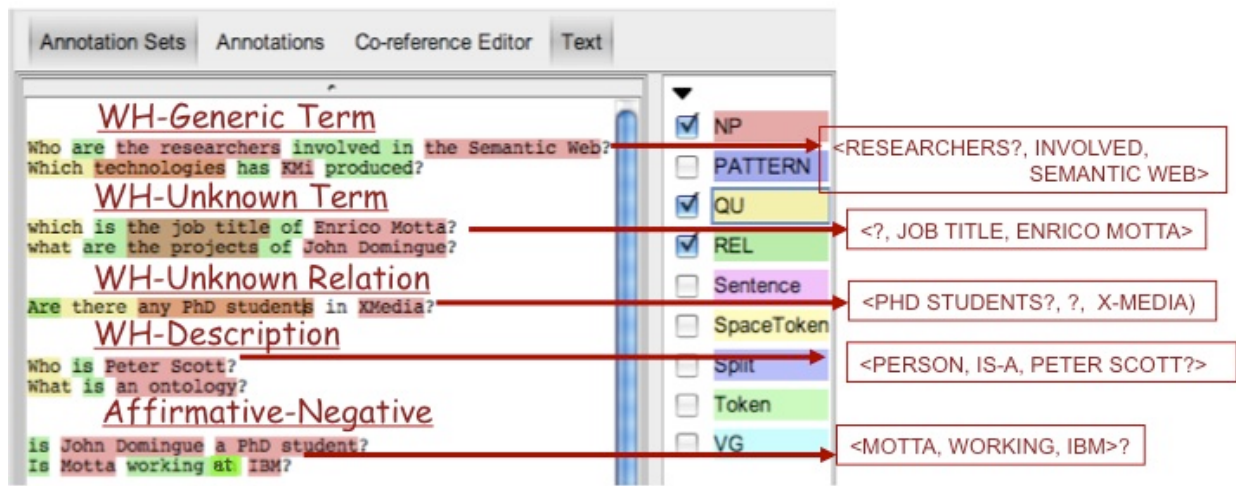


Figure 5.1 Screenshot example of GATE annotations and Query-Triples for basic queries

5.3.2 Linguistic triples for basic queries with prepositional modifiers

These queries (also called *3-terms queries*) are composed by one relation (explicit or implicit) between two nominal groups, and a third nominal group acting as a modifier clause, i.e., it modifies the meaning of other syntactic constituents. For example, let us consider the request “List all the publications in ECIR conferences about the Semantic Web”, where both “in ECIR conferences” and “about Semantic Web” are modifiers. The problem here is to identify the constituent to which each modifier has to be attached, e.g., does “about Semantic Web” refer to “publications” or to the “ECIR conferences”? If the ambiguity cannot be solved by linguistic procedures, the rest of the PowerAqua components are responsible for resolving this ambiguity through the use of the ontologies or by ranking the alternative readings. The task of the Linguistic Component is therefore to pass the ambiguity problem to the other components through the intermediate representation, as part of the

translation process, in our example: <publications?, ?, ECIR conference><publications/ECIR conference, ?, Semantic Web>, where the subject of the second triple is ambiguous and it can refer to either “publications” or “ECIR conference”, or both, according to the ontologies. The ambiguous role (i.e., more than one term can fulfil the same role) is represented in the Query-Triple with the symbol “/”.

Basic queries are those represented by just one Query-Triple at the linguistic stage. If modifiers are present, the triple will consist of three terms, instead of two terms and one relation. Once the ambiguity on the modifier attachment is solved, if not by the Linguistic Component by the subsequent PowerAqua components through the use of domain knowledge, these queries will generate at least two Query-Triples. For instance, the query “which projects on the Semantic Web are sponsored by epsrc?”, contains three terms: “projects” “Semantic Web” and “epsrc”, and one relation, namely “sponsored”, between “projects” and “epsrc”, in this query type the modifier clause “Semantic Web” relates to the subject of the query, “project”. There are four different categories depending on where the modifier clause is with respect to the relation, and whether the focus of the query (the *wh-term*) or the relation are explicitly, or not, defined (see examples in Figure 5.2).

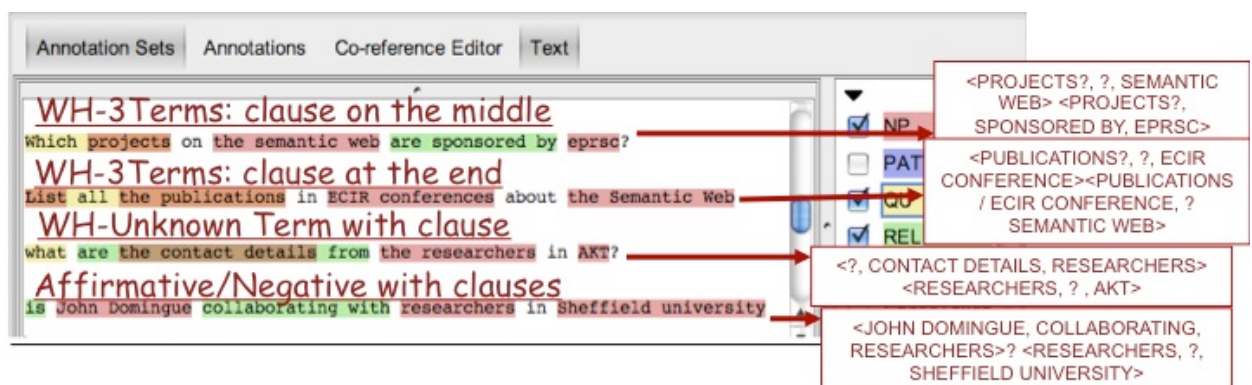


Figure 5.2 Screenshot example of GATE annotations and Query-Triples for queries with prepositional modifiers

5.3.3 Linguistic triples for combination of queries

A query can also be a composition of two explicit relationships between terms, or a composition of two basic queries. In either case, the intermediate representation usually consists of two triples, one triple per relationship.

These complex queries are classified depending not only on the kind of triples they generate but also on the resultant triple *categories*, which are the driving force to generate an answer by combining the triples in an appropriate way. There are three ways in which the query triples can be combined, as described in what follows (see examples for the different cases in Figure 5.3).

Firstly, queries can be combined by using “and” or “or” conjunction operators, as in “which projects are funded by epsrc and are about semantic web?”. This query will generate two Query-Triples with the same linguistic subject: <projects?, funded, epsrc> and <projects?, *is-are*, semantic web>, and the answer will be a combination of the partial answers obtained after resolving each triple.

Secondly, a query may be conditioned to a second query, as in “are there any PhD students supervised by researchers working in Question Answering?”, where the second clause modifies one of the earlier terms. At this stage, if the ambiguity cannot be solved by linguistic procedures, as in this example, the term to be modified by the second clause remains uncertain: <PhD students?, supervised, researchers> < PhD students/ researchers, working, Question Answering>.

Finally, we can have combinations of two basic patterns, e.g., “what is the web address of Peter who works for akt?”, or “which are the projects led by Enrico Motta which are related to multimedia technologies?”. Domain knowledge is normally required to find the common term that links the two query patterns, e.g., “who” normally refers to “person”, however, the Linguistic Component has no information about the type of the query terms (e.g., “Peter” is a “person”), thus, ambiguity is resolved in the next stages.

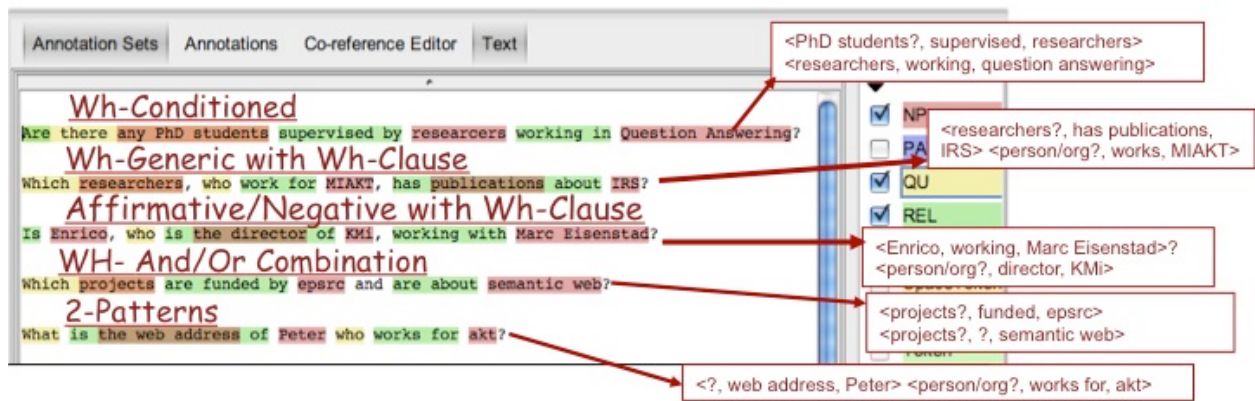


Figure 5.3 Screenshot example of GATE annotations and Query-Triples for combination of queries

5.4 The use of JAPE grammars: an illustrative example

In this section the use of *JAPE grammars* to classify a query and obtain the correct annotations, which are used by the Linguistic Component to create the Query Triples, is explained through an illustrative example query: “what are the research areas covered by the akt project?”.

Two *JAPE grammar* files are used in PowerAqua. The first one in the GATE execution list annotates the query terms as noun groups “research areas” and “akt project”; relations “are” and “covered by”; and query terms “what”. For instance, let’s consider the rule *Rule:NP* in Figure 5.4 which annotates terms as nouns. The pattern on the left hand side (i.e., before “→”) identifies noun phrases. Noun phrases are word sequences that start with zero or more determiners (identified by the (DET)* part of the pattern). Determiners can be followed by zero or more adverbs, adjectives, nouns or coordinating conjunctions in any order (identified by the ((RB)*ADJ)|NOUN|CC)* part of the pattern). A noun phrase mandatorily finishes with a noun (NOUN). RB, ADJ, NOUN and CC are macros and act as placeholders for other rules. Macros are defined to generalize POS tags produced by the POS tagger. Examples include the macro *LIST* used by the rule *Rule:QUI* and the macro *VERB* used by rule *Rule:REL1* in Figure 5.4. The macro *VERB* contains a disjunction with seven patterns, which means that the macro will fire if a word satisfies any of these seven patterns. POS tags are assigned in the *category* feature of a *Token* annotation used in GATE to encode the

information about the analysed question. Any word sequence identified by the left hand side of a rule can be referenced in other rules by its right hand side, e.g., *rel.REL={rule="REL1"}* is referenced by the macro *RELATION* in the second grammar (Figure 5.5).

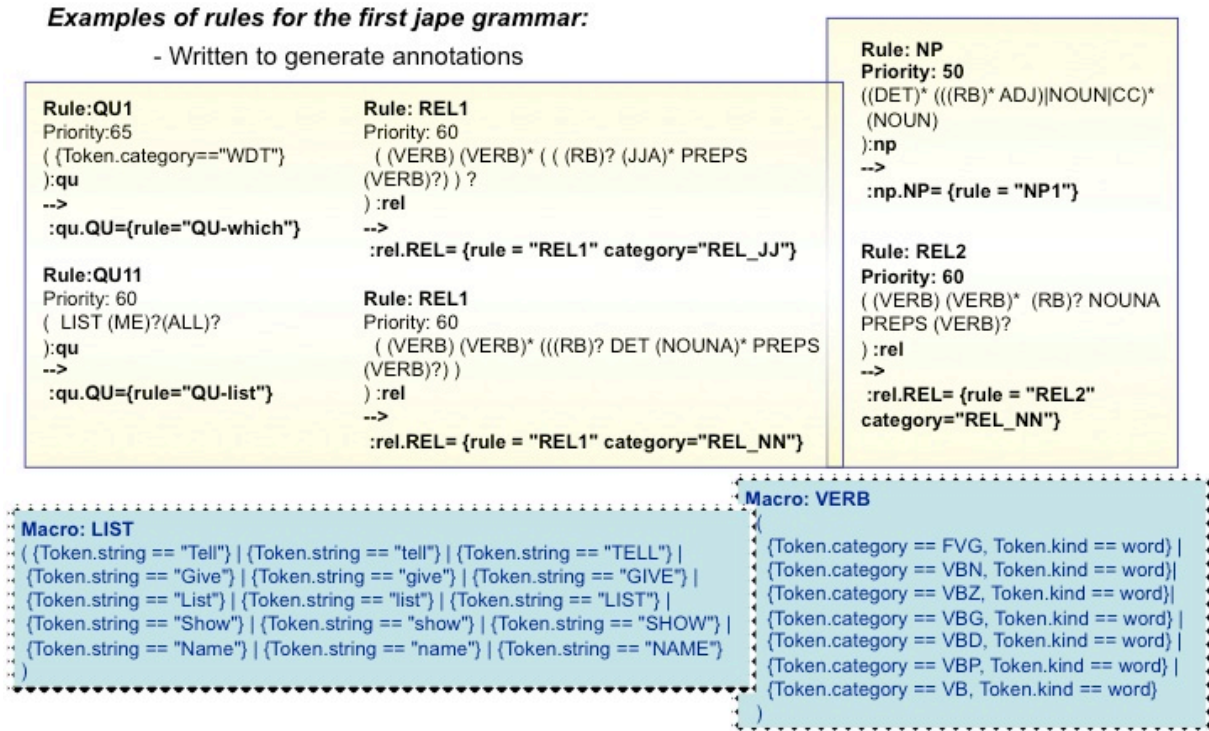


Figure 5.4. Set of JAPE rules used in the query example to generate annotations

The second grammar file based on the previous annotations (rules) classifies the example query in the category “*wh-genericterm*”. The set of *JAPE* rules used to classify the query can be seen in Figure 5.5. The pattern fired for that query “***QUWhich IS ARE TERM RELATION TERM***” is marked in bold. This pattern relies on the macros *QUWhich*, *IS ARE*, *TERM* and *RELATION*.

The advantage of using *JAPE grammars* is that although the subset of natural language the system is able to understand is limited by the coverage of the grammars, the architecture is very flexible and it is possible to extend this subset in a relatively easy way by updating the regular expressions in the *JAPE grammars*. This design choice ensures the portability of the system with respect to both ontologies and natural languages.

Examples of rules for the second jape grammar:

- Written to classify the query using previous annotations

```

Rule:PATTERN_whgenericterm
Priority:65
(
  ((Token.category==IN)? QUWhichClass (AUX|AUX_EXT) TERM RELATION) |
  (QUWhichClass (Modal)? TERM RELATION) | (QUWhichClass NOUN (NOUN)* RELATION) |
  (QUWhich IS_ARE TERM (QUWhich)? RELATION TERM) |
  (QUWho IS_ARE TERM RELATION) |
  (QUWhichClass (QUWhich|QU-who-what|Modal)? RELATION TERM) |
  (QU-who-what AUX TERM (QUWhich|QU-who-what)? RELATION TERM) |
  (QUWho RELATION TERM) |
  (QU-anybody (QUWhich|QU-who-what)? RELATION TERM) |
  (QUWhe AUX TERM RELATION) |
  (QUWhe RELATION TERM) |
  (QUListClass (QUWhich|QU-who-what)? RELATION TERM)
):pattern
-->
:pattern.PATTERN={rule="PATTERN_whgenericterm" category="wh-genericterm"}

```

Macro: QUWhich ({QU.rule == "QU-which"} {QU.rule == "QU-list"})	Macro: RELATION ({REL.rule == "REL1"} {REL.rule == "REL2"})	Macro: TERM ({NP.rule == "NP1"})
---	---	--

Figure 5.5. Set of JAPE rules used in the query example to classify the query

5.5 Discussion

We have examined the nature of the PowerAqua question answering task to identify what linguistic issues associated with the creation of a NL interface to ontologies need to be further addressed.

5.5.1 Question types and Query-Triple mapping

The first limitation of the PowerAqua Linguistic Component is related to the type of questions that it is able to handle. To bypass this limitation, when the system fails to classify a question, it treats it as an unclassified query, and instead of failing (as in AquaLog), it nevertheless generates an output, using the noun and verb annotations to create the triples. However, in this case the chances of misinterpreting the query are high due to the shallow analysis performed for uncategorized queries. For example, the query “Who are the people working in the same place as Fabio Ciravegna?” is mapped to the Query-Triples: <people?, working, same place> <same place, ?, Fabio Ciravegna>;

the meaning of “same – as” is not understood, thus PowerAqua erroneously maps “same place” to the title of an instance of “album”, generating incorrect ontological triples.

The Linguistic Component currently distinguishes affirmative/negative types of questions, “wh” questions and commands. However, it can not handle “why” and “how” queries, as these queries require understanding of causality or instrumental relations, rarely covered by ontological facts.

PowerAqua only supports questions referring to the present state of the world, as represented by an ontology and has limited capability to reason about temporal issues. Although simple questions formed with “how-long” or “when”, e.g., “when did the akt project start?” can be handled, it cannot cope with expressions such as, “in the last year” or “in the 80’s”. There is a research challenge in determining how to handle temporal data in a way which would be portable and open across ontologies.

In its current version, the Linguistic Component does not yet fully exploit quantifier scoping (“each”, “all”, “some”). Like temporal data, some basic aspects of quantification are domain independent but it is out of the scope of this thesis.

There are also a number of other linguistic forms that it would be helpful for PowerAqua to handle but which we have not yet tackled. These include other languages, genitive (‘s) and negative words (such as “not”, “never” and implicit negatives, such as “only”, “except”, and “other than”). PowerAqua also lacks a count feature in the triples indicating the number of instances to be retrieved to answer queries like “Name 30 different people...”. Prepositions and auxiliary words, such as: "in", "about", "of", "at", "for", "by", "the", "does", "do", "did", are removed from the Query-Triples because their semantics are not yet exploited to help in mapping Query-Triples to ontological triples in an open domain scenario. As discussed in (Cimiano and Minock, 2009) some systems have shown that it is possible to perform well by ignoring prepositions, e.g., PRECISE (Popescu et al.,

2003), AquaLog (Lopez et al., 2007a), Ginseng (Bernstein et al., 2006), Querix (Kaufmann et al., 2006), QuestIO (Tablan et al., 2008).

The reason why shallow approaches can successfully rely on the structure of the KB to interpret the query, ignoring linguistic details such as non-content words (prepositions), is that relevant relations in the user query are often implicitly hidden behind “light constructions” involving the verb “to have” or prepositions such as “with”, “of”³⁹, which cannot be mapped to the ontological relations by using deeper linguistic techniques and need to be inferred by relying on semantic techniques (based on the ontological type of the arguments); e.g., the query in Figure 5.2 “Which projects on the semantic web {..}?” is mapped into the Onto-Triple <projects, addressed-generic-area-of-interest, Semantic Web>. While a successful system can ignore non-content words (see evaluations in Chapter 9), in some domains PowerAqua could benefit from exploiting prepositions to disambiguate and select (rank) the most likely ontological representations of the input sentence. For example, the verb “flow” could map to ontological relations such as “flow-to” and “flow-from”, however, if we consider the preposition in the sentence “rivers flow into the Black sea”, then “flow-to” should be selected as the right mapping.

Thus, for PowerAqua to preserve the meaning of the user query without losing information and to increase its linguistic coverage, the Linguistic Component should be extended to use the domain independent role of spatial and temporal prepositions to solve ambiguities, and to account for the domain specific meaning of adjectives as well as superlatives.

In the creation of the Query-Triples, there is also loss of information regarding the directionality of the predicates. The subject of the Query-Triple, the wh-query term that indicates the expected type of the answers, does not necessarily corresponds to the subject of the NL query. However, as

³⁹ According to (Cimiano and Minock, 2009) more than 60% of queries in the Mooney dataset contains “light constructions”

discussed in Section 9.2.3, none of the queries in the evaluation gave inaccurate answers due to the directionality of the predicates. Directionality is determined by the way the information is structured in the ontologies, which often differs from the way the user formulates the query. Valid answers need to be obtained by inspecting relationships in both directions, e.g., $\langle \text{Russia, has_major_river, Neva} \rangle$ in the Russia ontology and $\langle \text{Lena_River, country, Russia} \rangle$ in DBpedia are both valid answers to “which rivers flow in Russia?”.

There are also some linguistic forms that we do not believe it is worth concentrating on. Since PowerAqua is used for stand-alone questions, it does not need to handle *anaphora resolution*, in which pronouns (e.g., “she”, “they”), and possessive determiners (e.g., “his”, “theirs”) are used to denote implicitly mentioned entities in an extended discourse.

Question complexity also influences the competence of the Linguistic Component. Currently, the Linguistic Component handles questions which generate one or two linguistic triples, which is the case for most of the questions we encountered in the evaluation studies presented in chapter 9. For example the NL query “which countries are traversed by the rivers that flow into the Caspian sea?” is translated into the Query-Triples: $\langle \text{countries?, traversed, rivers} \rangle \langle \text{rivers, flow, Caspian sea} \rangle$. Note that the Linguistic Component creates a triple for each relationship between terms, even if the relation is not explicit. At the same time this component minimizes the number of Query-Triples required to represent an NL query, independently of how this was formulated. For example, both queries “who are the academics working on the Semantic Web?” and “which academics are working on the Semantic Web?” are translated into the Query-Triple $\langle \text{academics, working, semantic web} \rangle$. However a query such as “What are the capitals of states that have cities named Springfield?”, which should be mapped to three Query-Triples $\langle \text{capitals?, ?, states} \rangle \langle \text{states, have, cities} \rangle \langle \text{cities, named, Springfield} \rangle$, is currently out of scope .

5.5.2 Ambiguity

In any non-trivial NL system, it is important to deal with the various sources of ambiguity and the possible ways of treating them. Some sentences are syntactically (structurally) ambiguous and although general world knowledge does not resolve this ambiguity, within a specific domain it may happen that only one of the interpretations is possible. Hence, the key issue is to apply the domain knowledge provided by the relevant semantic sources in order to resolve ambiguity. When the ambiguity cannot be resolved by domain knowledge the only reasonable course of action is to rank between the alternative readings.

Ambiguity is not only present at the level of terms and relations. For example, if we take the example question presented in (Copestake and Jones, 1990) “who takes the database course?” it may be that we are asking for “lecturers who teach databases” or for “students who study databases”. But also, ambiguity can be present in the way the modifiers are linked within a sentence. The Linguistic Component tries to solve the latter type of ambiguity by looking at the position of the modifier clause, however it is not always possible to disambiguate the modifier attachments at the linguistic level. In these cases, the disambiguation is regarded as part of the translation process and passed to the other PowerAqua components that will try to solve ambiguity by using domain knowledge. For example, for the query “what projects are headed by KMi researchers in natural language?”, two possible interpretations can be supported by available ontologies “projects headed by KMi researchers, where the KMi researcher has an interest in natural language” or “projects that are related to natural language and are headed by KMi researchers”.

The role of logical connectives and prepositions has also been recognized as a potential source of ambiguity in question answering. In the example presented in (Androutsopoulos et al., 1995): “list all applicants who live in California and Arizona”, the word “and” can be used to denote either disjunction or conjunction, introducing ambiguities which are difficult to resolve. In fact, the

possibility for “and” to denote a conjunction here should be ruled out, since an applicant cannot normally live in more than one state, but this, as in the earlier examples, requires domain knowledge. Therefore, the disjunction or conjunction disambiguation in an answer is passed as it is by the Linguistic Component, in order to be disambiguated by the rest of the PowerAqua components that can use the domain knowledge provided by the ontologies.

An additional linguistic feature not exploited yet is the use of the person and number of the verb to resolve the attachment of subordinate clauses. For example, consider the difference between the sentences “which academic works with Peter who has an interest in the semantic web?” and “which academics work with Peter who has an interest in the semantic web?”. The former example is truly ambiguous – either “Peter” or the “academic” could have an interest in the semantic web. However the latter can be solved if we take into account that the verb “has” is in the third person singular, therefore the second part “has interests in the semantic web” must be linked to “Peter” for the sentence to be syntactically correct. This approach is not implemented in PowerAqua. The obvious disadvantage is that disambiguation will be required for both examples in further stages of the algorithm. The advantage is that some robustness is obtained over syntactically incorrect sentences. Whereas methods such as the person and number of the verb assume that all sentences are well-formed, our experience with the sample of questions we gathered for the evaluation studies was that this is not necessarily the case.

5.5.3 Reasoning mechanism and services

As future work, PowerAqua should also benefit from the inclusion of Ranking Services that will solve comparative and superlative questions such as “what are the most successful projects?” or “what are the largest states in the USA?”. To answer these questions, the system must be able to carry out reasoning based on the information present in the relevant ontologies, usually over datatype properties. These services must be able to manipulate both general and ontology-dependent

knowledge, e.g., the criteria that make a project successful could be the number of citations or the amount of funding. Similarly, the largest state can refer to the area of the state or its population. Therefore, the first problem identified is how can we make these criteria as ontology independent as possible, and available to any application that may need them.

Finally, we also need fallback mechanisms that can help the user to reformulate her questions in a way that the system can support them.

5.6 Conclusions

Since the development of early syntax-based NLIDB systems, where a parsed question is directly mapped to a database expression by means of rules (Androutsopoulos, 1995), there have been improvements in the availability of lexical knowledge bases, such as WordNet, and modular and robust NLP systems, like GATE. Many closed-domain NL interfaces are very rich in NL understanding and can handle questions that are linguistically much more complex than the ones handled by the current version of PowerAqua. However, PowerAqua has a very light and extendable NL interface that allows it to produce triples in an open domain scenario, after only a shallow but efficient parsing.

It is important to emphasize that, at this stage, all the terms in a query are treated simply as strings, without introducing any correspondence with ontology entities. This is because the analysis is completely domain independent and is entirely based on the GATE analysis of the English language. Hence, the Query-Triple is only meant to provide a simplified way of representing the NL-query.

Chapter 6 Element Mapping (step 2)

The element mapping component of PowerAqua, PowerMap, is detailed in this chapter. The first version of PowerMap was presented at the International Semantic Web Conference in 2006 (Lopez et al., 2006b).

The evaluation of PowerMap, summarized in Section 6.4. of this chapter, is the result of a collaborative evaluation performed by the author and the *University of Zaragoza* to compare PowerMap with a Word Sense Disambiguation approach developed by the University of Zaragoza, in the context of an ontology-matching task. The full evaluation has been presented at the Workshop on Ontology Matching at the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference (Gracia et al., 2007).

6.1 Introduction to PowerMap Mapping Component

The Element Mapping Component, PowerMap, identifies the semantic resources and entities that may be relevant to the terms in the Query-Triple(s), created by the Linguistic component in the previous step. PowerMap is a hybrid knowledge-based matching algorithm which employs terminological and structural scheme matching techniques with the assistance of large scale ontological and lexical resources. As explained in Chapter 3, PowerMap is the solution adopted by PowerAqua to translate user terminology into several ontology-compliant terminologies, while at the same time performing effectively to guarantee real-time question answering. A critical problem is to distinguish (1) between different vocabularies used by different ontologies to describe similar information across domains; and (2) similarly spelled ontology classes in different ontologies that may not have precisely matched meanings. This problem is especially critical for a QA system that aims to aggregate answers from different sources in an open scenario.

In Section 4.4.2 of this thesis we described the underlying offline indexing mechanism, based on Lucene, which makes it possible to successfully manage large amounts of information distributed in

the back-end repositories in real time.⁴⁰ In this chapter, we provide more details on the PowerMap run time matching algorithm, which is able to map linguistic terms into entities drawn from different ontologies.

The PowerMap mapping component consists of two major phases exhibiting increasing complexity in the techniques they use: the first phase looks for the ontological entities which are syntactically similar to those on the linguistic terms, while the second phase identifies the sense of the matches and excludes those that do not match the input linguistic terms semantically. Hence the most time consuming techniques are executed last, when the search has been narrowed down to a smaller set of ontologies.

Phase I: ontology selection and discovery (syntactic mapping). The role of this phase is to identify, at run time, mapping candidates for all query terms in different online ontologies, and therefore to select the set of ontologies likely to provide the information requested by the query. This is the simplest phase as it only considers labels (i.e., ignores the structure of ontologies). It relies on Lucene searches, simple string-based comparison methods, and WordNet to look-up lexically related words (synonyms, hypernyms and hyponyms). Filtering heuristics are also applied to eliminate redundant mappings within an ontology, or the less relevant ones.

Phase II: semantic analysis and filtering (semantic mapping). This phase operates on the reduced set of ontologies identified in the previous phase. The goal is to verify the syntactic mappings identified previously from a semantic perspective, to elicit their sense in the context of the ontology, and, when possible, to exclude those for which the intended meaning of the query term differs from

⁴⁰ This mechanism allows PowerMap to efficiently discover mappings in large-scale ontology repositories that do not provide full text searches, like Sesame, but it is not necessary when using the Watson search engine or storage platforms that provide their own indexes, such as Virtuoso.

the intended meaning of the concept proposed as a candidate match. Unlike the previous phase, this phase relies on more complex methods. First, it exploits the hierarchical structure of the candidate ontologies to elicit the sense of the candidate concepts. Second, it uses WordNet based methods to compute the *semantic similarity* between the query terms and the ontology classes.

The output of PowerMap is a set of Entity Mapping Tables (EMT), each corresponding to one linguistic term, where each EMT contains the ontology elements (from different ontologies) to which the linguistic term was matched.

In the next sections, we describe the main phases of PowerMap, then conclude with an evaluation of PowerMap's ability to produce semantically sound mappings. Experiments to test PowerMap's ability to scale up to the ever-growing SW, in the context of Question Answering over Linked Data sources are reported in Chapter 9.

6.2 Phase I: ontology selection and discovery

The syntactic mapping phase identifies candidate entities from different ontologies (classes, instances, properties or literals) to be mapped to each input term in the triple(s) by means of *syntax-driven techniques*, using the labels and alternative names of the ontology elements.

Since the universe of discourse is determined by the ontologies in use, it is normal to encounter discrepancies between the natural language questions prompted by the user and the set of terms recognized in the ontologies. Terminological and background knowledge techniques are used to broaden the search space and bridge the gap between the user terminology and the terminology used by the multiple heterogeneous ontologies. To broaden the search space we use not just the terms of the linguistic triple, but also lexically related words obtained from WordNet (in particular

synonyms, meronyms and hypernyms⁴¹), in order to search for exact and approximate mappings in the ontologies, e.g., “PhDStudent” is an approximate match for “Student” while “metropolis” is a synonym match for “city”.

A generic thesaurus such as WordNet helps to compare terms that have dissimilar labels, but it does not include all possible synonyms and senses, and it is limited in the use of nominal compounds (e.g., WordNet contains the term “municipality” but not a term like “municipal-unit” which can be found in some ontologies). Therefore, to ensure a higher recall PowerMap also initiates a spreading activation search across ontologies to find additional ontological terms that are lexically different from the original keywords, and that are not covered by WordNet. In other words, the ontologies in the Semantic Web (SW) are also used as background sources of knowledge. Synonyms are found through properties like *owl:sameAs*, while hypernyms and hyponyms are found by looking at the superclasses and subclasses of the previous set of ontology matches. For example, in the KA2 ontology “AcademicStaff” is a superclass of “researcher”, and a few new hypernyms mappings can be found in relevant ontologies using the term “AcademicStaff” to wider the search scope.

Syntax driven techniques (approximate searches and lexically related words) are good mechanisms to broaden the search space as they can return a lot of hits that contain the term. However, their main weakness is that many of the discovered ontology entities may be similarly spelled words that do not have precisely the same meaning, providing potential alternative interpretations for the same query term. Thus, in the next phase PowerMap assesses the semantic validity of the mappings, along with their possible interpretations, in the context of the query and the

⁴¹ hyponyms are not currently used as they introduces a lot of noise.

candidate ontologies they belong to. However, some basic syntactic filtering is already performed at this stage through the use of:

String Distance Metrics for Name-Matching Tasks are applied using an open-source API from Carnegie Mellon University (Cohen et al., 2003). This API comprises a number of string distance metrics proposed by different communities, including edit-distance metrics, fast heuristic string comparators, token-based distance metrics, and hybrid methods - see (Cohen et al., 2003) for an experimental comparison between the different metrics. PowerAqua combines the use of the Jaro-based metrics with the use of different thresholds to filter noisy mappings, based on both their labels and alternative names. For example, the mappings “Guatemala city” or “Mexican city” are considered good mappings for “city” while a mapping to the class “Wild Mice Trapped in Anjo-City, Japan, M M Molossinus” in the nciOncology ontology is discarded by these String metrics.

Filtering syntactic heuristics are applied in those cases where one ontology presents several different mappings for the same linguistic term. For example, a filtering heuristic could select only exact mappings (singular and plural variations are also considered exact mappings), and discard all the other mappings (approximate, synonyms, hypernyms and hyponym mappings) within the same ontology. Second, information about the superclasses of a mapped entity could be used to select the most informative (up in the hierarchy) equivalent mappings within the same ontological taxonomy, e.g., to narrow down the number mappings for the query term “researchers” the mapping “researcher” is selected over its subclass, and also candidate mapping, “senior researcher”.

6.3 Phase II: semantic analysis and filtering

A semantic mapping component, which considers the content of an information item and its intended meaning, is applied after the syntactic mapping component to carry out the following tasks:

- To help in the disambiguation process by checking whether the meaning of a candidate ontology entity is consistent with that of the query term.
- To identify whether concepts for different ontologies that are matches to the same query term are semantically equivalent. This is crucial to ensure candidate alternative answers from different ontologies provide consistent interpretations of the query terms.

Therefore, once the set of possible syntactic mappings for each query term have been identified, PowerMap determines the sense of the identified entities to extract and check their semantic validity and, when enough information is available, to discard those matches that are semantically inappropriate. For instance, the ontological concept “Game” obtained as a synonym of the query term “Sport” should be discarded if in the context of the candidate ontology it refers to “Hunted Animals”.

The meaning of a class is given not only by its label but, more importantly, by its sense implied by its position in the ontology taxonomy. Therefore, the semantic similarity between the linguistic terms and the concepts from distinct ontologies (in the case of instances we consider the class they belong to) is computed by taking into account their meaning as given by their place in the ontology hierarchy, through the use of a WordNet based methodology.

In what follows we first define a WordNet based algorithm to compute the semantic similarity between two words (Section 6.3.1). Second, we describe how this WordNet based method is used to identify the meaning of a term and to check the semantic validity of the ontological mappings (Section 6.3.2).

6.3.1 A WordNet-based algorithm for computing Semantic Similarity

In this section, we detail the semantic similarity algorithm used to find shared senses between two words.

In *hierarchy distance based matchers* (Giunchiglia and Yatskevich, 2004), *relatedness* between words is measured by the distance between two concept senses in a given input hierarchy. In particular, *similarity* is measured by looking at the shortest path between two given concept senses in the WordNet *IS-A* taxonomy of concepts. As discussed in Section 3.2.2, similarity is a more specialized notion than relatedness, where relatedness refers to any kind of functional relationship or frequent association, which cannot always be determined purely from a priori lexical resources such as WordNet (Resnik, 1995).

We say that two words are semantically similar if any of the following three cases hold:

1. They have a *synset*⁴² in common, e.g., “capital” and “working capital” share the sense “assets available for use in the production of further assets”.
2. Any of the senses of a word is a hypernym / hyponym in the taxonomy of any of the senses of the other word, e.g., “seat” (a center of authority - as a city from which authority is exercised) is an hypernym of “capital” (a seat of government).
3. If they are not hypernym / hyponyms but there exists an allowable IS-A path (in the WordNet taxonomy) connecting a sense associated with each word. A path is allowable if it contains no more than 10 links. For example, between the terms “capital” and “city” presented in the example in Section 6.3.3.

To evaluate the semantic similarity between two words we use an algorithm that makes use of two WordNet indexes: the *depth*, which measures the minimum path distance between two terms, and the *common parent index* (C.P.I.), which is the most-specific concept in the IS-A taxonomy that subsumes two terms. The rationale behind this is based on two criteria of similarity between

⁴² Sense is the meaning of a word in WordNet. Each sense of a word is in a different synset (i.e., synonym set).

concepts established by Resnik (Resnik, 1995). The first one states that the shorter the path between two terms the more similar they are, this is measured using the *depth* index. The second one measures the extent to which the concepts share information in common, which in an *IS-A* taxonomy can be determined by inspecting the relative position of the *C.P.I.* index. Apart from point 1, in which the words have a synset in common, the most immediate case occurs in point 2 (*C.P.I.* = 1, *Depth* = 1), e.g., seat (a center of authority) is the common subsumer (hypernym) of capital (a seat of government)⁴³.

Formally, similarity is determined as a function of the minimum path distance between the senses of the terms and to the extent to which they share information in common in the *IS_A* hierarchy of WordNet, as given by the Wu and Palmer's formula (Wu and Palmer, 2004):

$$\text{Similarity}(t_1, t_2) = t_1 \sim t_2 = \frac{2 \times \text{depth}(\text{C.P.I.}(t_1, t_2))}{\text{depth}(t_1, t_2) + 2 \times \text{depth}(\text{C.P.I.}(t_1, t_2))}$$

Equation 1. Similarity formula between two different word senses: the uppercase letters T_1 and T_2 denote terms (words) and the lowercase letters t_1 and t_2 senses. We write $\text{depth}(\text{C.P.I.}(\mathbf{t}_1, \mathbf{t}_2))$ for the depth between the common parent of \mathbf{t}_1 and \mathbf{t}_2 and the root of the *IS-A* hierarchy.

6.3.2 Verifying the meaning and semantic validity of mappings

We elicit the senses of the ontological matches for a query term in three steps. Let S_T and S_C be all the WordNet senses of a query term T and its candidate mapped term C . First, the Similarity formula presented in Equation 1 is used to define the best senses of C with respect to T :

$$S_{C,T} = \{c \in S_C \mid \exists t \in S_T (\max(t \sim c))\}$$

That is, $S_{C,T}$ is the set of those senses c of C for which there exists a sense t of T such that t and c are similar ($t \sim c$) and for which the similarity between c and t is maximum. If $S_{C,T}$ is empty, the

⁴³ Banerjee and Pedersen (Banerjee and Pedersen, 2006) consider the definition (gloss) of the sense of a word to measure relatedness, e.g., the gloss for “capital” includes the hypernym word “seat”. However, as we are only focusing in similarity between words we do not use the glosses.

mapping C is discarded because the intended meaning of the term T is not the same as that of the concept C .

For example, “fowl-cholera” in the fao-agrovoc ontology⁴⁴ should be discarded as a valid mapping for “fowl”, because they do not share any senses and there are no relevant *IS-A* paths that connect them – “fowl” is connected to the root through the paths “bird” or “meat”, while “fowl cholera” is connected through “disease”.

Second, to elicit the sense of a candidate concept we adapted the algorithm presented in (Magnini et al., 2003), used to make explicit the semantics hidden in schema models, to exploit the use of the notion of *IS-A* similarity given by the ontology taxonomy. In a nutshell, given a concept C and either one of its ancestors R all WordNet senses for both labels are retrieved. Then, if any of the senses for C is similar to any of the senses for R , as previously defined by the similarity formula presented in Equation 1, then from these senses of C the ones (or one) with maximum similarity value are considered the right ones. Thus, the true senses of C are determined by its place in the hierarchy of the ontology:

$$S^H_C = \{c \in S_C \mid \forall R ((R > C) \rightarrow (\exists r \in S_R (\max(c \sim r))))\}$$

That is, S^H_C consists only of those senses of C that are similar to some senses of all of the ancestors of C in the ontology (we use $>$ to express the hierarchical order relation in an ontology).

Finally, we then intersect these senses, S^H_C , with the senses obtained in our previous step, $S_{C,T}$. Note that by intersection we mean to extract the senses that are similar (according to Equation 1), even if they are not exactly the same sense. Obviously, if this intersection is empty, it means that the *real meaning* of the concept in the ontology (i.e., the senses of the concept delimited by the

⁴⁴ www.fao.org/agrovoc

ontological hierarchy in the second step) is different from the *intended meaning* of the matched concept (i.e., the senses of the concept delimited by the query term in the first step), and therefore that mapping should be discarded. Otherwise, the intersection represents the shared senses that are captured by the mapping. An example is presented in Section 6.3.3.

As a result, mappings that do not have a valid semantic interpretation are discarded. The valid mappings across ontologies that refer to the same sense are grouped together. Valid mappings with different meanings would be disambiguated or ranked by PowerAqua in further steps. For example if the term “capital” is mapped to two concepts capital/assets and capital/city having the same label but different meanings in the ontology, PowerAqua should disambiguate the correct interpretation, according to the user query, in further steps. In here, we just group the element mappings according to their interpretations.

The technique explained here fully relies on sense information and coverage provided by WordNet to compute semantic similarity. However, this methodology, evaluated and discussed in Section 6.4 provides good precision for filtering meaningful mappings, and has a low negative impact on recall (discarding relevant elements).

6.3.3 Experimental example

Consider the following syntactic mappings for the term (T) “capital” in the query “what is the capital of Spain?” in these two ontologies:

1. Ksw-KB ontology. Contains the class (C1) “capital-city”, of which superclass is “city”.
2. ATO ontology: Contains the class (C2) “seat” as a WordNet hypernym of “capital”, of which superclass is “furniture”.

The possible senses for the mapped classes C1 and C2 when considering the query term T are reduced to:

$$S_{C1,T} = \{Capital\#all : \text{all senses for capital} - C1 \text{ and } T \text{ share the same lemma "capital"}\}$$

$$S_{C2,T} = \{Seat\#c \mid Capital\#c: \text{seat of government -- a centre of authority}\}$$

The senses of the mapped classes C1 and C2 in the context of the ontology they belong to (their ontological meaning) are obtained by looking at their ontology ancestor, as explained in Section 6.3.2. For instance, the results of computing similarity for the mapped term C1 “capital-city”, whose lemma is “capital”, when considering its ontology ancestor “city” are presented in Table 6.1.

Table 6.1. Similarity between “capital” and its ontology ancestor “city”

	City#1: large and densely populated urban area, metropolis	City#2: an incorporated administrative district
Capital#a (assets)	Not an allowable IS-A path	
Capital#b (wealth)	Not an allowable IS-A path	
Capital#c (seat of government)	Depth = 8, C.P.I = region Depth (C.P.I) = 3 (region, location, entity)	Depth = 7, C.P.I = region Depth (C.P.I) = 3 (entity, location, region)
Capital#d (capital letter)	Not an allowable IS-A path	
Capital#e (upper part column)	Depth = 8, C.P.I = location Depth (C.P.I) = 2 (entity, location)	Depth = 7, C.P.I = location Depth (C.P.I) = 2 (entity, location)

Analysing the results of Table 6.1 we can quickly choose the senses *Capital#c* and *Capital#e* for both *city#1* and *city#2*, and discard the others, for which there is not an allowable IS-A path. The sense *Capital#c* is chosen as the valid sense for C1 in the ontology, as it has the highest similarity value according to the Wu and Palmer formula. This is because between *Capital#e* and *City#1,2* there are only 2 common subsumers (entity and location), both of them representing abstract top elements of the WordNet taxonomy, while for *Capital#c* we have 3 common subsumers.

$$S^H_{C1} = \{Capital \#c\}$$

The valid sense of mapping C1 in the Ksw-Kb, as shown Figure 6.1, is therefore:

$$S^H_{C1} \cap S_{C1,T} = \{Capital \#c\}$$

Similarly, the same semantic similarity is computed for C2 in the ATO ontology. For C2 “seat”, whose superclass is “furniture”, the extracted ontological meaning refers to $S^H_{C2} = \{Seat\#f -$

furniture that is designed for sitting on}, so the intersection between its taxonomical meaning in the ontology {Seat#f - furniture} and its intended meaning a match for the query term {Seat#c - seat of government} is empty ($S_{C2}^H \cap S_{C2,T} = \emptyset$) and therefore the mapping should be discarded (see Figure 6.1).

Element Mappings in http://plainmoor.open.ac.uk:8080/sesame/ato for "capital"						
LABEL	SEMANANTIC RELATION	TYPE (SCORE)	SUPERCLASSES	Taxonomy synsets	Match synsets	Valid synsets
Seat	Hyperrym seat	class 1.0	foo:bar#Furniture label	[Synset] seat -- (furniture that is designed for sitting on;]	[Synset] seat -- (a center of authority (as a city from which authority is exercised))]	
Element Mappings found in http://plainmoor.open.ac.uk:8080/sesame/ksw-kb for "capital"						
LABEL	SEMANANTIC RELATION	TYPE (SCORE)	SUPERCLASSES	Taxonomy synsets	Match synsets	Valid synsets
Has-capital	Equivalent Matching	property 0.8			[Synset] capital, -- (assets available for use in the production of further assets)]	
Capital-city	Equivalent Matching	class 0.8	http://semanticweb.kmi.open.ac.uk/ontologies/active-portal-ontology-latest.owl#city	[Synset:] capital -- (a seat of government)]	[Synset] capital, chapter -- (the upper part of a column that supports entablature)] [Synset] capital -- (a seat of seat of government)] [Synset:] Das_Kapital -- (a book written by Karl Marx] [Synset] capital, capital_letter, majuscule]	[Synset:] capital -- (a seat of seat of government)]

Figure 6.1. Some of the element level mappings for the keyword "capital" and its senses

6.4 PowerMap evaluation of semantic capabilities

In this section we evaluate the performance of PowerMap to extract semantically sound mappings in the context of a mapping task. In Section 6.4.1 we describe the evaluation task, in Section 6.4.2 we explain how PowerMap has been applied for this task. We analyse the results in Section 6.4.3, and discuss the algorithm's limitations in 6.4.4.

6.4.1 Evaluation task: Improving term anchoring

A new ontology matching paradigm, which uses the SW to derive mappings from an exploration of multiple and heterogeneous online ontologies, has been proposed in (Sabou et al., 2006a). This method implemented in the *Scarlet* mapping algorithms (Sabou et al, 2006a) performs ontology matching by harvesting ontologies dynamically selected from Swoogle. For example, when

matching two concepts A and B, labelled Researcher and AcademicStaff respectively, the first step is to find online ontologies containing concepts A' and B' equivalent to A and B. This process is called anchoring, and A' and B' are the *anchor* terms. Based on the relations that link A' and B' in the retrieved ontologies, a mapping is then derived between A and B. In our example, the mapping can be either provided by a single ontology stating that Researcher \subseteq AcademicStaff, or spread in a first ontology stating Researcher \subseteq ResearchStaff, and that ResearchStaff \subseteq AcademicStaff in a second ontology.

While this method exhibited good performance, it relied on merely syntactical techniques to anchor the terms to be matched to those found on the SW, and as a result its precision was affected by ambiguous words. More concretely, an initial evaluation⁴⁵ of Scarlet published in (Sabou et al., 2007) showed 70% precision in obtaining mappings between ontologies. These experiments have also shown that more than half of the invalid mappings are due to ambiguity problems in the anchoring process (i.e., elements of the source ontology were anchored to online ontology using the considered terms with different senses).

For example the matcher retrieved the following matching between two terms from the AGROVOC and NALT ontologies: *game* \supseteq *sport*. However “Game” is a “wild animal” in AGROVOC ontology, while “sport” appears in NALT as “leisure, recreation and tourism” activity. The reason why this invalid mapping was derived is because “game” has been anchored in a background ontology where it is defined as a subclass of “Recreation or Exercise” and a superclass of “sport”.

⁴⁵ This technique was evaluated using the test data sets in the 2006 Ontology Alignment Evaluation Initiative, AGROVOC and NALT. A sample of 1000 mappings were manually validated

Obviously, these ambiguity problems are shared by any other system that needs to find correspondences across heterogeneous sources. Nevertheless, because the above matcher deals with online ontologies found in the open SW, it provides us with a suitable testing scenario that maximizes heterogeneity. In (Gracia et al., 2007) we aimed to solve this problem by introducing techniques from Word Sense Disambiguation, which validate the mappings by exploring the semantics of the ontologies involved in the matching process. Specifically, two technologies were discussed to filter out mappings resulting from the incorrect anchoring of ambiguous terms:

- Approach 1: the system proposed in (Trillo et al., 2007) exploits the ontological context of the matched and anchor terms, by calculating a *similarity measure* to determine the validity of the anchoring. The similarity between the ontological terms and their respective anchor terms (if they represent the same “ontological sense” according to the set of senses obtained using all the ontologies in the SW as background knowledge sources) is measured by analysing their ontological context similarity (the ontological context of a term comprises its superclasses, subclasses, descriptions, properties, etc.) up to a certain *depth* of exploration (number of levels explored in the hierarchy). More details on how this system is used to improve anchoring are given in (Gracia et al., 2007).
- Approach 2: we reused the WordNet based disambiguation introduced by PowerMap to filter semantically sound ontological mappings. In this approach, WordNet based methods are used to elicit the sense of the candidate concepts by looking at the ontology hierarchy, and to check the semantic validity of the mapping between those candidate concepts.

These experiments showed that each of the proposed disambiguation techniques can lead to an important increase in precision, as many results erroneously mapped due to bad anchoring can be detected and filtered, without having too negative impact on recall (as some good mappings could also be erroneously filtered). The purpose of this evaluation is twofold, on the one hand it highlights

the weaknesses and strengths of our PowerMap as a standalone algorithm, and on the other hand, it makes it possible to establish comparisons with the technique proposed in (Trillo et al., 2007). This standalone evaluation complements the general evaluation, presented in Chapter 9, in the context of Question Answering.

6.4.2 Applying PowerMap semantic filtering in the anchoring task

In this section we summarize the experiment we conducted (Gracia et al., 2007) on the use of the PowerMap WordNet based algorithm adapted to determine the validity of the mappings provided by the Scarlet matcher (Approach 2). In this approach we do not perform similarity computation between the ontological terms and the anchored terms in the background ontologies, measured between A and A' (or B and B'), as it is done for Approach 1. Instead, similarity is computed directly between the matched ontological terms A and B .

We compute the WordNet based confidence level 1 for the matching A and B as follows. Given the two ontological terms A and B , let $S_{B,A}$ be the set of those senses of B for which there exists a semantically similar (as defined in Section 6.3.1) sense of A . If $S_{B,A}$ is empty, the mapping B is discarded because the intended meaning of A is not the same as that of the concept B . Finally, the true senses of B are determined by its place in the hierarchy of the ontology. That is, S_B^H consists only of those senses of B that are similar to at least one sense of its ancestors in the ontology. We then obtain the valid senses as the intersection of the senses in S_B^H with the senses obtained in our previous step, $S_{B,A}$. Note that by intersection we mean the senses that are semantically similar, even if they are not exactly the same sense. In case the intersection is empty it means that the sense of the concept in the hierarchy is different from the sense that we thought it might have in the previous step, and therefore that mapping pair should be discarded. The same process is repeated for the term A and its mapped term B . The ontology mapping pair will be selected ($l = 1$) only if there is similarity between at least one pair of senses from the set of valid

senses for A-B and the set of valid senses for B-A. Otherwise, the mapping is rejected ($l = 0$). Note that this method is not appropriate if the term has no representation in WordNet. Therefore if one of the terms to be mapped is not found in WordNet (i.e., “zebrafish”), we left the value l as undetermined.

6.4.3 Analysis of results

In (Gracia et al., 2007) we have explored the application of two different similarity measures: Approach 1, based on the ontological context of terms, and Approach 2, based on WordNet. A final strategy has also been conceived by combining both measures. Our experimental results show that all filtering strategies improve the precision of the system (initially 70%). With Approach 1 it reaches values of 80% in precision with 0.60 effect on recall (effect on recall = 1 means the recall is not influenced and no valid mappings are rejected), the precision can be increased but the effect on recall decreases because more valid mappings are filtered out. The PowerMap WordNet based algorithm in Approach 2 evaluated as correct 70% of valid mappings and 22% of invalid ones, leading to a precision of 88%⁴⁶ and an effect on recall of 0.70. By combining both approaches we can either promote precision (Approach 3.1) or recall (Approach 3.2). Approach 3.1, in which only valid mappings that both methods estimate as valid can pass the filter, reaches a 92% precision, nevertheless it reduces recall almost to one half. On the other hand, Approach 3.2 can reach a precision of 87% but with very good behaviour in recall (overall recall is affected in only a factor of 0.76).

6.4.4 Discussion and WordNet limitations

Summing up, a precision of 92% is the maximum we can reach combining both methods. In (Gracia et al., 2007), it was found, by exploring the invalid mappings that pass our filters, that the number of

⁴⁶ That was the predicted value that precision can reach by improving anchoring according to (Sabou et al., 2007)

negative mappings due to bad anchoring is negligible, having found other types of errors that hamper our methods, as bad modelling of relationships (using for example subsumption instead of part-of relations).

Moreover, the meaning of an ontological concept must be precisely defined in the ontology: both similarity measures need to get the direct parents of the involved terms, but often the ancestor is <http://www.w3.org/2000/01/rdf-schema#Resource>, and therefore the taxonomical meaning cannot be obtained in ontologies with vaguely defined concept names.

The mappings that hamper the method in Approach 1 are the ontological terms which are poorly described in background ontologies, in other words the internal structure of background ontologies was not rich enough for the ontological context based method to filter mappings properly. In Approach 2, the PowerMap based technique takes advantage of the high quality description and coverage of WordNet, combining in a clever way some well founded ideas from traditional Word Sense disambiguation (Wu and Palmer, 1994) (Resnik, 1995).

Additionally, this evaluation also helped us to analyse the drawbacks of exclusively relying on sense information provided by WordNet to compute semantic similarity on ontology concepts. The drawbacks of WordNet are:

1. WordNet does not cover all ontology words, in particular instances, or intended senses, for example, the sense of “developer” as software developer is not reflected in WordNet 3.0. Moreover, ontology classes frequently use compound names without representation in WordNet as such, e.g., “sugar substitutes” corresponds to two WordNet lemmas (“sugar”, “substitutes”). Therefore, in these cases the meaning can be misleading or incomplete.
2. Senses are not related in the WordNet IS-A taxonomy. Some terms considered similar from an ontology point of view, are not connected through a relevant IS-A path in WN, e.g., the term “sweeteners” and its ontological parent “food additive” (*agrovoc*), therefore the

taxonomical sense is unknown. Or, in the worse case, a positive mapping can be rejected, e.g., “food” (in *agrovoc*) is not connected in WordNet through an allowable IS-A path to its mapped term “whipped cream” (in *nalt*).

3. The excessive fine-grainedness of WordNet sense distinctions, which is a frequently cited problem (Ide and Veronis, 1998). For instance, the senses of “crayfish” (*agrovoc*) considering its parent “shellfish” are (1) *lobster-like crustacean usually boiled briefly*; and (2) *warm-water lobsters without claws*, but while considering its mapped term “animal” (*nalt*) the sense is (3) *small fresh water crustacean that resembles a lobster*. The valid mappings (1) and (2) are discarded, as there is no relevant IS-A path connecting them with (3).
4. Computing semantic similarity applying Resnik criteria to IS-A WordNet does not always produce good semantic mappings. Senses connected through a relevant IS-A path may not be good semantically sound mappings (even if they score high on the Wu and Palmer formula). For instance, when computing the similarity between “Berries” and its parent “Plant”, the best sense obtained for “Berry” is “*Chuck_Berry - (United States rock singer)*”, which is unhelpful.

6.5 Summing up

PowerMap provides a mapping from linguistic terms to ontology entities. Given the linguistic triples identified by the Linguistic Component, PowerMap first identifies all the ontologies that are likely to describe the entities on these triples (i.e., those that contain syntactically similar entities). Then, it identifies the sense of the matches, clustering similar ones together, and excluding those that do not match the input linguistic terms semantically. The output of PowerMap is then a set of Entity Mapping Tables, each one corresponding to one linguistic term. Each EMT contains the ontology

elements (drawn from different ontologies) to which the term was matched. PowerMap generates WordNet senses for all classes and individuals in the EMT, if possible.

In this chapter, we also presented an evaluation of the PowerMap WordNet-based semantic component to assess semantically sound mappings in the context of a mapping task. It has proved to give good results, leading to a precision of 88% and an effect on recall of 0.70. Thus, this component represents a contribution on its own, and its algorithm can be re-used to perform disambiguation in different mapping tasks, beyond the original Question Answering purpose.

The senses assigned to each individual by PowerMap are crucial for the PowerAqua Merging and Ranking component, in order to rank the alternative answers coming from semantically different interpretations as it will be discussed in Chapter 9.

Chapter 7 Triple Mapping (step 3)

This chapter is an extended description of the triple mapping component, responsible to obtain a ontology compliant mapping of a user query, which has been published in the Knowledge Capture Conference in 2009 (Lopez et al., 2009c).

7.1 Introduction

Having worked at the level of individual mappings in Chapter 6, the goal of the Triple Similarity Service (TSS) is to identify those Onto-Triples from the various relevant resources that better represent, and jointly cover, a user query and lead to an answer. The TSS studies the relationships in the ontology to identify the meaningful mappings that determine the valid ontological interpretations equivalent to a user query.

As such, the TSS assembles the individual element level matches spread over several ontologies and recorded in the Entity Mapping Tables (these map query terms to ontology elements) to produce triple level matches encoded in Triple Mapping Tables (these map entire Query-Triples to Onto-Triples), as shown in Figure 7.1. In other words, the output is a set of Triple Mapping Tables, where each table relates each Query-Triple to all the equivalent ontology triples obtained from different ontologies that can jointly be used to generate an answer. We distinguish two major stages in the TSS algorithm.

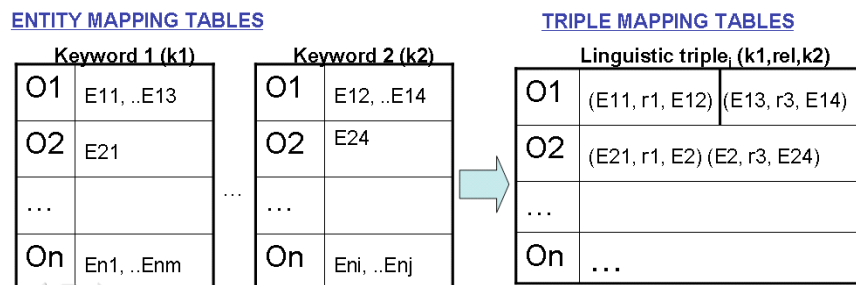


Figure 7.1. From Entity Mapping Tables to Triple Mapping Tables

The first stage of the algorithm is frequently required for queries that are split into more than one Query-Triple and in those cases where there is ambiguity about how the linguistic terms inter-relate. For instance, the query “which projects are led by academics that are related to the semantic web?” is split into two Query-Triples: <projects?, led, academics> <projects / academics, related, Semantic Web>. However, given the linguistically ambiguous way in which the second part of the query is asked, it is not clear whether the subject in the second triple refers to academics or projects. The TSS requires domain knowledge to solve the modifier attachment and disambiguate the shared term among triples. In order to analyse all possibilities within the relevant ontologies the second Query-Triple is modified to <projects/academics, related, Semantic Web>. As a result of this ambiguity, the second stage of the algorithm will search for Onto-Triples matching either <projects, related, Semantic Web> or <academics, related, Semantic Web>. If both query terms (projects and academics) produce valid Onto-Triples (in the same or different ontologies) for the second Query-Triple, the merging and ranking component will merge or rank the different interpretations.

The second stage of this algorithm performs its core functionality of transitioning from term level mappings (in the EMT) to triple level mappings (in the TMT). The algorithm, detailed in Section 7.2 has been optimized towards finding the most precise ontological translations and its design has been influenced by the following observations:

- An ontology with a higher coverage of a Query-Triple (i.e., one that covers entire Query-Triples and not just individual terms) is likely to lead to a better result.
- If no Onto-Triples can be found for a Query-Triple containing a compound term, this might not be the case for its parts. Potentially relevant Onto-Triples could still be found for the individual elements of the compound. Therefore, the TSS is re-invoked with new Query-Triples formed by splitting the compound term to search for Onto-Triples that could not be found by using the entire compound.

- We observed that the subject (QT_{i1}) of a Query-Triple frequently refers to a more general ontological entity than its object (QT_{i3}), as the objects are often mapped to instances or individuals while the subjects, which represents the *wh-query term* or the variable (the type of the answer) of the question, are mapped to classes. For example, in “who employs Enrico Motta?”, “which international organizations participate in humanitarian aid?”, “which Russian rivers flow into the Black sea?”, “what diseases have symptoms of hair loss?” the QT objects (“Enrico Motta”, “humanitarian aid”, “Black sea”, “hair loss”) are the most informative elements in the sentence— see the example queries in Appendix A and the user queries gathered in the evaluations presented in Chapter 9. Therefore, splitting QT_{i1} (e.g., into “Russian” and “rivers”) has less negative influence on the quality of the final ontology triples than splitting QT_{i3} (which is more likely to introduce noise, e.g., “Black” and “sea”). Moreover, if no ontology covers the whole triple, even when all compounds are split (subject first, object second), only ontologies that cover QT_{i3} are considered.
- The TSS algorithm is detailed in Section 7.2 , in Section 7.3 we describe the optimisations performed in the TSS to scale to a large scenario, the known issues and limitations are listed in Section 7.4, and a summary is presented in Section 7.5 .

7.2 The Triple Similarity Service Component

7.2.1 TSS algorithm

To avoid exploring an unfeasibly large space of possible solutions for a user query, the algorithm sequentially iterates through a series of steps. The TSS contains four steps that parallel the previously described three observations (Section 7.1) and lead to decreasingly precise translations (but increasingly higher recall). The TSS executes the highest quality steps first and only uses inferior quality steps if no answer is found (see Figure 7.2 and algorithm below).

In what follows we illustrate the behaviour of the algorithm, while a number of examples can be seen in the next Section.

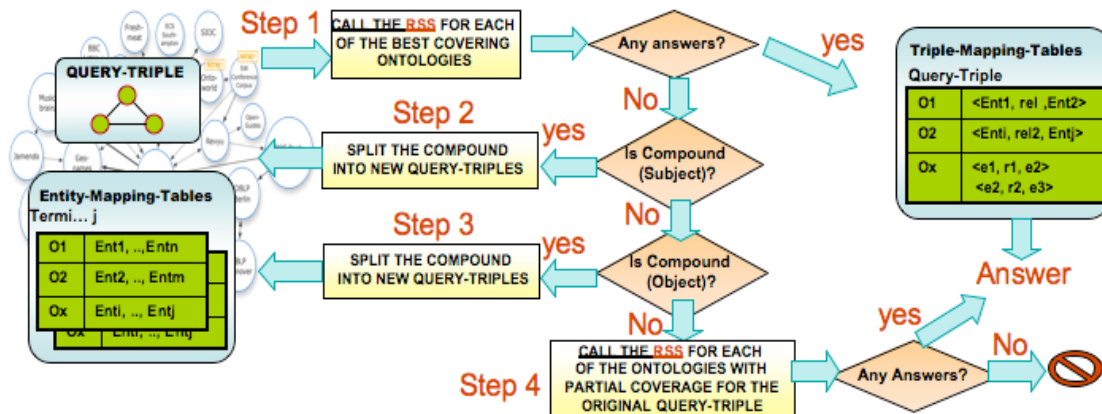


Figure 7.2. The TSS flow step by step

For each Query-Triple (QT_i), in step **S1**, TSS inspects all ontologies that contain mappings for at least two of the Query-Triple terms (including the object QT_{i3}). This coverage-centric criterion ensures that the algorithm focuses first on the ontologies most likely to address the domain of the query. The Relation Similarity component, RSS (explained in detail in 7.2.3), is called for each ontology in order to find the concrete Onto-Triples, which match the input Query-Triple. If any of these Onto-Triples leads to an answer, then they are recorded in the TMT. After all potentially relevant ontologies have been inspected, if at least one answer has been derived, then the algorithm stops (lines 2-8 in Figure 7.3)⁴⁷.

Otherwise, in step **S2**, the TSS increases recall by splitting the subject QT_{i1} , if it is a compound, and the TSS re-iterates and calls the RSS for the resulting Query-Triples (obtained after splitting the subject). At the end of this phase, if an answer has been obtained then the algorithm stops, otherwise it continues (lines 9-12).

⁴⁷ We have seen in the illustrative example in Chapter 4 how using the coverage criterion in “Who play in the rock group nirvana?”, the algorithm chooses ontologies related to music instead of an ontology about spiritual stages.

In step S3 we resort to splitting the object QT_{i3} and re-invoke the TSS for the resulting Query-Triples (lines 13-16). Finally, if none of the above strategies leads to an answer, recall is further improved in step S4 by inspecting ontologies that only cover the original (non split) QT_{i3} . In this case, the RSS is only called for this term and it returns all Onto-Triples, containing a match for this term, which partially cover the Query-Triple (lines 17- 24).

```

1: for all Query-Triplei do
2:   InitializeTMT(Query-Triplei)
3:   for all Oj with cover(Oj; QTi) ≥ 2 do
4:     Onto-Tripleij = RSS(QTi1; QTi2; QTi3)
5:     for all Onto-Tripleij with Onto-Tripleij.hasAnswer() then
6:       recordInTMT(Query-Triplei, Oj, Onto-Tripleij)
7:     end for
8:   end for
9:   if TMT(Query-Triplei) = Empty then
10:    if QTi1.isCompound() then
11:      resultingTriples = Split(QTi1)
12:      TSS(resultingTriples)
13:    else
14:      if QTi3.isCompound() then
15:        resultingTriples = Split(QTi3)
16:        TSS(resultingTriples)
17:      else
18:        for all Oj with cover(Oj; QTi3) = 1 do
19:          Onto-Tripleij = RSS(QTi2; QTi3)
20:          for all if Onto-Tripleij with Onto-Tripleij.hasAnswer() then
21:            recordInTMT(Query-Triplei, Oj, Onto-Tripleij)
22:          end for
23:        end for // if TMT(Query-Triplei) = Empty then NoAnswers
24:      end if
25:    end if
26:  end if
27: end for

```

The diagram illustrates the TSS algorithm with four main steps indicated by brackets on the right:

- S1** (Lines 1-8): Initialization and first pass through ontologies with $\text{cover} \geq 2$.
- S2** (Lines 9-12): Handling compound query triples by splitting QT_{i1} and re-invoking TSS.
- S3** (Lines 13-16): Handling compound query triples by splitting QT_{i3} and re-invoking TSS.
- S4** (Lines 17-24): Handling non-compound query triples by finding ontologies with $\text{cover} = 1$ and re-invoking RSS.

Figure 7.3. The TSS algorithm step by step (the input for the TSS are the Query-Triples and the EMTs)

7.2.2 TSS Illustrative examples

As explained earlier, the TSS uses the Relation Similarity Service (RSS) to inspect each selected ontology and to identify the ontological triples that are appropriate matches for the given Query-Triple. Table 7.1 contains four examples of TSS outputs, which we will use in what follows to illustrate the ways that answers are derived at each step of the algorithm.

Let's consider query Q1 solved in step **S1**: “Who works in the climaprediction project?”, whose corresponding Query-Triple is <person / organization?, works, climaprediction project>. The TSS algorithm focuses first on ontologies that cover most terms in the Query-Triple, because there exists a covering ontology with mappings for both “person” and “climaprediction” (as an instance of “project”), which contains valid ontological triples (no correct mappings were found for the relation “work” in this case). Therefore, an answer can be inferred at step S1. The query compound (climaprediction project) is not split in this step if there are ontological mappings for the compound as such, within at least one covering ontology, or, as in this case, the individual terms hold a direct IS_A relationship, i.e., “climaprediction” is an instance of “project” in the *KMi portal ontology* (see Figure 7.4). However, if a query is formed with a compound term that has no ontological mappings, the only course of action left is to split the compound into its individual elements and modify the Query-Triple accordingly before calling the RSS. For example in Q2: “Provide me information on the Bengal cat breeders” where there are no mappings for the query term “Bengal cat breeders” in any ontology, in order to find candidate mappings, which can potentially answer a query, the triple is split from: <what-is, ?, Bengal cat breeder?> into <Bengal cat, ?, breeders>. An answer can be inferred at step S1 because there is a covering ontology, *TAP*, that contains a valid Onto-Triple (<Breeder, has_breeder, Bengal_Cat>) able to generate an answer (see Figure 7.5), and therefore the algorithm stops.

However, if after executing step S1 none of the covering ontologies produces any valid ontological triples from which an answer can be inferred and the subject of the query is a nominal compound, this one is split into its parts in step S2. Then, the algorithm re-iterates again, calling the RSS with the new set of triples. For instance, query Q3: “Which Russian rivers flow into the Black sea?”, whose corresponding Query-Triple is $\langle \text{Russian rivers?}, \text{flow}, \text{Black Sea} \rangle$, is answered in step S2. The reason is that while there is a mapping for “Russian Rivers” as the name of a pub in an ontology about restaurants in Texas, no valid Onto-Triples were produced in step S1 as there are not covering ontologies for both arguments in the query. Therefore, the algorithm reiterates again in step S2 by splitting the compound term, and consequently modifying the Query-Triple into: $\langle \text{Russian / rivers?}, \text{flow}, \text{Black sea} \rangle$ and creating a new Query-Triple for the compound $\langle \text{Russian}, \text{?}, \text{rivers} \rangle$. For these resultant Query-Triples, in this second TSS iteration, a covering ontology containing the valid Onto-Triples that produce an answer is found by the RSS. The RSS analyses the ontology relations to disambiguate which part of the compound “Russian” or “Rivers” (or both) links to “flow into the black sea” in order to create valid Onto-Triples - in this case “Rivers” as shown in Figure 7.6 (note that different ontologies may link the terms in different ways, if both compound parts produce valid Onto-Triples the different interpretations will be ranked or merged at a later stage). If the object of the Query-Triple is a compound term and the previous step failed to generate an answer, then similarly to step S2, in step S3 the nominal compound QT_{i3} is split and the TSS re-iterates over the newly obtained triples.

In the case of query Q4: “What pathologies produce hair loss?”, there are several ontologies with mappings for both “pathologies” and “hair loss”, which do not contain links between the entities. Therefore steps S1, S2 and S3 fail to derive an answer using the covering ontologies, even after splitting “hair loss”. In step S4, the only course of action left is for the TSS to inspect all partial covering ontologies that contain at least one mapping for “hair loss” (QT_{i3}). The RSS completes the

triple by looking at the ontological information in the form of relationships to and from the object QT_{i3} . One such ontology is the *biomedical* ontology that contains the term “alopecia” (a medical term for “hair loss”) linked to the class “diseases” and produces a set of potential answers: hyperthyroidism, hypothyroidism, poisoning, trichotillomania, chemotherapy, mycotic infections, stress and iron deficiency, as shown in Figure 7.7.

Table 7.1: Examples of mappings for different queries (Triple Mapping Tables)

Q1: Who works in the climaprediction project?: <person/org.?, works, climaprediction project>		
KMi	<working-person, subclass, person> <working-person, has-project-member, climaprediction-net>	
	<working-person, subclass, person> <working-person, has-project-leader, climaprediction-net>	
Q2: Provide me information on the Bengal cat breeders: <Bengal cat, ?, breeders>		
TAP	<breeder, has_breeder, Bengal_cat>	
Bengal	<sunset_bengal_cats, IS_A, animal_breeders>, <Cukkla_bengal_cats, IS_A, animal_breeders>, etc.	
Q3: Which Russian rivers flow into the Black Sea?: <rivers/Russian?, flow, Black Sea> / <Russian, ?, rivers>		
RussiaB	<river, flow-to, Black_Sea>	<river, has_major_river, country> <country, has_political_fact, russian>
KIM	<>	<river, part of, entity> <entity, has Alias, Russian Soviet Federated Socialistic Republic>
Q4: What pathologies produce hair loss?: <pathologies?, produce, hair loss>		
Biomedical	<hyperthyroidism, hasSymptom, Alopecia_1> , <stress, hasSymptom, Alopecia_1>, etc	

Question Answering

Ask a query

who works in the climaprediction project?

Ask!

Examples

Make Use of Watson

Click here to show linguistic triples, individual mappings and ontologies in use

You are not logged in | [Log in](#)

[Group identical answers](#)

1 ontologies found with answers for the query triple: [person, organization] -- works -- climaprediction project

2 mappings in <http://plainmoor.open.ac.uk:8080/sesame/ksw-kb>

Mapping 1 (ranking: 1):

[person](#) (person equivalentMatching)

[has-project-member](#) (has-project-member ontology_ad_hoc)

[climaprediction-net](#) (climaprediction-net equivalentMatching)

3 answers

[john-dominique](#) (John Domingue)

☆

.

.

.

.

.

.

[martin-dzbor](#) (Martin Dzbor)

☆

.

.

.

.

.

.

[enrico-motta](#) (Enrico Motta)

☆

.

.

.

.

.

.

Save trust values

Mapping 2 (ranking: 1):

[person](#) (person equivalentMatching)

[has-contact-person](#) (has-contact-person ontology_ad_hoc)

[climaprediction-net](#) (climaprediction-net equivalentMatching)

1 answers

[enrico-motta](#) (Enrico Motta)

☆

.

.

.

.

.

.

Save trust values

Show all element mappings

Find these answers in Yahoo

Get discarded Triples!

Total Time to answer this query: 10.609 secs

Figure 7.4 Screenshot for Q1: “Who works in the climaprediction project?”

Question Answering

Ask a query

Provide me information on the bengal cat breeders

Ask!

Examples

Make Use of Watson

Click here to show linguistic triples, individual mappings and ontologies in use

[Group identical answers](#)

2 ontologies found with answers for the query triple: [bengal cat] -- null -- breeders

2 mappings in <http://kmi-web07.open.ac.uk:8080/sesame/tapfull>

Mapping 1 (ranking: 1):

[Breeder](#) (Breeder synonym)

[has_breeder](#) (has_breeder ontology_ad_hoc)

[Bengal_Cat](#) (data:Bengal_Cat equivalentMatching)

108 answers

[Bakhutan_cattery](#) (Bakhutan_cattery)

[bengal_cat](#)

☆

.

.

.

.

.

[Anjali_bengals](#) (Anjali_bengals)

[bengal_cat](#)

☆

.

.

.

.

.

[Hits_the_spot](#) (Hits_the_spot)

[bengal_cat](#)

☆

.

.

.

.

.

[See_spots](#) (See_spots)

[bengal_cat](#)

☆

.

.

.

.

.

[Purrfect_bengal_cats](#) (Purrfect_bengal_cats)

[bengal_cat](#)

☆

.

.

.

.

.

[Ramatut_bengal_cats](#) (Ramatut_bengal_cats)

[bengal_cat](#)

☆

.

.

.

.

.

Figure 7.5 Screenshot for Q2: Provide me information on the Bengal cat breeders

163

Question Answering

Ask a query [Examples](#)

Make Use of Watson ☐

You are not logged in | [Log in](#)

Mapping 1 (ranking: 1):
[river](#) (river synonym) [DEFAULT_ROOT_RELATION](#) (DEFAULT_ROOT_RELATION ontology_ad_hoc) [Black_Sea](#) (Black_Sea equivalentMatching)
1 answers
[Dnepr](#) (Dnepr) ☆

Mapping 2 (ranking: 1):
[river](#) (river synonym) [flow_to](#) (flow_to ontology_ad_hoc) [Black_Sea](#) (Black_Sea equivalentMatching)
1 answers
[Dnepr](#) (Dnepr) ☆

1 ontologies found with answers for the query triple: [russian] -- null -- rivers

1 mappings in <http://kmi-web07.open.ac.uk:8080/sesame/russiaB>

Mapping 1 (ranking: 1):
[river](#) (river synonym) [has_major_river](#) (has_major_river ontology_ad_hoc) [country](#) (country ontology_ad_hoc)
[country](#) (country ontology_ad_hoc) [has_potitical_fact](#) (has_potitical_fact ontology_ad_hoc) [russian](#) (russian equivalentMatching)
4 answers
[Neva](#) (Neva) [Russia](#) ☆
[Don](#) (Don) [Russia](#) ☆
[Volga](#) (Volga) [Russia](#) ☆
[Dnepr](#) (Dnepr) [Russia](#) ☆

Figure 7.6 Screenshot for Q3: “Which Russian rivers flow into the Black Sea?”

Question Answering

Ask a query [Examples](#)

Make Use of Watson ☐

You are not logged in | [Log in](#)

☐ [Click here to show linguistic triples, individual mappings and ontologies in use](#)
The ontologies with better coverage did not produce any valid ontology triple to generate an answer.
[Group identical answers](#)

1 ontologies found with answers for the query triple: [pathologies] -- produce -- hair loss

9 mappings in <http://kmi-web07.open.ac.uk:8080/sesame/biomedical>

Mapping 1 (ranking: 1):
[Hyperthyroidism](#) (Hyperthyroidism ontology_ad_hoc) [hasSymptom](#) (hasSymptom ontology_ad_hoc) [Alopecia_1](#) (hair loss equivalentMatching)
☆

Mapping 2 (ranking: 1):
[Hypothyroidism](#) (Hypothyroidism ontology_ad_hoc) [hasSymptom](#) (hasSymptom ontology_ad_hoc) [Alopecia_1](#) (hair loss equivalentMatching)
☆

Mapping 3 (ranking: 1):
[poisoning](#) (poisoning ontology_ad_hoc) [hasSymptom](#) (hasSymptom ontology_ad_hoc) [Alopecia_1](#) (hair loss equivalentMatching)

Figure 7.7 Screenshot for Q4: “What pathologies produce hair loss?”

7.2.3 The Relation Similarity Service

The *Relation Similarity Service* (RSS) is core to the TSS. The RSS is invoked by the TSS for each ontology relevant to each linguistic triple. Its role is to inspect an ontology and to identify the Onto-Triples that are appropriate mappings for the given Query-Triple, from which an answer can potentially be inferred. It preferably uses exact mappings (singular or plural variations are also considered exact mappings) to obtain the Onto-Triples; otherwise equivalent mappings and synonyms are selected, leaving the use of hypernyms and hyponyms as the last choice.

The RSS can map a Query-Triple to either one Onto-Triple (direct mapping) or to two Onto-Triples (indirect mappings). Depending on the type of their predicate, direct mappings can be “IS-A” (a subsumption relation) or *domain* relations (any other relation, considering also relations inherited from the superclasses). For example, for the Query-Triple <city?, ?, USA>, the ontological triples <city, attribute_country, USA>, <city, isIn, USA> are domain direct mappings while <city, isCityOf, state> <state, isStateOf, USA> is an indirect mapping. Domain direct relationships between the arguments of the triple are analysed before “IS-A” unless the original question contains an IS-A relation (an indication that such a relation is expected). For example, for the query “which animals are reptiles” the answers are encoded as the subclasses of the class “reptile”, which is a subclass (IS-A) of the class “animal”.

A typical situation the RSS has to cope with is one in which the structure of the intermediate query does not match the way the information is represented in the ontology (i.e., the subject and object of a Query-Triple are not necessarily mapped to those in the ontological triple). Here we present a few representative examples to illustrate the RSS algorithm:

Case 1: The ontology does not contain a match for the linguistic relation. This is a rather typical case, either because the linguistic relation is implicit (like in “find me cities in USA”) or a “IS-A” type (like in “which animals are reptiles?”), or because the ontology relations have labels that are

difficult to detect by syntactic techniques (like in “who works in the climaprediction project?” where “work” should map to both ontology relations: “has-project-member” and “has-contact-person”). The problem becomes one of finding domain relations that link the two terms, whenever there are successful matches for both arguments⁴⁸. The algorithm is as follows:

- Finding a domain relation that links the two terms: superclasses and subclasses are also considered due to the inheritance of relations in the subsumption hierarchy. For instance, in “who works in the climaprediction project?” the relevant relations can be defined only for researchers, students or academics, rather than people in general. To avoid ending up with ontological triples that produce no answers, the relations are restricted to only those that can lead to results.
- If no domain relations are found then “is-a” relations are inspected. Unless the linguistic relation is marked as an “IS-A” type, then IS-A relations are inspected before domain ones.
- If such relations are not found then indirect mapping relations are inspected, as in the example “Find me cities in USA” which maps to two Onto-Triples through one mediating concept “state” <city, isCityOf, state> <state, isStateOf, USA>.

The resultant ontological triples for the above examples are presented in Table 7.2 and Figure 7.8

⁴⁸ Otherwise, if only one argument is matched partial translations are obtained selecting the domain relations from and to the matched term or any of its superclasses.

Figure 7.8. Answers from different ontologies to the query “find me cities in USA”

Table 7.2. Examples of ontological translation for different queries in Case 1

Find me cities in USA: <cities?, ?, USA>	
TAP	<city, hasCapitalCity, countryUnitedStates>
Sweto	<city, attribute_country, USA (literal)>
Utexas	<city, isCityOf, State (class)> <State, isStateOf, USA>
Which animals are reptiles?: <animals?, IS-A, reptiles>	
TAP, Galen	<reptile, subClassOf, animal>
coastalOntology	<aquatic-reptiles, subClassOf, aquatic-organisms (<i>hypernym</i>)>

Case 2: the ontology contains a set of candidate ontology entities for the linguistic relation.

Here, if the candidate entity is a property, the matching and joining of triples is controlled by the domain and range information of the relation and the type of the mapped ontological arguments. In the case of *wh-queries*, where there is no information about the type of the *wh-query term*, the RSS identifies the ontology relationships that are valid for the mapped term, to obtain the set of candidate

values or ontological terms for the *wh-query term* that can complete the triple. For example, in “what is the diet of the manatee?” the mapped property “has-diet” links the mapped instance “manatee” to the class “DietFood”, which becomes the *ad-hoc wh-query term* of the Onto-Triple (see screenshot in Figure 7.9). Otherwise, if the candidate entity for the linguistic relation is a class (or instance), it acts as a mediating entity in the path to link both arguments, like in “where are mountains in Asia?” where the QT relation “mountains” is mapped to an ontological class that links between a location (where) and Asia. In the case where there are candidate mappings for both the arguments in the triple and the relation, but no valid triples are obtained, the RSS ignores the relation name and initiates a search for ontological triples linking the arguments only (Case 1). The rationale behind this approach is that a relation’s meaning is often given by the type of its domain and range rather than by its name. The resultant ontological triples are presented in Table 7.3

Table 7.3: Examples of ontological translation for different queries in Case 2

What is the diet of the manatee?: <what-is, diet, manatee>	
TAP	<DietFood, has_diet, manatee>
Where are mountains in Asia?: <location/state?, mountains, Asia>	
KIM	<location, part of, mountain> <mountain, locatedIn, Continent_T.2 (Asia)>

Ask a query [Examples](#)

Make Use of Watson ☐

You are not logged in | [Log in](#)

[Click here to show linguistic triples, individual mappings and ontologies in use](#)

[Group identical answers](#)

1 ontologies found with answers for the query triple: [what_is] -- diet -- manatee

1 mappings in <http://kmi-web07.open.ac.uk:8080/sesame/tapfull>

Mapping 1 (ranking: 1):
[DietFood](#) (DietFood ontology_ad_hoc) [has_diet](#) (has_diet equivalentMatching) [Manatee](#) (Manatee equivalentMatching)

3 answers

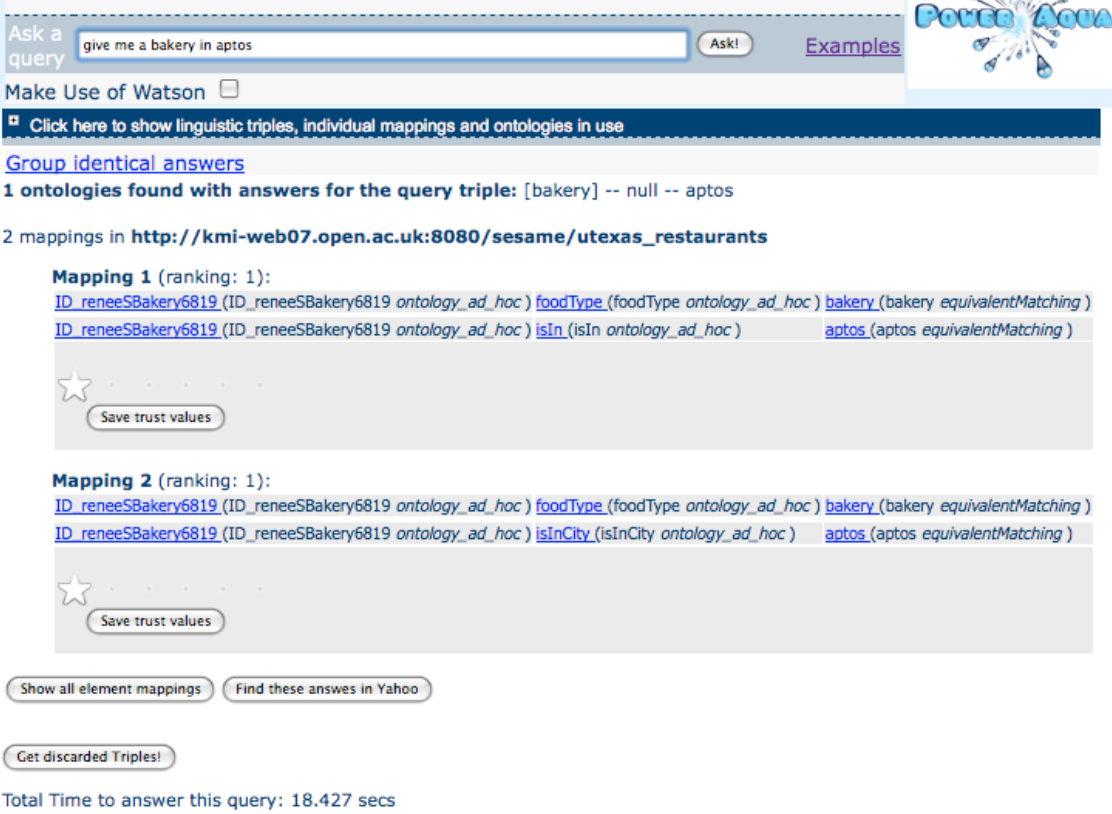
[turtle_grass](#) (turtle_grass) ★

[mangrove](#) (mangrove) ★

[algae](#) (algae) ★

Figure 7.9. Answer to the query “What is the diet of the manatee?”

Case 3: Handling literals. A special situation is when the RSS has to handle relevant mappings that are literals. Answers are generated in two different ways depending on how they are related with the literal, as exemplified next. The simplest case is found in the query “Find me cities in Spain”, where for the SWETO ontology “Spain” corresponds to a literal. Therefore, the answer is generated by looking for all the instances of the mapped class “city” that have “Spain” as the value of one of its attributes. The second case is found in “Give me a bakery in aptos?” where the answer is an instance of the class “restaurant” which contains a relation to the instance “aptos” and another “foodType” to the literal bakery (see screenshot in Figure 7.10). Analysing literals can be a time consuming task, for example, we have literals matches for the word “Spain” for each instance representing a Spanish city in the SWETO ontology.



Ask a query [Examples](#)

Make Use of Watson ☐

[Click here to show linguistic triples, individual mappings and ontologies in use](#)

[Group identical answers](#)

1 ontologies found with answers for the query triple: [bakery] -- null -- aptos

2 mappings in http://kmi-web07.open.ac.uk:8080/sesame/utexas_restaurants

Mapping 1 (ranking: 1):

[ID_reneeSBakery6819](#) ([ID_reneeSBakery6819 ontology_ad_hoc](#)) [foodType](#) ([foodType ontology_ad_hoc](#)) [bakery](#) ([bakery equivalentMatching](#))

[ID_reneeSBakery6819](#) ([ID_reneeSBakery6819 ontology_ad_hoc](#)) [isIn](#) ([isIn ontology_ad_hoc](#)) [aptos](#) ([aptos equivalentMatching](#))

☆

Mapping 2 (ranking: 1):

[ID_reneeSBakery6819](#) ([ID_reneeSBakery6819 ontology_ad_hoc](#)) [foodType](#) ([foodType ontology_ad_hoc](#)) [bakery](#) ([bakery equivalentMatching](#))

[ID_reneeSBakery6819](#) ([ID_reneeSBakery6819 ontology_ad_hoc](#)) [isInCity](#) ([isInCity ontology_ad_hoc](#)) [aptos](#) ([aptos equivalentMatching](#))

☆

Total Time to answer this query: 18.427 secs

Figure 7.10. Answer to the query “Give me a bakery in aptos”

Summing up, the RSS tries to make sense of the input linguistic triple through the analysis of relationships between the ontological mappings for the linguistic terms. As one ontology can contain more than one matched ontological term for the same linguistic term, the RSS needs to obtain all possible ontology compliant triple combinations that match the linguistic triple. As a result a linguistic triple can be mapped to more than one Onto-Triple within an ontology, each one being a complete alternative translation of the linguistic triple, or to partial translations that combined together cover the whole linguistic triple. Answers are extracted for each Onto-Triple as the list of instances in the ontology that satisfy the Onto-Triple (*subject-predicate-object*). As explained in Chapter 4.4.1, each ontology is associated to a plug-in which implements its functionality according to the platform where the ontology is stored (e.g., in the form of SPARQL queries for a Virtuoso end-point, SERQL for Sesame, or through the Watson API).

7.3 Efficiency of the TSS

By and large the triple and relation similarity services are designed to avoid expensive computations in those scenarios where simple methods are able to yield solutions. We performed optimizations to the PowerAqua similarity services, not just to improve the algorithms to exhibit better performance, but also to obtain a tighter interaction between PowerAqua and the Watson system, which made it possible for us to compete in the first *Billion Triple Challenge 2008*⁴⁹ (D'Aquin et al., 2008b). To obtain these improvements, PowerAqua was coupled with a new instance of Watson that was produced relying on indexes generated on top of the billion triple dataset. The main modifications made to PowerAqua's similarity services aiming to reduce the number of expensive calls to query the ontologies and therefore improve performance are:

⁴⁹ See <http://challenge.semanticweb.org> for more details on this competition.

(1) Instead of operating in a strictly sequential phase, by first collecting all candidate ontological entities for the linguistic terms in a query and then identifying relevant relationships between them, the algorithm can re-iterate through the two different phases in order to look only for the mappings needed in the first instance. This is useful in the case of compound terms, where the algorithm would look for mappings for terms decomposing the compound only if required (step 2 and 3). For example, for the query in the previous examples “What pathologies produce hair loss?”, the algorithm first finds mappings for “pathologies”, “produce”, “hair loss” and their lexically related variations. Because a valid ontological triple can be found in the biomedical ontology with the previous mappings, it does not attempt to find mappings for the individual terms “hair” and “loss”⁵⁰.

(2) The time consuming process of analysing indirect relationships in the RSS (i.e., relationships which require two triples to be joined, as the length of the path is limited to one mediating concept) is only carried out in those cases when no satisfactory is-a or domain direct relationship between any of the candidate entities within the same ontology is found. Unfortunately, domain and range information is not always explicitly defined in the ontologies’ schema, and it has to be discovered by looking at the instantiation (relationships among instances), which has a negative effect on performance.

(3) In some cases the TSS algorithm can use semantic information to disambiguate how the triples link to each other (modifier attachment) and therefore minimize the number of triple combinations to be analysed in order to translate a query. For instance, in “Which cities are located in the region of Sacramento?”, whose corresponding QT are: <cities?, located, region> <region, ?, Sacramento>, the TSS in step S1 finds an ontology stating that “SacramentoArea IS-A region”.

⁵⁰ Splitting nominal compounds depending on the ontology can be very noisy and time consuming, for instance a compound like “hair loss” will generate many irrelevant mappings for “hair” and “loss”. Thus, compounds are only split if the compound as such has no ontological match that can be used to generate an answer.

Using such semantic information, it merges the two QTs into just one: <cities?, located, Sacramento Area> (see Figure 7.11).

(4) Finding a compromise between recall and performance: In the case that the TSS can not find any valid Onto-Triples in any of the covering ontologies (i.e., these that cover at least two of the terms in a linguistic triple: *predicate-object* or *subject-object*), even when compounds are split, then it tries to find ontological triples considering only mappings for one of the linguistic terms, in particular the *object*. However, when trying out this possibility, the system only looks for straight identity matches and does not use any additional information, such as synonyms.

(5) In close relation with the previous point, the queries (and consequently the amount of reasoning) that PowerAqua can perform over the semantic data in a reasonable amount of time is largely imposed by the performance and capabilities of the ontology repositories and storage platforms. Currently, a query can be translated into various Query-Triples, where each triple is covered by one or more ontologies. However, for performance reasons, we imposed the limitation that each individual Query-Triple can have alternative translations into many ontologies as long as each one represents a complete translation of the Query-Triple. For example, in the query “In which country is Mizoguchi?”, with Query-Triple <country?, ?, Mizoguchi>, as no ontology covers the whole Query-Triple, currently only translations using the object of the query are selected in step S4 of the TSS: <subject?, relation?, Mizoguchi>. A future extension to the TSS algorithm, which is currently computational expensive, is that when only partial translations belonging to different ontologies are found for a given Query-Triple, the TSS can go further in order to obtain a complete translation from partial translations across ontologies. That requires to analyse on the fly whether there is an object in one ontology and a subject in another ontology such that they represent the same individual and link the terms in the Query-Triple: <country, relation1?, object?> <subject?,

relation2?, Mizoguchi>, e.g., to infer the fact that Riichiro Mizoguchi is affiliated to the Osaka University from Ontology 1 and that Osaka is in the country Japan from the Ontology 2.

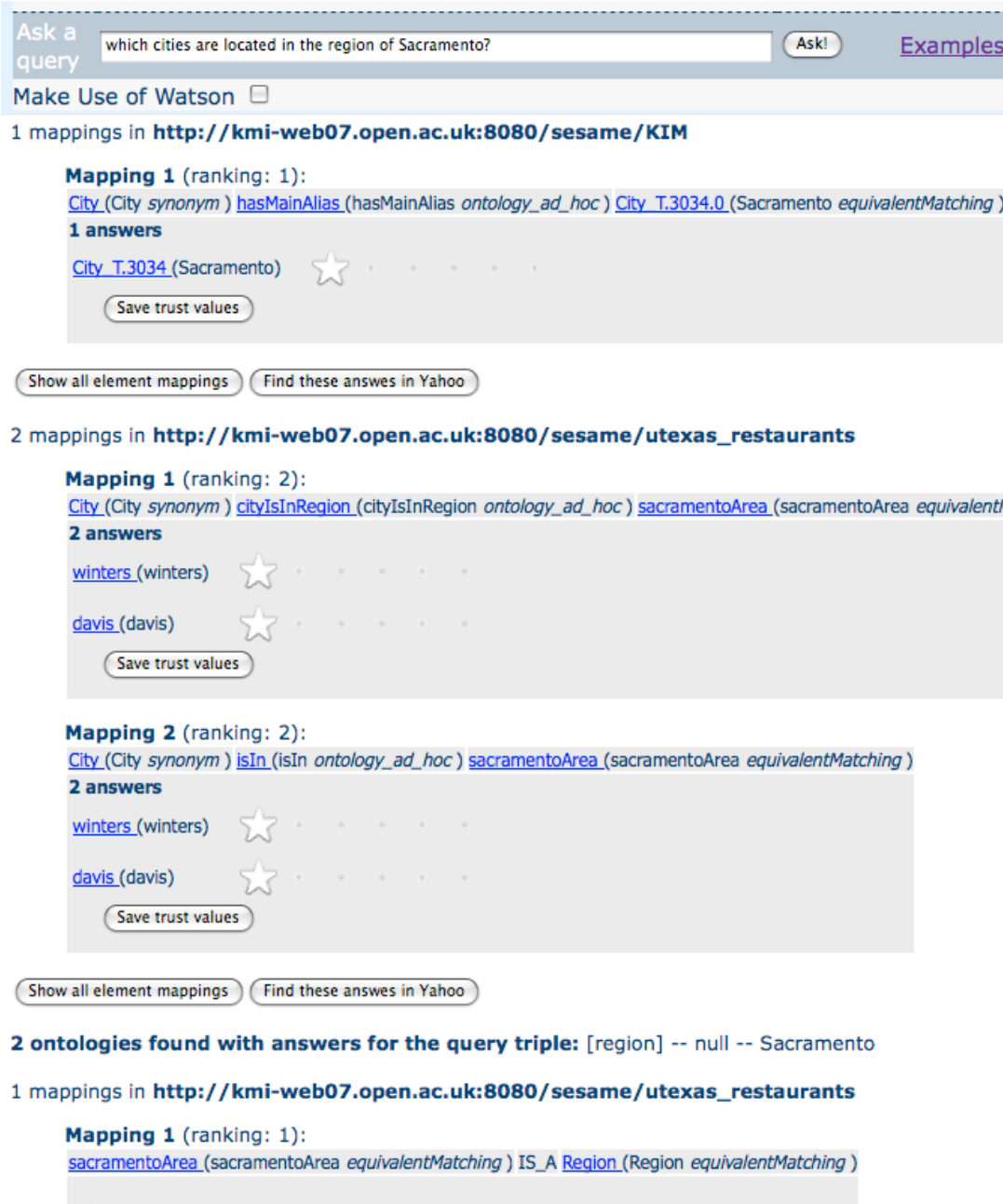


Figure 7.11. Screenshot for the query “Which cities are located in the region of Sacramento?”

Apart from querying selected online repositories, PowerAqua can find and retrieve answers from any of the datasets provided by Watson. Watson interacts with applications like PowerAqua through an API, which relies on several Web services allowing remote calls to its functionalities.

PowerAqua's algorithm has also been modified to achieve a tighter integration and better performance with Watson in the following way:

(1) The number of mappings returned by Watson is reduced through a functionality, provided by the Watson API, used to restrict the mappings for a given term to the ontologies that also contain mappings for another given term. PowerAqua can use this functionality when it needs to apply the coverage criterion. For instance, for the query "Which rivers flow into the Black Sea", PowerAqua looks for mappings for "Black Sea" in the usual way. However, it looks for ontological mappings for "rivers", "flow", and their lexically related words, only in ontologies in which there are also mappings for the second term "Black Sea" or its lexical variations. Experimentally we showed that this reduces drastically the number of candidate mappings returned by Watson without affecting recall.

(2) Watson's performance has also been improved with additional functionalities to better fit the needs of applications like PowerAqua. For example, Watson returns results ranked according to Lucene string similarity measures. This basic ranking mechanism allows PowerAqua to control the amount of information it wishes to inspect in detail, e.g., by selecting only results above a certain threshold.

7.4 Known Issues

Summing up, the TSS tries to find an answer by re-iterating through the space of solutions, augmenting the search space at each iteration until either an answer is found, or all the compounds (if any) are split, and all the ontologies with weak coverage have been analysed. The TSS algorithm maximizes recall at each step, which may lead to a decrease in accuracy and an increase in execution time. The TSS has a couple of "blindspots" where it returns noisy or incomplete answers, even though correct answers exist.

Noisy answers are typically produced when the knowledge encoded on the SW only covers part of the user query (or the user vocabulary could not be fully mapped to the ontologies) and the algorithms have to resort to ontologies with less coverage to generate an answer. In this case irrelevant results may be produced, as PowerAqua cannot fill in the missing information in order to fully understand (map) the query. For example, in “Who are the professors in Southampton?” for which there are no mappings for “professor” in the selected ontologies (only mappings for both “person” and “Southampton” were found), the algorithm returns the persons who were born and died in Southampton from the *DBpedia* ontology about people, and the persons who are part of the University of Southampton, from the *ISWC* ontology. The latter set contains “Nigel Shadbolt”, one of the answers we were looking for, but also false answers. Another example is the query “Which animals live in water?” where there is not enough context in the SW for PowerAqua to discover that both ontological classes “aquatic_animals” and “marine_mammals” from the *fao-agrovoc* ontology represent animals that live in water.

Incomplete answers can sometimes be produced because of the different heuristics used to get better response times, like: (1) the use of filtering algorithms to limit the amount of candidate ontological entities within an ontology, (2) the use of iterative algorithms that do not split compounds if an answer is found for the compound as such, and (3) the restriction that prevents PowerAqua from searching chains of relations, to avoid the explosion of the search space and computational expensive queries. For example, the query “Find me cities in Europe” is successfully answered by various ontologies but not all the ontologies containing valid answers are discovered. For instance, the relevant ontology KIM is missing because it contains an exact mapping “Europe”, and consequently the inexact mappings in the same ontology (“central_europe”, “eastern_europe”, “poland”, “country_holland”, etc.) are discarded. As a result, valid triples with more than 1 indirect relation are missed in the *KIM* ontology, such as *<Warsaw (city), locatedIn, Republic of Poland>*

<Republic of Poland, part of, central Europe> <central Europe, locatedIn, Europe>. Moreover, for the same query valid triples are also missed in the SWETO ontology, because the mapped entity “Europe” is not connected to the list of countries, and we can not find triples like *<city, attribute-country, Poland>*, where the European countries are represented as literals. This last case relates more to the quality of ontologies than to performance issues.

The similarity services evaluation and experiments in performance with large scale datasets are discussed in more detail in Chapter 9.

7.5 Summing up

The goal of the TSS is to decide which of the mappings identified by PowerMap better instantiate the user query to create the ontology compliance triples. This is done by (a) determining those ontologies that cover entire triples and not just individual terms in the triples, and (b) carrying out a deeper analysis of the ontology relationships to determine the most likely semantic interpretation for a user query.

The similarity services cover many different scenarios in which the structure of the triples in a candidate ontology does not match the way the information was originally represented by the Linguistic Component (Query-Triples). The TSS solves the mismatches by, for example, increasing the number of Query-Triples when nominal compounds are split. Analogously, in each relevant ontology, the RSS can generate more than one Onto-Triple for each Query-Triple depending on the way the relevant data is organized in the ontology schema. Nevertheless, the similarity algorithm steps are designed to find the most precise and least time-consuming ontological translations first (the ones more similar to the Query-Triple), consecutively re-iterating and extending the search exploration if needed.

As a result, a linguistic triple can be mapped into one or more ontology triples, each one belonging to the same or different ontologies, and those may represent complete alternative translations of the linguistic triple, or partial translations that need to be combined by the merging and ranking component (Chapter 8).

Chapter 8 Merging and Ranking (step 4)

The component to merge and rank answers across ontologies in response to a user query is detailed in this chapter, which has been published at the Asian Semantic Web Conference in 2009 (Lopez et al., 2009b) where it won the best paper award.

8.1 Introduction

PowerAqua derives answers complying with the Onto-Triples obtained from multiple online semantic resources, operating in a highly heterogeneous search space. Hence, it requires mechanisms for merging and ranking answers to generate a commonly agreed set of answers across ontologies. Concretely, a side effect of the fact that PowerAqua explores multiple knowledge sources to obtain an answer is that the Triple Similarity Service (TSS) frequently associates the query to several Onto-Triples from different ontological sources, each Onto-Triple or set of Onto-Triples generating an answer. Depending on the complexity of the query, i.e., the number of Query-Triples (QTs) it has been translated to and the way each QT was matched to Onto-Triples, these individual answers may fall in one of the following categories:

- Valid but duplicated answers
- Part of a composite answer
- Alternative answers derived from different ontological interpretations of the QTs.

Different merging scenarios suit different categories: some cases require the intersection of the partial answers while other cases require their union. In Section 8.3 we exemplify these cases and explain the various merging scenarios and the fusion algorithm they rely on.

Additionally, whenever a set of answers is returned, it is important to be able to rate them in terms of their quality, correctness and completeness with respect to the user query. PowerAqua provides a novel semantic ranking mechanism, which lists the answers to a query in an order which

reflects PowerAqua’s own assessment of the reliability of the results. We explain these individual criteria and the way they are applied in Section 8.4 of this chapter.

8.2 Illustrative example

Consider the query “Which languages are spoken in South American countries?”, which can only be answered by aggregating partial answers across sources. Following PowerAqua’s cascade model, at the first stage the Linguistic Component transforms the NL query into the QTs: <languages?, spoken, South American countries>.

At the next step, the QTs are passed on to the PowerMap component, which identifies potentially suitable ontologies to answer a query, storing the initial element level mappings between the QT terms and the entities in these sources in the Entity Mapping Tables (EMTs). In addition, the semantic validation component attempts to generate WordNet senses for all classes and individuals included in the EMTs.

In the third step, the TSS takes as input the EMTs and the initial QTs and matches the QTs to ontological expressions, returning a Triple Mapping Table (TMTs) for each QT. The TSS chooses, whenever possible, the ontologies that better cover the user query and domain. In our example, as PowerAqua does not find any covering ontology with mappings for both arguments in the QT: “languages” and “South American countries”, the TSS algorithm reiterates again by splitting the compound term “South American countries”, and consequently modifying the QT into: <languages?, spoken, countries / South American>⁵¹ and creating a new QT for the compound <South American, ?, countries>. For the QTs obtained in this second iteration, the TSS extracts, by analysing the ontology relations, a small set of covering ontologies containing the valid Onto-Triples

⁵¹ Ambiguity is represented in the QT by “/”, i.e., more than one query term that can fulfil the same role.

(OTs) that jointly cover the query and produce an answer. The TMTs generated for each QT by the TSS are presented in Table 8.1.

Table 8.1 Triple Mapping Tables returned by PowerAqua for the example query

QT ₁ : <languages?, spoken, countries / South American>	
Dbpedia_infoboxes	OT ₁ <language, states, country> - 1945 answers: E.g.: Turkish (Albania), Juhuri (Azerbaijan), French (Algeria), Kurdish (Iran), Pashto (Pakistan), Welsh (United Kingdom), Tucano (Brazil), Catalan (Spain), etc.
RussiaB	OT ₂ <language, has_political_fact, country > - 1 answer: Russian (Russia)
QT ₂ : <South American, ?, countries>	
Dbpedia_infoboxes	OT ₃ <country, populationPlace, South_American_Brazilian> - 1 answer: Brazil
TAP	OT ₄ <country, locatedIn, SouthAmerica> - 17 answers: E.g.: CountryArgentina, CountryBolivia, CountryBrazil, CountryChile, etc.
Eculture	OT ₅ <country, hyponymOf, South American country> - 1 answer: synset-country-noun-1

Finally, because each resultant Onto-Triple only leads to partial answers (see Figure 8.1 and Figure 8.2), they need to be combined into one complete answer (see Figure 8.3). The goal of the fourth component of PowerAqua is to merge and rank the various interpretations that different ontologies may produce. In our example, this is achieved by intersecting the answers to the QT “languages spoken in a country” the ones obtained from “countries that are South American”. Among other things, merging requires to identify the same entities across ontologies, e.g., “Bolivia” and “CountryBolivia”.

In the above example, from a total of 1966 partial answers retrieved by PowerAqua the final set of answers obtained after merging contains 70 answers, including Yuracaré (Bolivia), Western Bolivian Guaraní (Bolivia), Pacahuara_language (Bolivia), Kariri (Brazil), Portuguese (Brazil), Chamacoco (Paraguay), among others. All of these answers, aggregated mainly from the DBpedia and TAP ontologies are correct, while partial incorrect answers derived from eculture and russiaB ontologies have been filtered during the merging processing. Nevertheless, ranking measures (as

will be explained in Section 8.4) can be applied to sort the answers, thus providing a ranking based on the confidence assigned by PowerAqua to the various ontological mappings. In this example all answers are ranked at the same level. We now describe these algorithms.

ASK ANOTHER QUESTION

Which languages are spoken in South American countries

Ask

Make use of WATSON ☐

LINGUISTIC TRIPLES
<subject, relation, object>

Query-Triples: < languages , spoken , countries South American > < South American , ? , countries > , Category: WH_GENERICTERM

Individual Answers
Merged Answers

Ontologies found with answers for < [languages], spoken, countries South American > [2]

1. mappings in kmi-web03.open.ac.uk:8890/http://dbpedia.org [1]

Turkish language, Macedonian language, Juhuri language, Tat language (Caucasus), Tsakhur language, ...

Mapping 1 (rank @ 1): < Language (language synonym) , states (states ontology_ad_hoc) , Country (country synonym) >

1945 answer(s)

Turkish language (Turkish language)

Macedonian language (Macedonian language)

Juhuri language (Juhuri language)

Tat language %28Caucasus%29 (Tat language (Caucasus))

Figure 8.1 Partial answers in DBpedia for languages in a country (illustrative example)

Ontologies found with answers for < [South American], null, countries > [3]

1. mappings in kmi-web03.open.ac.uk:8890/http://dbpedia.org [1]

Brazil, ...

2. mappings in <http://kmi-web07.open.ac.uk:8080/sesame/tapfull> [1]

"Argentina"@en, "Bolivia"@en, "Brazil"@en, "Chile"@en, "Clipperton Island"@en, ...

Mapping 1 (rank @ 1): < Country (Country synonym) , locatedIn (locatedIn ontology_ad_hoc) , SouthAmerica (South America equivalentMatching) >

17 answer(s)

CountryArgentina ("Argentina"@en)

CountryBolivia ("Bolivia"@en)

CountryBrazil ("Brazil"@en)

CountryChile ("Chile"@en)

CountryClipperton_Island ("Clipperton Island"@en)

View all Answers >>

Figure 8.2. Partial answers in TAP for countries in South America (illustrative example)

Individual Answers		Merged Answers
Sort by: Alphabet / Confidence / Popularity / WordNet Synset / Combined		
We found 70 answers in total from 4 ontologies		
Aikana language(Aikana language) dbpedia.org [Brazil]	⊕ Explain	score: 3
Akawaio language(Akawaio language) dbpedia.org [Guyana]	⊕ Explain	score: 3
Amanay%C3%A9 language(Amanayé language) dbpedia.org [Brazil]	⊕ Explain	score: 3
Anamb%C3%A9 language(Anambé language) dbpedia.org [Brazil]	⊕ Explain	score: 3
Apala%C3%AD language(Apalaí language) dbpedia.org [Brazil]	⊕ Explain	score: 3
Apiac%C3%A1 language(Apiacá language) dbpedia.org [Brazil]	⊕ Explain	score: 3

Figure 8.3. Merged answers for “Which languages are spoken in South American countries?”

8.3 Merging Algorithm

The merging of answers is performed either by their union or intersection, depending on how the terms are linked across Onto-Triples, as described in Section 8.3.1. A co-reference algorithm used during merging, defined for the purpose of identifying those answers from different ontologies that represent the same individual, is explained in Section 8.3.2.

8.3.1 Merging scenarios

Four scenarios may arise, in which merging is needed. These are described below:

Scenario 1 - The query is mapped to one Query-Triple. These are the simplest queries, and therefore the easiest ones to merge, as each Onto-Triple provides a set of final answers on its own. The complete set of answers is therefore the union of all answers within one or multiple ontologies. An example query “Find me cities in Virginia” is described in Section 8.4.1.

Scenario 2 - The query is mapped to two Query-Triples that share a common subject. Because each QT only leads to partial answers, they need to be merged together to generate a complete response. This is achieved by intersecting the answers from both triples. For instance, for the question “Which Russian rivers flow into the Azov sea?” (QT: <Russian rivers?, flow, Azov

sea>) the QT is split into two triples formed with the compounds that make up the first QT term (step S2 of TSS, as explained in Section 7.2). Then, the final answers are composed by intersecting the results obtained with “*rivers* in Russia” and “*rivers* that flow in the Azov sea”, i.e., <*rivers*, ?, Russia> and <*rivers*, flow, Azov Sea>”.

Scenario 3 – The query is mapped to two Query-Triples where the object of the first one is also the subject of the second one. Similarly to scenario 2, a complete answer can only be assigned by merging the partial answers. The answers for the first main QT are conditioned by the answers for the second QT. For example, in the query “Which rivers flow in European countries?”, the second QT term is split (step S3 of TSS in Section 7.2) and the final set of answers comprises the set of all countries in which rivers flow, <*rivers*?, flow, country>, and which are linked to the set of European countries <country, ?, European>.

Scenario 4 – Complex queries which are translated into multiple Query-Triples. These queries are solved as a combination of scenarios 1, 2 and 3. For instance, in “What are the main cities located in US states bordering Georgia?”, where Georgia is an ambiguous term that can represent a state in the USA or a country in the Caucasus, the valid answers come from both the *intersection* of “*cities* bordering Georgia (both as a state and as a country)” and “*cities* located in US states” (triples linked through the same subject), and the *intersection condition* of “*cities* located in US states” and “US states bordering Georgia” (triples linked through the same object-subject). Both paths give as a solution the cities in the state of Georgia (USA), rather than the ones in the country of Georgia. Similarly, consider a query with the same subject and object, such as “Which Russian rivers flow into Russian rivers”, PowerAqua would obtain the following QTs after splitting the compound Russian rivers: <river, flow, Russia>, <river, flow, river> and <river, ?, Russia>, if all QTs were mapped to valid Onto-Triples containing that information, the result would be the intersection between the answers obtained for the Onto-Triples sharing a common QT object (Scenario 2), that is <*rivers*, flow, Russia> , <*rivers*, ?, Russia> and <river, flow, river>, where the

objects of the last triple <river, flow, river> are conditioned to the answers of the triple with the same QT subject <river, ?, Russia> (Scenario 3).

In sum, the merging procedure deals with these four scenarios by applying three types of operators over the set of retrieved answers: ***union***, ***intersection*** (subject/subject) and ***intersection condition*** (object/subject). The union operator combines answers related to the same QT but coming from different ontologies (scenario 1). The intersection operator is needed when a query specifies more than one constraint for a single first query term (scenario 2). The intersection operator merges the answers from two corresponding QT and removes those, which only occur in one answer set. The *intersection condition* operator is similar to the intersection one; however, instead of intersecting the answers derived from both Onto-Triples (referring to the same subject), it filters the answers of the first Onto-Triple that hold a relation with the answers of the second Onto-Triple through the object (scenario 3). These operators can be applied to any complex cases in which the query is translated into several combinations of QTs or Onto-Triples (scenario 4), so that all answers produced by alternative paths are merged. For instance, in “Show me the capitals of the countries located in Asia?”, the answers produced by following different alternative mappings, e.g., <capital-city, has-capital-city, Asia > in *RussiaB* ontology and <capital, hasCapital, country> <country, part of, entity> <entity, located, Asia> in the *KIM* ontology, in the end must be merged by the union operator.

Merging is crucial to eliminate duplicates for queries in which a union operator is applied and to get valid answers for those queries in which an intersection or intersection condition operation is required. Examples from scenario 1-3 are shown in Figure 8.4.

Ultimately, to integrate the answers (in any merging scenario) it is necessary to know whether two instances from different ontologies represent the same individual, e.g., in the case of merging instances of countries whether “republic of Poland” in the *KIM* ontology is the same as “Poland” in the *mid-level* ontology. Such co-referencing is achieved with the algorithm described in the next section 8.3.2.

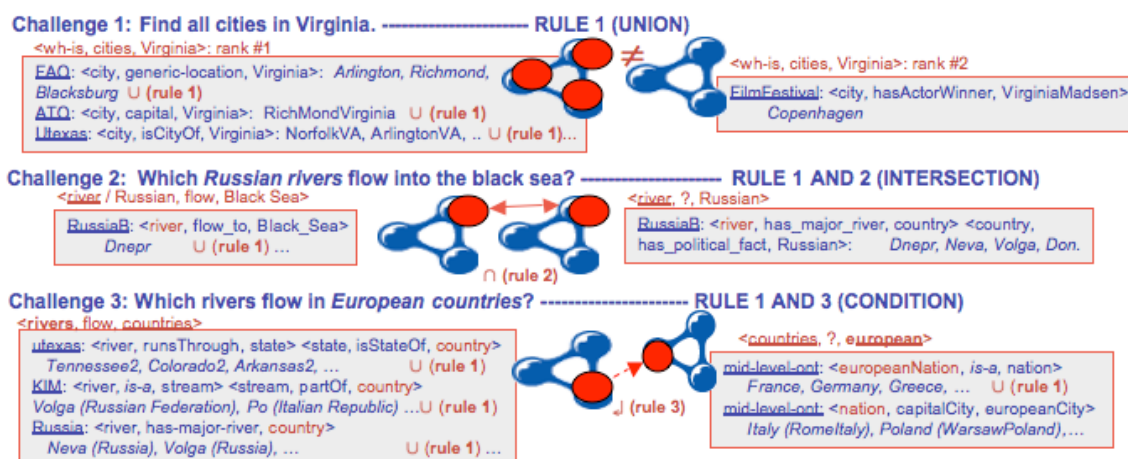


Figure 8.4. Merging examples and rules for scenarios 1, 2 and 3

8.3.2 The co-reference algorithm

The merging procedure assigns the individuals returned as answers from different ontologies into subsets of answers that represent identical entities. The union operation processes a set of answers from a single QT and merges the similar answers representing identical entities. For example, the QT: $\langle \text{countries?}, \text{locatedIn, Asia} \rangle$ returns, among its answers, “Thailand” from the TAP ontology and “Kingdom of Thailand” from the KIM ontology. These answers need to be grouped into a single subset as they refer to the same entity. As described above, depending on the query type, if there is more than one QT, these subsets of answers can afterwards be combined by intersection or intersection condition operations. Union and intersection operations look for identical pairs between subjects of ontological triples returned as answers, while intersection condition compares objects of triples from one set with subjects from another.

The atomic procedure performed by all of these operations is matching. Two answers are compared and a decision is made about whether or not they are identical. To increase the speed, initial matching is performed only between the labels and local names of the returned entities. Two kinds of similarity functions are involved: (1) string similarity functions (Jaro, Jaro-Winkler, edit distance); and (2) WordNet synonymy. The entities are considered identical if they are synonyms or if one of the string similarity functions returns a value above a certain threshold, obtained

experimentally. A special case is the processing of ambiguity, which occurs when an entity has two potentially identical matching entities that belong to the same ontology. For instance, in “Give me all cities in the USA,” a single entity, “arlington” from the *Fao-Agrovoc* ontology, has two potential matches, “arlingtonVa” and “arlingtonTx” from the *UTexas geographic* ontology. Assuming that individuals belonging to the same ontology are distinct, the system tries to choose the best match out of the two using additional context data from the ontologies. The system receives, for each entity, all of their property values from their respective ontologies and compares these sets using the same similarity functions as above on their elements. Thus, in our example, context sets for both entities “arlington” and “arlingtonVa” mention “Virginia”, while “arlingtonTx” mentions “Texas” instead. The similarity between the context sets of “arlington” and “arlingtonVa” is greater and, therefore, these entities are merged.

Pairwise comparison of all pairs of entities would make the complexity of the procedure N^2 with respect to the input set size. In order to avoid this, candidate matches are selected using a search over the indexes and the comparison involves only the entities that appear among the search results. This makes the complexity linear with respect to the answer set size.

An evaluation of the merging and co-reference algorithms is presented in Chapter 10.

8.4 Ranking algorithm and criteria

As we can see in Figure 8.5, a filtered set of answers for each query is obtained after the merging step. While an unsorted list of answers can be manageable in some cases, the search system may become unusable if the retrieval space is too big. In these cases a clear ranking criterion is needed to sort the final list of answers. The aim of the ranking measures presented here is to: a) assign a score to each individual answer and b) cluster the set of answers according to their score. Cluster analysis of ranking data attempts to identify typical groups of rank choices. In our case, according to the chosen ranking criteria, clusters identify results of different quality, popularity or meaning. The

cluster ranked at position one ($C@1$) represents the best subset of results according to the chosen ranking method.

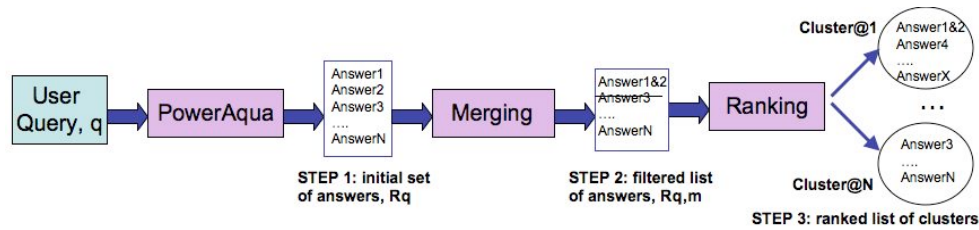


Figure 8.5: Flow of the data retrieval, merging and ranking process.

The ranking algorithm takes advantage of three different ranking mechanisms:

- Ranking by semantic similarity: this ranking criterion exploits a semantic similarity (the WordNet-based disambiguation algorithm described in Chapter 6.3) to compute the distance between Onto-Triples, and it leads to a partitioning of the answer set into clusters of answers with different meanings. For those cases in which alternative interpretations of the same query term are identified, answers obtained by means of the most popular meaning, i.e., the one appearing in a higher number of ontologies, are ranked first (Section 8.4.1).
- Ranking by confidence on the mapping: this ranking criterion is based on the confidence of the Onto-Triples from which the answer is extracted. The quality of the Onto-Triple depends on the fuzziness of the mapping at element level, i.e., if the mapping has been extracted using the original query term or by means of any of its synonyms or hypernyms, and on the quality of the mapping at triple level, i.e., how well the Onto-Triple leading to each answer covers the information specified in the Query-Triple (Section 8.4.2).
- Ranking by popularity: this ranking criterion is based on the popularity of the answer, defined as the number of ontologies from which this answer can be derived (Section 8.4.3). Popular answers are prioritised over non-popular ones.

Finally, the independent scores obtained for each of the three ranking algorithms can be combined to obtain a combined score using a weighting strategy (Section 8.4.4).

8.4.1 Ranking by semantic similarity

Answers are ranked according to the popularity of the semantic interpretation of the Onto-Triple they belong to. The hypothesis behind this is that if an answer is derived from an Onto-Triple that has similar interpretations (meaning) to other Onto-Triples from different ontologies, it is more likely to be correct than answers coming from unique, semantically different interpretations. This criterion takes advantage of both the knowledge inherent in the ontology and WordNet descriptions, and combines some well-founded ideas from the Word Sense Disambiguation community to compute a semantic similarity distance across ontological entities, as seen in Chapter 6.

Let's clarify this idea with the example "Give me cities in Virginia", a query matched by PowerAqua into eight ontologies (and eight Onto-Triples). The final set of answers obtained after merging should be the union of all the answers describing cities in Virginia. However, an instance labelled "Copenhagen" appears among the set of merged answers. In order to rank this inaccurate answer last, semantic similarity between Onto-Triples is computed by comparing the distance path and common ancestors between the WordNet senses for each ontological concept representing the subject and object of the triple (predicates are not well covered in WordNet, and in the case of instances we look at their types). The WordNet sense of an ontological term A, is determined by its parents in the hierarchy of the ontology (that is, those senses of A that are similar to at least one sense of its ancestors in the ontology), and by its intended meaning in the user query (those senses of A that are similar to at least one sense of the user term it matches to, if their labels differ). Having said that, while "city" has similar meanings in all its eight ontological matches (the senses are semantically similar, even if they are not exactly the same), the ontological meaning of "Virginia" differs. Indeed, seven of the ontologies are referring to Virginia as an instance of a *state* or *province* (in USA), while the answer "Copenhagen" is derived from an eighth ontology about *film festivals*, using the Onto-Triple, namely <city, hasActorWinner, VirginiaMadsen>, where "VirginiaMadsen" is classified as *person* in the ontology and not as a *state*, and therefore the intended meaning of the Onto-Triple differs from the previous ones.

Ranking among answers is then calculated according to the popularity criterion, in terms of the number of ontologies which support the same interpretation. Therefore, the first complete set of ranked answers comes from the union of the answers from the seven semantically similar Onto-Triples referring to cities in the *state* or *province* of Virginia. The answer labelled “Copenhagen”, because it is derived from the only semantically different Onto-Triple, would be ranked lower than the previous answers.

To conclude, the score for each answer is the number of ontologies that share the semantic interpretation of the Onto-Triple they belong to, or -1 if an answer is coming from two Onto-Triples with different semantic interpretation. The C@1 groups all the answers ranked with score 2 or highest (at least two ontologies with the same semantic interpretation).

8.4.2 Ranking by confidence

The quality of the matching between a QT and one (or, in some cases, two) Onto-Triples often has an influence on the quality of the derived answers. We identified a set of rules to predict which of these Onto-Triples are likely to be more reliable and thus potentially lead to a better set of answers. These rules are listed in the same order as they are applied, i.e., from the most to the least significant (from 1 to 6). These rules can be seen as nodes in a decision tree. Their order of preference is discriminative in order to avoid conflicts. The ranking of Onto-Triples is based on assigning scores sequentially. The answers that rank higher using the first rule will have a better score (score 1) than the ones using the subsequent rule (and so on), therefore being more reliable.

1. Onto-Triples that are based on only equivalent (i.e., exact and approximate match) or synonym type mappings to the corresponding QT terms are ranked higher than the ones based on lexically (hypernym / hyponym) related matches. For instance, for the QT <capitals?, ?, USA> (“Find me capitals in the USA”) the Onto-Triple1= <capital (exact), isCityOf, State> <State, isStateOf, USA (exact)> with only equivalent mappings is ranked

higher than the Onto-Triple2= <City (hypernym), attribute_country (domain relation), USA (exact)> which contains one hypernym.

2. Onto-Triples that link the two arguments in the QT through an IS-A relation are ranked in a lower position than the triples based on a domain relationship. The reason for this is that many online ontologies misuse IS-A relations to model other types of relations, such as partonomy (Sabou et al., 2008). We do not apply this rule when the original question contains an IS-A relation, as this is an indication that such a relation is expected (“Which animals ARE reptiles?”). For instance, the QT <person/organization?, plays, Nirvana> is matched to Onto-Triple1 = <person, hasMember, MusicianNirvana> and Onto-Triple2 = <Nirvana Meratnia, IS-A, person>. Note that while rule 1 ranks these two triples equally, this rule ranks Onto-Triple1 higher (even if, in this particular case, Onto-Triple2 complies with correct modelling).
3. Onto-Triples that cover not only all of the terms in the respective QT, but also the linguistic relation (mapped as an ontological entity), are ranked first over triples that do not cover the relation. For instance, for the QT <states?, bordering, Colorado> (“What are the states bordering Colorado?”), the Onto-Triple that also maps the relation “bordering” <state, borders, Colorado (state)> is ranked higher than <state, runsThrough, Colorado (river)>.
4. The Onto-Triples containing more exact mappings are ranked higher over Onto-Triples based on less exact mappings. For example, for the QT <London?, capital, country> (“Is London the capital of any country?”), the Onto-Triple <London (exact), hasCapitalCity, Country(exact)> is preferred over <capital_city (WNSynonym), has_capital_city, country (exact)>. Note that this rule is similar to rule 1, but because it is applied at a later stage, it is more restrictive.
5. For “who queries”, Onto-Triples formed with “person” are ranked higher than the ones formed with “organization”.

6. Onto-Triples based on direct mappings (1:1 mapping between a QT and an Onto-Triple) are ranked higher to those relying on indirect mappings (1:2 mappings). For example, “Who works at the open university?” is translated to both Onto-Triple1= <person, memberOf, openUniversity> and to Onto-Triple2 = <person, mentions-person, kmi-planet-news (subclassOf publication)>, <kmi-planet-news, mentions-organization, the-open-university>. Onto-Triple1 is ranked higher than Onto-Triple2 as it contains a single mapping.

As said before, these criteria can be seen as nodes in a decision tree, where the order of preference is discriminative, i.e. Onto-Triples that comply with criteria 1 will always be ranked the highest, and among them we can apply the other criteria, in order. For example, in “Who believe in the apocalypse?” the Onto-Triple <religious_organizations, announce, apocalypse> formed with all equivalent mappings (criteria 1) is ranked higher than <musicArtist (hypernym), maker, revelation (synonym)>, even if the latter is preferable to the former according to criterion 6.

Once the score is assigned to each answer the clusters are created as follow: C@1 contains all the answers ranked highest (score 1), C@2 contains all the answers ranked in position 2, and so on.

8.4.3 Ranking by popularity

This criteria ranks answers according to their popularity, i.e., the number of individual ontologies from which they are derived. For instance, “Where is Paris?” produces two answers: France (or French Republic) and United States (as Paris is a city in the state of Texas). In this case, France is the most popular answer across ontologies and therefore is ranked first. An answer is C@1 if its popularity is higher than 1 (more than 1 ontology).

8.4.4 Ranking by combination

Finally, we propose a last strategy to improve ranking, by the combined use of all the ranking methods presented before. We argue that, due to the different nature of these approaches, relevant answers not selected and irrelevant answers not filtered by one specific method, are suitable to be selected or filtered by the others. For the combination strategy we have used the weighted Borda

method (Bartell, 1994), in which votes are weighted taking into account the quality of the source. The combined weight for the answer i within the context of the query q , $W_{i,q}$, is therefore computed as:

$$W_{i,q} = 2 * x \text{ (} x=1, i \in \text{confidence } C@1 \text{)} + 1 * x \text{ (} x=1, i \in \text{confidence } C@2 \text{)} + 1 * x \text{ (} x=1, i \in \text{semantic similarity } C@1 \text{)} + 1 * x \text{ (} x=1, i \in \text{popularity } C@1 \text{)}.$$

We have empirically tested, in the merging and ranking evaluation presented in Chapter 10, that the most important ranking algorithm is confidence. With the proposed combination we attempt, on the one hand, to provide this measure for a significant number of answers (selecting $C@1$ and $C@2$) and, on the other hand, to provide a higher score to those answers with a higher confidence value. Once the scores are computed each answer is then clustered according to: a) its final score value and b) the selected degree of relevance for precision and recall measures in the final answer. To maximize recall, $C@1$ is generated with all the answers for which $W_{i,q} > 1$. To maximize precision, $C@1$ is generated with all the answers for which $W_{i,q} > 2$.

8.5 Initial Research work on using Trust for Ranking

We have identified a set of ranking criteria to predict which of the ontological mappings are likely to be more “robust” and thus potentially lead to a good answer. Our criteria are twofold. On the one hand they provide the system with an initial “self-awareness” capability, which allows it to estimate its own confidence in the quality of the answers it is able to retrieve. On the other hand, it uses the heuristic that if multiple ontologies with different conceptualizations point to the same answer, they are more likely to be correct than unique answers found in “isolated” ontologies.

However, answers replicated across many ontologies can bias the answers generated from specialist knowledge ontologies. PowerAqua does not make any assumption a priori about the ontologies it uses. However, the different SW datasets differ in the level to which we can trust them. For this reasons, future research can be done in the direction of integrating PowerAqua with a trust

engine for ontologies, as an additional mechanism to rank answers to queries. We also believe that integrating a trust engine would be a significant step, allowing users to make judgments about semantic resources in the context of their use, a key requirement for the uptake of this type of technology.

One potential trust engine, described in (Lopez et al., 2009a), takes as input feedback from an user about the quality of the information received from a particular source, and formally computes the current trust values for all entities associated in the ontology. The trust values are assigned to statements or ontological triples and are propagated to the individuals ontologically related, directly or indirectly, with each triple. The rules used to update trust values for ontological entities, as a result of an assignment of a quality rating to a triple by an agent, consider all the key primitives in RDFS: domain, range, subClassOf, type, and subpropertyOf.

An initial integration of PowerAqua and the trust engine was realized by providing a simple interface to the user, which allowed users of PowerAqua to provide feedback in the traditional fashion used by review sites, by rating each answer with a number of stars (from 1, not satisfied, to 5, very satisfied), thus allowing to build (over time) sophisticated models of the quality of information available on different SW sources. The trust value returned by the trust engine can be used to rank the different ontological answers returned by PowerAqua. For instance, for a query like “What is the capital of Spain?” the user can evaluate the triple <Madrid, capital-of, Spain> as very good in ontology “A” (5 stars), and the triple <Seville, capital-of, Spain> as invalid (1 star).

Nevertheless, some issues remain open:

- It is important to note that, in contrast with the ranking mechanism presented in Section 8.4, the integration of PowerAqua and the trust engine only allows a user to make a quality judgment about the intrinsic correctness of the statement returned by PowerAqua. In particular, in some cases PowerAqua fails to correctly interpret the user’s question and as a result wrong answers are returned, not because there is anything wrong about the ontology,

but simply because of the mapping error made by PowerAqua. In these cases it would be incorrect for users to give low quality rating to the statements in the question

- In order to use the trust engine in PowerAqua, a user needs to be logged in through the PowerAqua login mechanism. The current implementation of the trust model only supports individual user views over an ontology. In other words both assigning quality values and retrieving trust values are user-specific mechanisms. In the future, it will be useful to extend it to support also community-oriented views.
- The propagation rules implemented by the trust engine do not scale to a run time scenario when very large ontologies are evaluated.

8.6 Summing up

In this chapter we have presented a novel merging approach, which automatically combines and reconciles answers from different sources, filtering-out irrelevant information. For the instance fusion task, the SW community has adopted similarity approaches from the database community, in which the distance between database records is calculated as a weighted average of distances over their attributes. Basic similarity metrics based on string comparison have been developed in the database community, e.g.: (Bilenko and Mooney, 2003), (Winkler, 1999). These metrics are used as a basis for the majority of algorithms, which compare values of attributes of different data instances and aggregate them to make a decision about two instances referring to the same (usually real-world) entity - see (Elmagarmid et al., 2007) for a survey. More specifically, the problem of merging, or finding identical individuals, has been mostly considered in the context of offline data fusion scenarios. In contrast with these approaches, we have developed an approach for combining answers obtained from multiple distributed heterogeneous data sources on the fly, which aggregates composite answers, filter irrelevant answers, and fuse similar answers together.

This component also applies a range of ranking measures to rank answers derived from different knowledge sources in the context of a user query, which take into account (in this order):

- Criteria derived by the quality of the mapping between QTs and their corresponding Onto-Triples (how confident is PowerAqua about the mappings used to derive the answer?).
- Criteria about the popularity of the interpretation of the answer (are the answers from different Onto-Triples representing a similar interpretation of the query in order to obtain an unambiguous and unique set of answers?).
- Criteria derived after fusion referring to the frequency / popularity of the answer across ontologies (*which answer is returned by the most ontologies?*).

An evaluation of the efficiency of the merging and fusion algorithm and these ranking techniques, based on both confidence and popularity criteria, is presented in Chapter 10.

PART III: EVALUATION, SCALABILITY, CONCLUSIONS AND OUTLOOK

In Part 3 a number of evaluations are presented, which assess the ability of the proposed approach to support users in querying and exploring the Semantic Web and Linked Data. This thesis then concludes with a summary of the main contributions and an outline of the main directions for future research.

"When we think we know all the answers, life comes and changes all the questions" (anonymous)

Chapter 9 Evaluation of PowerAqua with respect to Question Answering on the SW

The first evaluation of PowerAqua presented in this chapter has been published in the Knowledge Capture Conference in 2009 (Lopez et al., 2009b).

The second evaluation presented in this chapter (on merging and ranking) has been published and awarded the prize for best paper in the Asian Semantic Web Conference in 2009 (Lopez et al., 2009b).

The third evaluation was carried out following the formal benchmark proposed for the SEALS 2010 semantic search evaluation campaign. SEALS is an EU funded project (FP7 238975) and information on the evaluation campaign on semantic search tools and the 2010 results are available at: <http://www.seals-project.eu/seals-evaluation-campaigns/semantic-search-tools>.

9.1 Introduction

In contrast with the Information Retrieval (IR) community, where evaluation using standardized techniques, such as those used for the annual TREC competitions, has been common for decades, the Semantic Web (SW) community is still a long way from defining standard evaluation benchmarks for semantic search tools. Important efforts have been made in the last few years towards the establishment of common datasets, methodologies and metrics to evaluate semantic technologies, e.g., the SEALS project (see Section 9.4). However, the diversity of semantic technologies and the lack of uniformity in the construction and exploitation of the data sources are some of the reasons why there is still not a general adoption of evaluation methods. Evaluating PowerAqua constitutes a major research challenge, not just because of this lack of standard evaluation datasets, but because of the inherent difficulties in defining evaluation methodologies able to assess the quality of its cross-ontology search capabilities. Therefore, in order to provide an in-depth view of PowerAqua's capabilities and current shortcomings, in this chapter we present three sets of evaluations focused on different capabilities of PowerAqua:

- Evaluation of PowerAqua's ability to interpret and derive answers to Natural Language (NL) queries from multiple ontologies selected on the fly, during the QA process. Overall

accuracy (the percentage of questions that are answered correctly from the total) and time performance (the time the system requires to answer a query) were measured for this evaluation. Specific PowerAqua limitations and the component that lead to the failure are analysed in detail.

- Evaluation of PowerAqua's merging and ranking of answers across ontologies (Section 9.3). The merging and ranking component was under development when the experiments for the first evaluation were conducted, thus this second evaluation assesses the quality of the results obtained after the application of the merging and ranking module. A complex enough information space and a set of representative user queries that could be correctly mapped by PowerAqua into several ontological facts, preferably across different ontologies, were selected. Thus, in this scenario, merging or ranking techniques were required in order to obtain accurate and complete answers. Precision and recall were selected as evaluation metrics, where precision is the number of correct answers from the total of retrieved answers; and recall is the number of correct answers, after applying the merging and ranking, with respect to the number of correct answers before merging and ranking algorithms are applied.
- Usability evaluation (Section 9.4). While the previous evaluations aimed to test PowerAqua's competence to answer queries from distributed information sources, the third evaluation presents the first usability results of PowerAqua as a NL interface to semantic repositories. This evaluation was carried out following the formal benchmark proposed for the SEALS 2010 semantic search evaluation campaign, focused on the interface and usability aspects of different search tools within a user-based study in a controlled scenario.

The aim of the evaluations presented here is to probe the feasibility of performing QA in an open SW scenario, defined by multiple heterogeneous semantic sources. However, the wealth of semantic data accessible by PowerAqua could also be used for more ambitious experimental purposes on the

Web. Thus, in Chapter 10 we investigate the added value of PowerAqua as the NL interface of a complex IR system, where it plays the role of an advanced, semantic query expansion module. The answers obtained by PowerAqua for the user NL queries are evaluated in terms of their ability in supporting document retrieval by using query expansion. The practical advantage of this evaluation with respect to the ones presented in this chapter is that for the first time the evaluation could be conducted using input semantic data that is heterogeneous, while the queries and success criteria are externally sourced and independently built in the context of an international benchmark (using standard IR TREC precision and recall evaluation metrics).

Finally, the appearance of Linked Data has defined a turning point in the evolution of the SW and its applications, moving towards the exploitation of real-world, massive, heterogeneous and distributed semantic information. Note that the set of experiments presented in this Chapter to measure PowerAqua's real time performance did not include the latest large Linked Data datasets. Thus, in Chapter 11 the feasibility of PowerAqua to scale to this new massive semantic information space, as well as the implications and challenges on PowerAqua's algorithms, are investigated, by performing experiments on one of the largest and most heterogeneous LOD datasets, DBpedia.

9.2 Evaluation 1: Deriving answers from multiple ontologies

This first evaluation of PowerAqua as a standalone system is targeted to assess its capability to derive answers to NL queries by relying on the information provided by multiple ontologies on the fly. As such, the evaluation described here primarily assesses the mapping capabilities of the system (i.e., its ability to map a user query onto ontological triples in reasonable time), rather than its current linguistic coverage, which we are constantly improving. Nevertheless, this evaluation also gives us an insight into the extent to which PowerAqua satisfied user expectations about the range of questions the system should be able to answer. The experimental setup is explained in Section 9.2.1, the list of queries and results are presented in Section 9.2.2, and the analysis of the results is given in Section 9.2.3.

9.2.1 Experimental Setup

In this section we describe the evaluation criteria as well as the data sets (ontologies and a corpus of questions) used for our experiments.

Our goal is to build a system that provides correct answers to a query, in a reasonable amount of time, by making use of at least one ontology that provides the required information. In order for an answer to be correct, PowerAqua has to align the vocabularies of both the query and the answering ontologies. PowerAqua fails to give an answer if the knowledge is in the ontologies but it can not find it. Total recall cannot be measured in this open scenario, as we don't know in advance how many ontologies can potentially answer the user's query. However because we ensured that there was at least one ontology covering each query, there are no conceptual failures (i.e., the case in which the knowledge is not actually in any of the ontologies) in this evaluation. Therefore, the major evaluation criteria are *success*, in terms of getting the correct answer, and *speed*.

We tested our prototype on a collection of ontologies saved into online repositories and indexed by PowerMap. The collection includes high level ontologies, like *ATO*, *TAP*, *SUMO* and *DOLCE*, in which the biggest source, *SWETO*, contains over 3 million triples, with around 800,000 entities and 1,600,000 relations (1 GB of data). In total, we collected around 3GBs of data stored in 130 Sesame repositories (each repository containing one or more semantic sources, in total more than 700 ontological documents), which are accessible through <http://poweraqua.open.ac.uk:8080/sesame>. For this evaluation, we preferred to use this big static data set rather than directly fetching ontologies from the Watson semantic search engine to ensure that: 1) the experiments are easily reproducible, 2) the question designers can easily browse the semantic sources in the collection to generate queries, and 3) the size and quality (in terms of richness) of the ontologies is higher than in those found by Watson, which includes a very large number of small, lightweight ontologies (often not populated) and foaf files. An evaluation of PowerAqua as an interface to Watson is presented in 11.6.2 as part of a study on scalability.

The questions used during the evaluation were selected as follows. We asked seven members of KMi, familiar with the SW, to generate factual questions for the system that were covered by at least one ontology in our collection. Factual queries are formed with wh-terms (which, what, who, when, where), or commands (give, list, show, tell, etc.), that vary in length and complexity: from simple queries with adjunct structures or modifiers to complex queries with relative sentences and conjunctions or disjunctions. We pointed out to our colleagues that the system is limited in handling questions that required temporal reasoning (e.g., today, last month, between 2004 and 2005, before 2008, etc) and compositional semantic constructions (quantification, comparison, negation and number restrictions). As no ‘quality control’ was carried out on the questions, it was admissible for them to contain grammatical errors. Also, we pointed out that PowerAqua is not a conversational system, information about each query is resolved on its own with no reference to previous queries, and it cannot prompt users for extra information. We did not provide them with any additional details about the kind of queries PowerAqua is able to understand, so that we can also assess to which degree PowerAqua satisfies the expectations of the users about the range of questions they can ask. We collected a total of 69 questions presented in the next Section.

9.2.2 List of queries and results

The 69 questions collected and the results are listed as follows in Table 9.1:

Table 9.1. List of queries used in PowerAqua’s first evaluation

Q1: Give me all papers authored by Enrico Motta in 2007 SUCCESS: 23.305 secs.
Q2: What is the title of the paper authored by Enrico Motta in 2007? Linguistic Component failure: out of coverage.
Q3: Give me all albums of Metallica SUCCESS: 1.817 secs.
Q4: Which are the albums of the rock group Metallica? SUCCESS: 18.428 secs.
Q5: Give me all Californian dry wines PowerMap failure (ontology discovery): it fails to map the term "Californian" to the concept "CaliforniaRegion", therefore it only gives dry wines as an answer.
Q6: Which wines are dry? SUCCESS: 6.123 secs.
Q7: Which religion does Easter belong to?

SUCCESS: 5.337 secs.

Q8: Which are the main fasting periods in Islam?

SUCCESS: 12 secs.

Q9: Is there a railway connection between Amsterdam and Berlin?

Linguistic Component failure: out of coverage.

Q10: What are the main rivers of Russia?

SUCCESS: 6.108 secs.

Q11: Which Russian rivers flow into the Black Sea?

SUCCESS: 8.867 secs.

Q12: Which prizes have been won by Laura Linney?

SUCCESS: 9.302 secs.

Q13: What are the symptoms of Parkinson?

SUCCESS: 4.916 secs.

Q14: Who stars in Bruce Almighty?

SUCCESS: 12.89 secs.

Q15: Who presented a poster at ESWC 2006?

Triple Similarity Services failure: the TSS fails to understand the term "ESWC 2006" because, although the relevant ontology is about the 2006 European Semantic Web Conference, it does not have an ontological element as such to represent the event "ESWC 2006" as it assumes all the ontology is about that.

Q16: What is the capital of Turkey?

SUCCESS: 14.19 secs (ranking is not accurate because the main ranked ontological triple gives as an answer the list of all local capitals. This is not because of the ranking component but because PowerMap identifies both an exact mapping, "Capital", and an approximate one, "CountryCapital" within that ontology. It then proceeds to rule out the approximate one, which is actually the one that would have led to the correct answer, the exact mapping "Capital" generates regional capital cities instead).

Q17: Who are the people working in the same place as Paolo Buquet?

Linguistic Component failure: out of coverage.

Q18: Give me all the articles written by people from KMi.

PowerMap failure (mapping discovery): The term "people" is not mapped to "person" in the relevant ontology (they are not synonyms in WordNet). Moreover the term "KMi" does not appear as an alternative name for "knowledge media institute" in the ontologies, and therefore it cannot be found. Note that the use of acronyms to find mappings could have solved this.

Q19: Give me the list of restaurants which provide Italian food in San Francisco

SUCCESS: 28.414 secs.

Q20: Which restaurants are located in San Pablo Ave?

SUCCESS: 22.287 secs.

Q21: Which cities are located in the region of Sacramento Area?

SUCCESS: 13.461 secs.

Q22: What is the apex lab?

SUCCESS: 0.501 secs.

Q23: Who believe in the apocalypse?

SUCCESS: 12.016 secs.

Q24: What is skoda doing?

Linguistic Component failure: Query not correctly classified, it can be reformulated to "What is skoda?".

Q25: Where are Sauternes produced?

PowerMap failure (filtering heuristics): The system correctly maps the linguistic term "Sauternes" to the wine "Sauterne". However this is not related to a region, leading to a failure at the next stage of the process. PowerMap had indeed identified the mapping which would have led to the answer ("SauterneRegion", which is located in "Bordeaux"), however this mapping was discarded because PowerMap considered it less likely to be correct than the exact mapping to "Sauterne".

Q26: Give me the papers written by Marta Sabou

SUCCESS: 3.822 secs (However, it can not find all relevant ontologies because the term "papers" is not mapped to the entity "publication").

Q27: which organization employs Enrico Motta?

SUCCESS: 10.785 secs.

Q28: where is Odessa?

SUCCESS: 58.301 secs.

Q29: which russian cities are close to the Black Sea?

SUCCESS: 15.533 secs.

Q30: give me oil industries in Russia

SUCCESS: 4.066 secs.

Q31: Which sea is close to Volgograd?

SUCCESS: 8.782 secs.

Q32: Name the president of Russia

SUCCESS: 3.491 secs.

Q33: which countries are members of the EU?

SUCCESS: 27.855 secs.

Q34: What is the religion in Russia?

SUCCESS: 3.816 secs.

Q35: Which sea do the Russian rivers flow to?

SUCCESS: 79.197 secs.

Q36: what is activated by the Lacasse enzyme?

SUCCESS: 12.896 secs.

Q37: What enzymes are activated by adrenaline?

SUCCESS: 5.212 secs.

Q38: what enzymes are used in wine making?

SUCCESS: 7.391 secs.

Q39: give me the actors in "the break up"

SUCCESS: 6.013 secs.

Q40: Give me fishes in salt water

SUCCESS: 7.606 secs.

Q41: Give me the main companies in India

SUCCESS: 23.646 secs.

Q42: What is the birthplace of Albert Einstein

PowerMap failure (ontology discovery): fails to map "birthplace", and it splits the compound "Albert Einstein".

Q43: Show me museums in Chicago

SUCCESS: 5.075 secs.

Q44: Give me types of birds

SUCCESS: 0.647 secs.

Q45: In which country is Mizoguchi?

Triple Similarity Service failure: it can not infer from Riichiro Mizoguchi who is affiliated to the Osaka University from Ontology 1 to the fact that Osaka is in Japan from the Ontology 2. PowerAqua can not do a "double cross-ontology jump" to link two linguistic terms within the same linguistic triple.

Q46: Find all rivers in Asia

SUCCESS: 15.501 secs (partial answer because, for efficiency reasons, as there is a direct relation between rivers and Asia in the ontology it does not look for the indirect relation river-country-Asia which would have produce more answers within the same ontology).

Q47: Find all Asian cities

PowerMap failure (ontology discovery): It fails to map "Asia" to "Asian", the query can be reformulated to "find me cities in Asia".

Q48: Which universities are in Pennsylvania?

SUCCESS: 10.52 secs.

Q49: Who won the best actress award in the toronto film festival?

SUCCESS: 57.246 secs.

Q50: Which desserts contain fruits and pastry?

PowerMap failure (ontology discovery): it fails to map "dessert" to "dessertDishes" and "fruit" to "fruitDishes".

Q51: Are there any rivers in Russia that flow to Caspian and Black Sea?

Linguistic Component failure: out of coverage.

Q52: What ways are there for weight management?

SUCCESS: 6.311 secs.

Q53: What kinds of brain tumour are there?

PowerMap failure (ontology discovery): It can't find the literal "Brain Tumor SY NCI" which is a synonym of the class "Brain Neoplasm".

Q54: Where is Lake Erie?

SUCCESS: 14.448 secs.

Q55: Which islands belong to Spain?

SUCCESS: 40.61 secs.

Q56: List some Spanish islands

PowerMap failure (ontology discovery): It cannot map "Spanish" to "Spain". It can be reformulated to "list me some islands in Spain". This error have been solved in the latest version of PowerAqua by using WordNet derived words

Q57: Which islands are situated in the Mediterranean sea?

SUCCESS: 12.44 secs.

Q58: Which Spanish islands lie in the Mediterranean sea?

SUCCESS: 8.12 secs.

Q59: How many airports exist in Canada?

SUCCESS: 17.839 secs.

Q60: Which terrorist organization performed attacks in London?

PowerMap failure (filtering heuristics): The ontology is not well modeled (redundant terms not connected between themselves). The literal "London, United Kingdom" is discarded by the exact instance mapping "London", which is not related neither to the instance "United Kingdom" nor to the literal "London, United Kingdom". Those two last ontological terms being the only ones that link to the class "terrorist organization".

Q61: Which are the main attacks that took place in the United Kingdom?

SUCCESS: 35.729 secs.

Q62: What are the terrorist organizations that are active in Spain?

SUCCESS: 12.077 secs.

Q63: What type of buildings exist?

PowerMap failure (ontology discovery): It does not find the term "building" in the sweto ontology.

Q64: Which RBA banks are situated in Switzerland?

SUCCESS: 24.252 secs.

Q65: What are the bank types that exist in Switzerland?

SUCCESS: 5.789 secs.

Q66: How many tutorials were given at iswc-aswc2007?

PowerMap failure (ontology discovery): it cannot find "tutorialEvent" as a mapping for "tutorials". The query can be re-formulated to "how many tutorial events were given at iswc-aswc 2007?" (in this case it only maps the term "2007" which is the localname of the entity "/iswc-aswc/2007" with no label).

Q67: How can you treat acne?

PowerMap failure (ontology filtering): it finds the class "acne" which is not connected to any other entity, while it discards the approximate mapping "acneTreatment" that would have lead to the answer.

Q68: What drugs can be used for reducing fever?

Triple Similarity Service failure: 12.016 secs, it tries to find mappings for the terms “drugs”, “reducing” and “fever” while the answer is contained in a unique class, namely “FeverReducingDrug” (ontology modeled with different levels of granularity).

Q69: Which drug can treat colds and reduce headache?

PowerMap failure: it can not map the term “drug” to the class “drugMedicine” and the term “colds” to the class “coldCoughCareProduct” to obtain for example (ibuprofen, is-a, coldCoughCareProduct) (ibuprofen, is-a, drugMedicine). Nevertheless it maps “cold” to “CommonCold”, however the class “CommonCold” is not connected to “drugMedicine” or to “coldCoughCareProduct”.

9.2.3 Analysis of Results

PowerAqua successfully answered 48 (thus 69.5%) out of the 69 questions collected. This is actually a rather good result, given i) the open nature of the question answering set up –hardly any constraints were imposed on the choice of the questions, and ii) the size and heterogeneity of the dataset. We analysed the failures and divided them into the following categories according to the component that led to the error (see Table 9.2):

Linguistic analysis: a failure can be due to the query being out of the scope of the linguistic coverage (4 failures), or an incorrect annotation on the underlying GATE linguistic platform and grammars (e.g., annotating a verb as a noun) that leads to a misunderstanding of the query (1 failure). In total 5 of the queries, 7.2%, failed because of incorrect linguistic analysis. Extending the coverage of the linguistic grammars, which currently only focus on factual queries, to queries that require a meaningful dependency structure of the sentence elements might solve such errors (e.g., Q17: “Who are the people working in the same place as Paolo Bouquet?”).

PowerMap: this component tries to maximize recall to broaden the search space. Accuracy is not so crucial at this stage, as incorrect mappings will probably be discarded at a later stage by using the semantics of the ontologies. However, too many irrelevant mappings collected in this phase inevitably affect the overall performance of the system, therefore filtering heuristics are applied to achieve a compromise between performance and recall. In our evaluation 13 queries, 18.8% of the total failed either because of relevant mappings that could not be found (10 of them), or because they were indeed found but later discarded by the filtering heuristics, as they were considered less likely to lead to the correct solution than other mappings (3 of them). PowerMap needs semantic

sources with enough human understandable labels to obtain high performance. In this evaluation there were no failures due to the WordNet-based semantic component used to discard semantically invalid mappings. An alternative evaluation of this module to assess semantically sound mappings is detailed in Section 6.5.

Triple Similarity Service: a TSS-related failure occurs when PowerMap correctly selects an ontology (and all mappings) containing the answer to the query, but the TSS fails to complete the matching process by locating the correct triple(s) answering the query. This can be due to several reasons, such as incorrectly linking the terms into triples, not looking for paths longer than 2 between two entities, selecting wrong mappings to create a triple, or because of low quality or incomplete ontologies. In our evaluation, 3 of the queries, 4.3%, failed due to this component.

Merging component: most queries did not require merging across ontologies in this evaluation. An appropriate evaluation in which the ranking and merging component is thoroughly tested is presented in Section 9.3.

Table 9.2. Analysis of the evaluation results

Successful queries			Total
Q1, Q3, Q4, Q6, Q7, Q8, Q10, Q11, Q12, Q13, Q14, Q16, Q19, Q20, Q21, Q22, Q23, Q26, Q27, Q28, Q29, Q30, Q31, Q32, Q33, Q34, Q35, Q36, Q37, Q38, Q39, Q40, Q41, Q43, Q44, Q46, Q48, Q49, Q52, Q54, Q55, Q57, Q58, Q59, Q61, Q62, Q64, Q65			48 (69.5%)
Failures			Total
Linguistic Failure	Out of coverage	Wrong classification/ annotation	5 (7.2%)
	Q2, Q9, Q17, Q51	Q24	
PowerMap Failure	Fail to syntactically find the mappings	Filtering out of valid mappings	13 (18.8%)
	Q5, Q18, Q42, Q47, Q50, Q53, Q56, Q63, Q66, Q69	Q25, Q60, Q67	
TSS Failure	Q15, Q45, Q68		3 (4.3%)

As far as speed is concerned, the average answering time for the successful queries was 15.39 seconds⁵²; with queries ranging from 0.5 to 79.2 secs, the best time was 0.5 secs. The worst case was for Q35: “Which sea do the Russian rivers flow to?” (see Figure 9.1), linguistically translated into the Query-Triple: $\langle \text{sea?}, \text{flow}, \text{Russian rivers} \rangle$, where the keyword “sea” produces more than 300 mappings (most of which are later discarded). “Flow” produces also a few mappings in different ontologies, and “Russian rivers” produces two mappings: “ID_russianriverPub3493”, an instance of a restaurant in Forestville from an ontology about restaurants, and the literal “russian river tule perch”, which corresponds to a species name from the *Embiotocidae* family in the *FAO* ontology. The *FAO* ontology is selected by the TSS as the only covering ontology because it also has 40 approximate mappings for “sea” (“sea-bass”, “sea-catfish”, “sea-cucumber”, “sea-pollution”, “sea-level”, “sea-sickness”, among others) and 1 for “flow” as a subclass of “situation”. As the linguistic property “flow” is translated into an ontological class, the RSS tries to find ontological triples for the triples $\langle \text{sea?}, ?, \text{flow} \rangle$ and $\langle \text{flow}, ?, \text{Russian rivers} \rangle$. As the TSS fails to find valid ontological triples, the linguistic relation “flow” is ignored, and the algorithm searches for ontological triples by looking first for *ad-hoc* domain or is-a relations and second for indirect relations (with 1 mediating concept) between any of the *FAO* hits for “sea” and “Russian rivers”. As no results are produced, the TSS splits the compound “Russian rivers” into “Russian” and “rivers” and re-iterates to find other relevant ontologies for the new Query-Triples: $\langle \text{sea?}, \text{flow}, \text{rivers} \rangle$ or $\langle \text{sea?}, \text{flow}, \text{Russian} \rangle$, and $\langle \text{Russian}, ?, \text{rivers} \rangle$. There are 18 covering ontologies for the former triple: $\langle \text{sea?}, \text{flow}, \text{rivers/Russian} \rangle$ and 8 covering ontologies for the latter: $\langle \text{Russian}, ?, \text{rivers} \rangle$, from which only one, the *russiaB* ontology, contains the answers encoded in the Ontology Triples: $\langle \text{sea}, \text{flow_to}, \text{river} \rangle$ and $\langle \text{river}, \text{has_river}, \text{country} \rangle$ $\langle \text{country}, \text{has_political_fact}, \text{Russian} \rangle$. Among these covering ontologies, some of them contain more than 30 hits. Therefore, exploring all possible relationships among all the hits in so many potentially relevant ontologies, while only one contains the answer, is

⁵² Answering times can vary depending on the computer specifications, load of the server and network connections.

time consuming as it implies many calls to the plugins (the major bottleneck in the system) and longer searches on the indexes. For this query 2254 SeRQL queries are needed. As discussed in Chapter 5.5 there is loss of information regarding the directionality of the predicates when mapping the NL query into Query-Triples, as in the given example the Query-Triple subject (the *wh-query term* “sea?”) does not necessarily correspond to the subject of the NL query (“Russian rivers”). Similarly the subject of the NL query does not necessarily map to the subject of the Onto-Triples. This is because the way the information is presented in the ontologies differs from the way the user formulates the query and the directionality is given by the ontologies. It is not possible to limit the search space based on directionality without affecting recall, e.g., in “Who stars in Bruce Almighty” valid answers are obtained by inspecting relationships in both directions, e.g., from the Onto-Triple <Bruce Almighty, starring, person?> in DBpedia and the Onto-Triple <person?, appeared_in, Bruce_almighty> in a movie ontology.

The screenshot shows the POWERAQUA web interface. At the top, there is a search bar with the query "Which sea does the russian rivers flow to?". To the right of the search bar are buttons for "Ask!" and "Examples". Below the search bar, there is a checkbox for "Make Use of Watson" and a "Log in" link. The main content area displays the query triple: [sea] -- flow -- rivers russian. It shows 1 mapping in the ontology <http://kmi-web07.open.ac.uk:8080/sesame/russiaB>. The mapping is ranked 1 and shows the following triple: [sea](#) (sea equivalentMatching) [flow_to](#) (flow_to equivalentMatching) [river](#) (river synonym). Below this, there are 3 answers listed in a table:

Answer	Trust	Star
Azov_Sea (Azov_Sea)	Don	☆
Caspian_Sea (Caspian_Sea)	Volga	☆
Black_Sea (Black_Sea)	Dnepr	☆

Below the table is a "Save trust values" button. At the bottom of the first section, there are buttons for "Show all element mappings" and "Find these answers in Yahoo". The second section shows the query triple: [russian] -- null -- rivers. It shows 1 mapping in the same ontology. The mapping is ranked 1 and shows the following triple: [river](#) (river synonym) [has_major_river](#) (has_major_river ontology_ad_hoc) [country](#) (country ontology_ad_hoc). Below this, there are 4 answers listed in a table:

Answer	Trust	Star
Neva (Neva)	Russia	☆
Don (Don)	Russia	☆
Volga (Volga)	Russia	☆
Dnepr (Dnepr)	Russia	☆

Below the table is a "Save trust values" button.

Figure 9.1 Screenshot for “Which sea do the Russian rivers flow to?”

The linguistically most complicated query in this evaluation is Q22: “Give me the list of restaurants which provide Italian food in San Francisco” (with Query-Triples: *<restaurants?, provide, Italian food>* *<restaurants?, provide, San Francisco>* and *<Italian food, ?, San Francisco>*). The answer is encoded in an ontology about restaurants from the university of Texas. The first Query-Triple leads to only a couple of restaurants that are approximate matches for the compound term, as partial answers, e.g., “Florentino Italian food”. The second Query-Triple produces several restaurants in San Francisco as partial answers. The third Query-Triple does not map to any Onto-Triple, therefore the TSS splits the compound “Italian food” and re-iterates. The term “food” produces more than 300 hits or mappings, while the term “Italian” produces 503 mappings as literals, among others. As a result of the re-iteration we obtain many OTs with answers, e.g.: *<MagicFluteRestaurant, foodType, Italian (exact literal mapping for "Italian")>* *<MagicFluteRestaurant, isIn, SanFrancisco>*. Unfortunately it also produces noisy results like *<DragonRestaurant, foodType, fastfood (approximate literal mapping for "food")>* *<DragonRestaurant, isIn, SanFrancisco>* (nevertheless, the noisy results should be filtered or ranked last by the merging and ranking component).

In sum, we found that the tool was able to answer correctly more than half of the proposed queries, with most of the failures being due to lexical (syntactic) issues rather than the multi-ontology matching mechanism. The biggest group of PowerAqua failures are because relevant mappings could not be found. For instance, PowerMap fails to find the ontological entity “Spain” in Q56: “List some Spanish islands”⁵³, “CaliforniaRegion” in Q5: “Give me all Californian dry wines”, “brain tumor SY NCI” in Q53: “What kind of brain tumours are there?”, “person” in Q18: “Give me all the articles written by people from KMi”, “dessertDishes” in Q50: “Which desserts contain fruits and pastry”. Even when the relevant mappings were found, they were discarded by PowerMap’s

⁵³ Some of these errors have been solved a posteriori of performing this evaluation, e.g., by augmenting the search space with WordNet derived words (*Spain* is a derived word for *Spanish*).

filtering heuristics. For example in Q25: “Where are Sauternes produced?”, PowerMap correctly maps the linguistic term "Sauternes" to the wine "Sauterne", but this mapping is not related to a region, leading to a failure at the next stage of the process. PowerMap had indeed identified the mapping which would have led to the answer ("SauterneRegion", which is located in "Bordeaux"), however, this mapping was discarded because PowerMap considered it less likely to be correct than the exact mapping to "Sauterne".

In fact, many of these errors are the consequence of poorly modeled or incomplete ontologies, i.e., filtering heuristics and iterative algorithms are hampered by the presence of, for example, redundant and disconnected terms which may lead to a mismatch between the question, as phrased by the user, and the ontologies. For instance, in Q60: “Which terrorist organization performed attacks in London?”, PowerMap correctly maps the linguistic term “London” to the instance representing the city, but this instance is not related to the entity “terrorist organizations” leading to a TSS failure. PowerMap had indeed identified the mapping which would have led to the answer (the approximate literal “London, United Kingdom”), however this mapping was discarded because PowerMap considered it less likely to be correct than the instance mapping “London”. If the literal “London, United Kingdom” had an ontological relation to the instance “London”, or this one would have been ontologically connected with the instance “United Kingdom” (also connected to “terrorist organizations”), PowerAqua would have found the answer, as it correctly does for the similar query Q61: “Which are the main attacks that took place in the United Kingdom?”. Another example is the query Q69: “Which drug can treat colds and reduce headache?” in which the term “colds” maps to “commonCold”, which is not connected to the class “coldCoughCareProduct” that leads to the answer (*<ibuprofen, is-a. coldCoughCareProduct><ibuprofen, is-a, drugMedicine>*).

Once the relevant mappings are found, the TSS only failed to translate three of the queries to ontology triples for different unexpected reasons. For example in Q68: “What drugs can be used for reducing fever?”, the answers are found in the subclasses of the single conjunctive class “FeverReducingDrug”, which is not connected to the mapped class “Fever”; In Q45: “In which

country is Mizoguchi?” where the relevant ontology only says that Riichiro Mizoguchi is affiliated to Osaka university and the algorithm can not yet infer the fact that Osaka is in Japan from another ontology; and in Q15: “Who presented a poster at ESWC 2006?” where the TSS fails to map the term ESWC 2006 in the relevant ontology about the 2006 European Semantic Web Conference, as there is not an explicit ontological element to represent such a event. Poor modelling in the ontologies can also affect recall in the number of answers, for instance in Q46: ”Find all rivers in Asia” the answer “Amu Darya” is found for the direct Onto-Triple (*<River, locatedIn, Asia>*) and therefore, the algorithm stops searching in that relevant ontology without exploring the indirect relationship “river-country-Asia”, which contains most of the answers in that ontology.

These errors highlight how designing a real-time generic algorithm, that makes no assumptions a priori over the structure of the datasets, in the SW scenario is not only challenging because of its scale but more importantly because of its enormous heterogeneity, as entities are modeled at different levels of granularity and with different degrees of richness. In what follows we draw the conclusions obtained from this evaluation.

9.2.4 Conclusions

Our evaluation shows promising results, proving that it is feasible to answer questions with not just one but many ontologies selected on the fly in a reasonable time frame. Indeed, in the first evaluation we obtained a success rate in answering questions of about 70% over a data set of 69 queries. The average answering time was 15.39 seconds, with some queries being answered within 0.5 seconds.

This evaluation has highlighted an illustrative sample of problems for any generic algorithm that wishes to explore SW sources without making any a priori assumptions about them:

- Firstly, such algorithms are not only challenged because of the *scale* of the SW but more importantly because of its considerable *heterogeneity*, as entities are modelled at different

levels of granularity and with different degrees of richness. Under this perspective, the TSS algorithm has performed well.

- Secondly, while the distinctive feature of PowerAqua is its openness to unlimited domains, its potential is overshadowed by the *sparseness* of the knowledge on the SW. Indeed, in the evaluation presented in Chapter 10, we found that the content collected by semantic search engines such as Swoogle and Watson only covers 20% of the query topics put forward in the TREC 9 and 2001 (<http://trec.nist.gov>) IR collections. To counter this sparseness, the PowerAqua algorithms maximize recall, which leads to a decrease in accuracy and an increase in execution time.
- Thirdly, in addition to the sparseness, most of the identified ontologies were *sparsely populated* with instance data. This caused PowerAqua's failure to retrieve a concrete answer in some cases even when a correct mapping of the query was found in an ontology.
- A fourth aspect that hampered our system was the existence of many *low quality ontologies* which contained redundant, unrelated terms (causing the selection of incorrect mappings), presented unclear labels (thus hampering the system's ability to align query terms and ontology labels) or lacked relevant domain and range information (thus requiring more complex and time-consuming SeRQL queries or being unable to fill in the missing information in order to fully understand a query).
- Finally, as a fifth aspect, we note the yet *suboptimal performance of ontology repositories and semantic search platforms* to query large datasets. This limits the amount of information PowerAqua can acquire in a reasonable amount of time and hence prevents the system from considering all available information.

Summing up, our evaluation shows that good results can already be obtained, despite the limitations imposed by a still sparse SW, with limited quality control. This version of the tool was able to correctly answer almost 70% of the proposed queries. However, we believe that as the SW grows to

become the biggest source of semantic knowledge and the performance of the search engines improves, there will be more chances to fill in the missing information in order to understand a query and find its answers.

9.3 Evaluation 2: Merging and Ranking

In Chapter 8 we presented the algorithms for combining and ranking answers (given as ontological facts) obtained from multiple distributed heterogeneous data sources. The merging and ranking component includes merging and fusion algorithms that aggregate, combine, and filter ontology-based search results, and three different ranking algorithms that sort the final answers according to different criteria such as popularity, confidence and semantic interpretation of the results. These methods have been integrated in PowerAqua and evaluated in the context of a multi-ontology QA task. The question we try to answer here is: are aggregated answers from many heterogeneous, independent, semantic sources better than answers derived from single ontological facts?. The experiments presented here confirm that the quality of derived answers can be improved by cross-ontology merging and ranking techniques.

In this section we describe the evaluation of PowerAqua's merging and ranking capabilities for queries that require to be answered by combining multiple facts from the same or different ontologies. The main goal of these techniques is to improve the precision of the obtained answers while minimizing the loss in recall. Therefore, the design of this evaluation is focused around two main questions:

- How do we measure if the quality of the collective results obtained after merging and ranking are better than the individual answers?
- Which datasets are more suitable to be used for this evaluation?

Because there are different steps in the merging and ranking process that can influence the final quality of the answers, we have divided the evaluation in three main stages:

- Evaluation of the efficiency and effectiveness of the fusion algorithm.
- Evaluation of the level of filtering performed by the merging algorithm over the initial set of answers retrieved by PowerAqua.
- Evaluation of the three proposed ranking algorithms applied to the final set of answers obtained after the merging process.

Although there is a lot of semantic data available nowadays, there is a lack of datasets to formally evaluate cross-ontology capabilities for systems such as PowerAqua. Thus, this evaluation was conducted using our own benchmark, which comprises:

Ontologies and Knowledge Bases: To represent the information space with the purpose of obtaining a representative set of queries, which could be correctly mapped by PowerAqua into several ontological facts, preferably across different ontologies, additional metadata was collected with respect to the previous experiments in Section 9.2. We collected around 4GBs of data stored in more than 130 Sesame repositories. Each repository contains one or more semantic sources. We have collected in total more than 700 documents. The dataset includes high-level ontologies, e.g., ATO, TAP, SUMO, DOLCE and very large ontologies, e.g., SWETO (around 800,000 entities and 1,600,000 relations) and the DBpedia Infoboxes (the 2008 version, which comprised around 1GB of metadata). Even though PowerAqua can access larger amounts of SW data through Watson, in this experiment we decided to use a substantial static dataset in order to make these experiments reproducible.

Queries: We collected a total of 40 questions selected from previous PowerAqua evaluations and query logs obtained from the PowerAqua website⁵⁴, which were complex enough to require merging or ranking in order to obtain accurate and complete answers. These are factual questions that

⁵⁴<http://technologies.kmi.open.ac.uk/poweraqua/evaluation.html>

PowerAqua maps into several OTs, each of them producing partial answers. Merging and ranking is needed for these queries to generate a complete answer, or to rank between the different interpretations.

Judgments: In order to evaluate the merging and ranking algorithms a set of judgments over the retrieved answers is needed. To perform this evaluation two ontology engineers provided a True/False manual evaluation of answers for each query. Precision and recall were selected as evaluation metrics.

The construction of this benchmark was needed due to the lack of SW standard evaluation benchmarks comprising all the required information to judge the quality of the current semantic search methods (Fernandez, M. et al., 2009).

9.3.1 Evaluating the fusion algorithm

The gold standard for the evaluation of the fusion algorithm was created by manually annotating the answer sets produced by the 40 test queries. For each answer set, subsets of identical answers were identified. The generated gold standard was compared to the fusion produced by the merging algorithm and standard precision and recall measures were calculated. Each pair of answers correctly assigned to the same subset was considered a “true positive” result, each pair erroneously put into the same subset constituted a “false positive” result, and each pair of individuals, which were assigned to different subsets, while being in the same subset in the gold standard, represented a “false negative” result. The results are shown in Table 9.3.

Table 9.3 Test results of the co-reference resolution stage

Gold standard size	Precision	Recall	F1-measure
1006	0.946	0.931	0.939

When analysing the results we found that most errors of the merging stage were caused by:

- Syntactically dissimilar labels for which no synonyms could be obtained from WordNet, e.g: #SWEET_17874 (Longview/Gladewater), or grammatical mistakes (like “she_sthe_one” instead of “she_the_one”).
- Homonymous or syntactically similar labels for different entities.
- Incorrectly modelled ontologies, which contain duplicate instances under different URIs: e.g., in SWETO the city of Houston, Texas has 5 distinct URIs. Since such errors were not caused by the fusion algorithm, they were not counted during the evaluation experiments (although they would affect the user experience).

9.3.2 Evaluating the filtering performed by the merging algorithm

A major advantage of merging the multiple answers derived by PowerAqua is that irrelevant answers are filtered out (eliminated). The filtering obtained by the merging algorithm allows PowerAqua, on the one hand to eliminate duplicate information by means of fusing redundant answers together and, on the other hand, to compose a complete answer using different subsets of partial responses. The filtering of duplicated and partial information helps to eliminate non-relevant responses from the initial set of results. The following measure is used to compute the level of non-relevant results filtered by the merging algorithm:

$$f_q = \frac{|R_q| - |R_{qm}|}{|R_q|}$$

Where f_q is the percentage of filtering for the query q , R_q is the set of initial results retrieved by PowerAqua for the query q and R_{qm} is the set of answers that remain after merging. Note that, for simplicity, we consider that all the eliminated answers are irrelevant. This is not necessarily true when the merging algorithm intersects partial answers. For those cases, the rate of false positives (or number of relevant results lost in the filtering process) has been computed (section 9.3.1) and discarded as irrelevant. Results are presented in Section 9.3.4.

9.3.3 Evaluating the ranking algorithms

In this section we present the evaluation of the three ranking algorithms detailed in Chapter 8.4 in terms of precision and recall. As the gold standard for the evaluation we consider the completed list of answers for query q , including all the potential relevant and irrelevant results as the unsorted list of answers obtained after the merge step, $R_{q,m}$. For each ranking metric we consider as retrieved list of answers for the query q the first ranked cluster ($C@1$). Taking into account this, we define precision and recall as:

$$P_q = \frac{|\{Rel_q \cap C@1_q\}|}{|C@1_q|}, R_q = \frac{|\{Rel_q \cap C@1_q\}|}{|Rel_q|}$$

Where: P_q is precision for query q , R_q is recall for the query q , Rel_q is the set of relevant answers included in $R_{q,m}$ for the query q and $C@1_q$ is the set of retrieved answers, or answers included in the first ranked cluster.

Once these measures have been defined we compare the results obtained by our three different ranking metrics against our baseline, $R_{q,m}$. For the ranking based on confidence the precision is computed not just for the first ranked cluster $C@1$ but also for the union of the first two clusters $C@1 \cup C@2$. This is because the most accurate ranking algorithm is confidence and therefore, both confidence clusters are used in the combined ranking.

9.3.4 Results

The results of our experiments for the 40 queries listed in Table 9.4 are reported in Table 9.5 and Table 9.6. Table 9.5 contains the queries merged by union while Table 9.6 contains the results for the queries merged by intersection and condition. The different columns of the tables represent:

- The type of merging done for that query (U=union, I=intersection, C=condition in Table 9.5 and Table 9.6) / the number of ontologies involved in the merging process.
- The percentage of irrelevant queries filtered by the merging algorithm.

- The precision obtained for the set of answers returned after the merging process (the baseline ranking).
- The error type as explained below.
- Precision/Recall measures for the confidence ranking at the level of the first cluster $C@1$.
- Precision/Recall measures for the confidence ranking at the level of the first two clusters $C@1 \cup C@2$.
- Precision/Recall measures for the popularity ranking at the level of the first cluster $C@1$.
- Precision/Recall measures for the semantic similarity ranking at the level of the first cluster $C@1$.
- Precision/Recall measures for the combined approach at the level of the first cluster $C@1$ with the target of optimizing recall.
- Precision/Recall measures for the combined approach at the level of the first cluster $C@1$ with the target of optimizing precision.

An empty set $\{\}$ represents the case that no answer was retrieved for that cluster while – indicates that the query generates only 1 unique answer after merging, and therefore there is nothing to rank.

Table 9.4 List of queries used for the merging and ranking evaluation

Q1. What are the main rivers in Russia
Q2. Show me the films of Jenifer Aniston
Q3. Which Russian rivers flow into the Black Sea?
Q4. Which Russian rivers flow into the Azov Sea?
Q5. Give me the paper(s) written by Enrico Motta
Q6. Who play in Nirvana?
Q7. Give me all cities of Spain.
Q8. Which countries speak Arabic and English language?
Q9. Which languages are spoken in Islamic countries
Q10. Which languages are spoken in countries in Eastern Europe
Q11. which rivers flow in European countries.
Q12. Give me mountains in countries near the black sea
Q13. Show me the publications of Enrico Motta.

- Q14. Find me university cities in Japan
- Q15. Which countries are members of the EU?
- Q16. Give me the cities which are in Texas.
- Q17. How many cities does Rhode island have?
- Q18. List me all rivers crossing USA.
- Q19. Give me cities in Virginia.
- Q20. Find all the lakes in California.
- Q21. What states are next to Arizona?
- Q22. What is the capital of Maryland?
- Q23. Which state is Kalamazoo in?
- Q24. Where is san Diego?
- Q25. How many states are there in USA?
- Q26. Tell me all the projects related to academics in akt.
- Q27. How many cities in the USA are located in Georgia?
- Q28. Give me a French restaurant in alameda?
- Q29. Where is Houston?
- Q30. What mountains are in Alaska?
- Q31. Which organizations participate in humanitarian aid?
- Q32. List me all films with Tim Robbins and Morgan Freeman
- Q33. Which RBA banks are situated in Switzerland
- Q34. Who stars in Bruce Almighty
- Q35. What are the publications of Marta Sabou or/and Frank Van Harmelen?
- Q36. Who belongs to the open university
- Q37. In what state is Salem?
- Q38. San Antonio is in what state?
- Q39. Name the president of Russia
- Q40. Which countries are traversed by the rivers that flow into the Caspian sea?

Table 9.5 Test results for union queries

	Type /N° Ont.	Filter	Baseline		P/R confidence		P/R Pop	P/R Sem. Sim.	P/R Combined	
			P	Er	C@1	C@1 U C@2			+ R	+ P
Q ₁	U/3	0.03	0.97	M	0.97/0.88	0.97/1	1/0.03	0.97/1	0.97/0.88	1/0.03
Q ₂	U/3	0.53	1	-	1/0.66	1/1	1/0.73	1/1	1/1	1/0.97
Q ₅	U/4	0	1	-	1/0.4	1/0.6	{}	1/0.97	1/0.97	1/0.97
Q ₆	U/4	0.27	0.77	R,I	0.66/0.4	0.5/0.4	1/0.4	0.77/1	0.77/1	{}
Q ₇	U/6	0.38	1	-	1/0.53	1/1	1/0.29	1/1	1/1	1/0.52
Q ₁₃	U/2	0.91	1	-	1/0.81	1/1	1/0.08	1/1	1/0.81	1/0.09
Q ₁₅	U/5	0.55	1	-	1/0.47	1/1	1/0.35	1/1	1/1	1/0.47
Q ₁₆	U/7	0.31	1	-	1/1	1/1	1/0.32	1/1	1/1	1/1
Q ₁₇	U/8	0.25	0.35	I,M	1/1	1/1	1/0.14	0.53/1	1/1	1/1
Q ₁₈	U/3	0.13	1	-	1/1	1/1	1/0.13	1/1	1/1	1/1
Q ₁₉	U/8	0.55	0.94	I	1/1	0.94/1	1/0.61	0.94/1	0.94/1	1/1
Q ₂₀	U/2	0.05	1	-	1/1	1/1	1/0.05	1/1	1/1	1/1
Q ₂₁	U/4	0.53	0.75	R	1/1	0.75/1	1/0.16	1/1	0.75/1	1/1
Q ₂₂	U/6	0.36	0.28	I	1/1	0.66/1	0/0	0.16/0.5	0.5/1	0.5/0.5
Q ₂₃	U/4	0.4	0.5	R	1/1	1/1	0.5/1	0/0	0.5/1	0.5/1
Q ₂₄	U/8	0.57	0.12	R	1/1	0.12/1	1/1	0/0	0.12/1	1/1
Q ₂₅	U/10	0.64	0.88	I,M	1/1	0.96/1	0.96/0.98	0.9/1	0.95/1	0.98/1
Q ₂₉	U/9	0.21	0.45	R	0.45/1	0.45/1	1/0.4	0.4/0.8	0.45/1	0.45/1
Q ₃₀	U/3	0.08	0.95	M	1/1	0.95/1	1/0.09	0.95/1	0.95/1	1/1
Q ₃₁	U/3	0	1	-	1/0.5	1/1	{}	{}	1/0.5	1/0.5
Q ₃₄	U/3	0.29	0.74	I	1/0.07	0.74/1	1/0.07	0.74/1	0.74/1	1/0.07
Q ₃₆	U/5	0.15	0.14	I	1/0.32	0.14/1	1/0.02	1/0.77	1/0.77	1/0.32
Q ₃₇	U/3	0.5	0.66	R	1/1	0.66/1	1/0.5	0.66/1	1/1	1/0.5
Q ₃₈	U/6	0.3	0.57	M,I	1/0.25	1/0.5	1/0.5	0.57/1	0.75/1	1/0.5
Q ₃₉	U/2	0	0.33	I	1/1	0.33/1	{}	0.33/1	1/2	{}
Avg	4.84	0.32	0.74		0.96/0.77	0.81/0.94	0.82/0.31	0.72/0.85	0.86/1	0.86/0.66

Table 9.6 Test results for intersection and condition queries

	Type /N° Ont.	Filter	Baseline		P/R confidence		P/R Pop	P/R Sem. Sim.	P/R Combined	
			P	Er	@1	C@1 U C@2			+ R	+ P
Q ₃	I/3	0.96	1	-	-	-	-	-	-	-
Q ₄	I/3	0.93	1	-	-	-	-	-	-	-
Q ₈	I/3	0.97	1	-	1/1	1/1	{}	1/1	1/1	1/1
Q ₉	C/4	0.96	0.88	I	1/0.87	0.88/1	{}	0.88/1	0.88/1	1/0.87
Q ₁₀	C/3	0.92	1	-	1/1	1/1	{}	1/1	1/1	1/1
Q ₁₁	C/13	0.88	1	-	1/0.77	1/1	1/0.07	1/1	1/1	1/0.77
Q ₁₂	C/4	0.78	0.75	M	1/1	0.75/1	{}	0.75/1	0.75/1	1/1
Q ₁₄	I/8	0.98	1	-	1/1	1/1	1/1	1/1	1/1	1/1
Q ₂₆	I/2	0.72	0.97	I	0.97/1	0.97/1	1/0.03	0.97/1	0.97/1	0.97/1
Q ₂₇	I/6	0.99	0.83	R	0.83/1	0.83/1	0.83/1	0.83/1	0.83/1	0.83/1
Q ₂₈	I/1	0.99	1	-	-	-	-	-	-	-
Q ₃₂	I/1	0.99	1	-	-	-	-	-	-	-
Q ₃₃	I/2	0.84	1	-	1/1	1/1	{}	1/1	1/1	0/0
Q ₃₅	I/1	0.98	1	-	-	-	-	-	-	-
Q ₄₀	C/7	0.99	1	-	-	-	-	-	-	-
Avg	4.06	0.93	0.96		0.99/0.98	0.96/1	0.65/0.54	0.96/1	0.96/1	0.92/0.91

As we can see in the tables, the merging component is able to filter an average of 93% of irrelevant answers for intersections/conditions and 32% for unions. For instance, in Q14: “Find me university

cities in Japan”, 20 final answers are selected out of 991 partial answers, by intersecting 417 from cities with a university from *DBpedia ontology* and 574 from cities in Japan from *FAO*, *ATO*, *KIM*, *TAP*, *SWETO*. The average recall of the fusion algorithm, as shown in Section 9.3.1, is 0.93, i.e., a 0.07 loss in recall occurs in the case of intersections/conditions when partial answers representing the same individual are not recognized. The average precision of the fusion algorithm is 0.94, which indicates that most of the answers are correctly fused. The high precision and recall values obtained for the fusion algorithm, as well as the high percentage of filtering of irrelevant answers performed by this method, reflect PowerAqua’s ability to derive valid semantic interpretations of a query across ontologies.

The causes of the merging algorithm leading to irrelevant results in the final answers are:

- Incorrect modelling of the ontological elements in the Onto-Triples that lead to the answer (M). For instance in Q30: “What mountains are in Alaska?”, the instance Germany is given as an answer because it is defined as `rdf:type {country, mountain}` in one of the ontologies.
- An inaccurate semantic interpretation given by PowerAqua (I). For instance Q36: “Who belongs to the Open University?”, among Onto-Triples representing people that work for the Open University, there is an OT : `<organization, type, open universities>`.
- Retrieval of pragmatically irrelevant answers (R). For instance, the answer Houston to Q29: “Where is Houston?”.

These sets of errors are often filtered out afterwards by the ranking algorithms. As we can see in the tables, for the union queries all ranking methods are able to provide better precision than the baseline, with an increase of 0.22 points of precision for the best ranking algorithm, in this case ranking by confidence at $C@1$. This increase in precision is usually translated in a recall loss as in the case of the popularity ranking algorithm where recall drops to 0.31. However, the rest of the ranking metrics are able to keep the recall measure between 0.77 and 0.94. Finally, the best

combined approach is able to enhance by 0.12 percentage points the precision of the baseline without causing a drop in recall.

9.3.5 Discussion of the results

For the intersection and condition queries all the ranking methods are able to keep or increase the precision, except in the case of the popularity algorithm that decreases precision to 0.31 points. The same effect occurs with recall. All the ranking algorithms are able to provide levels of recall between 0.98 and 1, which means nearly no loss of relevant answers, except for the popularity ranking, which reduces recall to 0.54. The best ranking method for intersection and condition queries is the ranking by confidence at C@1. This ranking slightly increases precision by 0.03 points with respect to the baseline, keeping the level of recall at 0.98. Finally, for this set of queries, the best combined approach is able to preserve the same precision and recall values as the baseline: 0.96/1. In other words, the effect of ranking measures on intersection queries is neutral, this was expected as for intersection and condition queries the filtering has already eliminated most (if not all) of the inaccurate answers.

In summary, we can say that the best ranking method for both subset of queries is the ranking by confidence at C@1 that is able to produce a 0.22 percentage increase of precision for union queries and a 0.03 for intersection ones. Semantic similarity depends on being able to calculate the semantic interpretation of each Onto-Triple, but that's not the case if the Onto-Triple entities are not covered in WordNet, or the taxonomical information is not significant enough to elicit the meaning of the entity in the ontology. The worst ranking method in both cases is ranking by popularity. It drops precision by 0.14 points for union queries and by 0.31 points for intersection and condition queries. This is because popularity at C@1 (answers obtained from at least two ontologies) is empty in the cases in which no answers were fused from different ontologies (empty set {} being equivalent to 0/0 for precision/recall). Interestingly, in the 25 cases where C@1 is not empty, this measure gives precision 1 in 22 cases. Therefore, precision would have been closer to 1 than with any other ranking if we had chosen to put into C@1 all the answers with popularity 1, when there are not

answers with popularity 2 or higher (empty set $\{\}$ equivalent to 1/1 for precision/recall as all the answers are rank at the same level). The effect in recall is even worse, dropping to 0.31 for union queries and to 0.54 for the intersection and condition ones. At this early stage of the SW, PowerAqua's results are hampered by the knowledge sparseness and its low quality. We believe that any extension of the online ontologies and semantic data will result in direct improvements for both popularity (hampered by knowledge sparseness) and semantic similarity (hampered by low quality data) ranking measures.

Even with the different behaviour of these ranking methods, the combined algorithm is outperformed by the confidence ranking in terms of precision but it is able to improve the precision and recall ratio. Contrary to what was expected, maximizing precision does not improve the precision value on the combined measure. This is because the average measure was affected by queries in which none of the answers ranked high enough ($C@1=\{\}$).

These results confirm our initial hypothesis that the use of cross-ontological information to rank the retrieved answers helps to enhance the precision of the results, and therefore, to provide better answers to users. An important remark is that this increase of precision does not imply, in any case except for the popularity algorithm, a significant loss in recall.

9.3.6 Conclusions

In this work we present a set of merging and ranking algorithms that aim to integrate information derived from different knowledge sources in order to enhance the results obtained by PowerAqua. The experiments are promising, showing that the ranking algorithms can exploit the increasing amount of collectively authored, highly heterogeneous, online semantic data, in order to obtain more accurate answers to questions, with respect to a scenario where the merging and ranking algorithms were not applied. On the one hand, the merging algorithm is able to filter out a significant subset of irrelevant results. On the other hand, all the ranking algorithms are able to increase the precision of

the final set of, without significant loss in recall, thus showing a deeper semantic “understanding” of the intent of the question.

More specifically, the merging algorithm is able to filter out up to 91% (32% on average) for union-based queries, and up to 99% (93% on average) for intersection based queries. The fusion algorithm (the co-reference algorithm to identify similar instances from different ontologies) exhibited a 94% precision and 93% recall.

The best ranking algorithm (ranking by confidence) is able to obtain an average of 96% precision for union queries and 99% for intersection queries. An interesting, observed, side effect of this approach is that answers to some questions that are distributed across ontologies can only be obtained if the partial results are merged. In this case, the introduction of the merging algorithm provides PowerAqua with the capability to answer queries that cannot be answered when considering a single knowledge source.

The high precision values produced by the merging and ranking algorithms, which are responsible for amalgamating information from different sources, support the comparison with the idea of the *Wisdom of Crowds*⁵⁵. We further observe that it is known that the Wisdom of crowds only works if the crowd is diverse and free to think independently (Surowiecki, 2004), allowing it to converge on good solutions. Similarly PowerAqua works well where ontologies have different emphasis, to allow the assembly of composite answers, but also where overlaps between ontologies exist, to allow mapping and identification of ranking criteria, such as popularity. Both too much homogeneity and isolated "silo" ontologies would weaken our approach.

⁵⁵ A book written by James Surowiecki in 2004, primarily on the fields of economic and psychology, stating that “a diverse collection of independently-deciding individuals is likely to make certain types of decisions and predictions better than individuals or even experts.” (http://en.wikipedia.org/wiki/The_Wisdom_of_Crowds).

Another interesting side-effect of this approach is that, apart from the obvious advantage to the final user, the filtering of negative results and the ranking capabilities of the retrieval system increase its adaptability for other tasks, e.g., query expansion using SW resources.

An issue remains open: the use of our own dataset to perform the experiments. However, to our knowledge, the SW community has not yet proposed standardized benchmarks to evaluate semantic merging and/or ranking evaluation. Nonetheless, we have tested our algorithms with a significant amount of queries and large amounts of distributed semantic metadata (around 4GB).

Finally, as a future work, we believe that the trust propagation mechanism explained in Section 8.5, where the user can rank the answers as a way of giving feedback to the system, can be used to improve the ranking so that answers replicated across many ontologies do not bias less frequently occurring facts generated from specialist knowledge from trusted ontologies.

9.4 Evaluation 3: Usability study

While the previous evaluations of PowerAqua focused on accuracy and performance, in what follows we present the first usability results of PowerAqua as a NL interface to semantic repositories. The evaluation was carried out following the formal benchmark proposed for the SEALS⁵⁶ 2010 semantic search evaluation campaign, and focused on the interface and usability aspects. Hence, the evaluation procedures emphasize the users' experience and degree of satisfaction with each different search tool (Wrigley et al., 2010).

9.4.1 Evaluation approach: the SEALS evaluation campaign

The SEALS evaluation methodology consists of a two-phase approach in which tools with different interfaces, are evaluated and compared both in a fully automated fashion as well as within a user

⁵⁶ Results on the 2010 evaluation campaign on semantic search available at: <http://www.seals-project.eu/seals-evaluation-campaigns/semantic-search-tools> and SEALS D13.3: <http://about.seals-project.eu/deliverables>.

study. The evaluation comprises a series of reference benchmark tests that focus on evaluating the fundamental aspects of the tools in a strictly controlled environment, rather than their ability to solve open-ended, real-life problems. Thus, this evaluation cannot assess PowerAqua's ability to query multiple ontologies and fuse answers across sources in a large open scenario, but the user study is used to evaluate PowerAqua's performance and usability in a controlled scenario. It also allows us to directly compare the usability of different semantic search interfaces (keyword-based, from-based, NL-based).

For this usability study, 10 human subjects were given a number of tasks (questions) to solve using the Mooney geography dataset⁵⁷, including complex queries that enclose more than three concepts and include comparatives, superlatives and negations. The set of questions used for this experiment is presented in Table 9.7. For each task (question) the users could reformulate the questions. Since a number of the tested tasks had a high complexity level, the users may need to formulate various questions in order to get an answer. The subjects in the usability experiment are guided throughout the process by the SEALS software controller, which is responsible for presenting the questions and gathering the results and range of user-centric metrics. Metrics such as the time required to input a query, the time required to display the results and the number of attempts required to obtain the answer were collected. In addition, data regarding the user's impression of the tool is gathered using the System Usability Scale (SUS) questionnaire (Brooke, 1996), an extended questionnaire (similar to SUS in terms of questions but more detailed) and a demographics questionnaire, to find out whether a particular tool is suited for their target user group(s). Such questionnaires represent a well-known and often applied procedure in the domain of Human Computer Interaction to assess user satisfaction and to measure possible biases and correlations between the test subject characteristics and the outcomes of the evaluation.

⁵⁷ Provided by Ray Mooney and his team from the University of Texas at Austin, translated to OWL in order to evaluate ontology-based QA systems in (Kauffman, 2009). The geography OWL KB contains 9 classes, 11 datatype properties, 17 object properties and 697 instances.

Table 9.7. Set of queries used in the SEALS usability study.

Give me all the capitals of the USA?
What are the capitals of the states that border Montana?
Which rivers in Arkansas are longer than the Allegheny river?
What are the states that border the state with the smallest population?
Which states have cities with a city population over 50,000 and have a lake, a mountain, a road and a river?
Which rivers run through Arizona, Colorado, California and Nevada?
Which states have cities named Springfield?
What are the cities in states through which the Mississippi runs?
Which states which have elevations higher than what Pennsylvania has?
Which roads pass through Alabama, Mississippi, Louisiana, new Mexico, Florida, Arizona, California and Texas?
Which states border more than four other states?
Which states have a city named Columbia with a city population over 50,000?
Which lakes are in the state with the highest point?
Which states border on the state whose capital is Lincoln?
Tell me which rivers do not traverse the state with the capital Nashville?
What are the capitals of states that have cities named Springfield?
Which states have a capital with a city population over 100000 and have a lake?
Which states border more than three other states and have a mountain?
Which states have a higher point than the highest point of the state with the largest capital in the US?
Which states have an area greater than 1 Mil and border states that border states with a mountain and a river?

9.4.2 Results

This section presents PowerAqua's results and compares them with the results obtained in the first SEALS evaluation campaign conducted during summer 2010. In this campaign the emphasis was mostly on the usability experiment. The list of participant tools is shown in Table 9.8:

Table 9.8. List of participant tools in the SEALS user-based evaluation campaign

Tool	Description
K-Search	K-Search allows searching of semantic concepts in ontologies and documents using a form-based interface (Bhagdev et al., 2008).
Ginseng	Ginseng Guided Input Search Engine is a NL interface that restricts the user input via word lists extracted from the underlying ontology (Bernstein et al., 2006).
NLP-Reduce	NLP-Reduce is a NL query interface that allows users to enter full English questions, sentence fragments and keywords (Kaufmann et al., 2007).
PowerAqua	PowerAqua is an open multi-ontology QA system for SW using a NL user interface.

Results are presented in Figure 9.2. The metrics presented in Figure 9.2 are defined as follows in (Wrigley et al., 2010):

- The mean experimental time: how long on average the entire experiments (with the 20 pre-defined questions) took for each user.
- The mean extended questionnaire: shows the average response by the users to the extended questionnaire to establish the levels of satisfaction.
- The mean number of attempts: shows how many times the users had to reformulate their queries to obtain the answers with which they are satisfied.
- The mean answer found: is the rate between finding the appropriate answer after a number of attempts or “giving up”.

SUS scores have a range of 0 to 100. A score of around 60 and above is generally considered as an indicator of good usability. The results show that the difference in usability between k-search and ginseng is not significant (almost identical SUS scores), while for NLP-Reduce the score is much lower. PowerAqua is the system with the highest SUS score. Only PowerAqua received a score indicating a good user experience.

These scores are also consistent with the number of attempts the user required to formulate their query before they got satisfied with the answer or moved on (the mean answered found rate). For example NLP-Reduce required double the number of attempts than any of the other systems, while K-Search and PowerAqua found satisfactory answers twice as often as the other tools, with an average of two attempts to formulate their query. Subjects using NLP-Reduce, however, required more than five attempts. PowerAqua was the system where the users found the highest number of satisfactory answers, PowerAqua found suitable answers for over half (55%) of all attempts, and with the best precision (0.57) and recall (0.68). Still precision and recall values don't give the full picture, because of the presence of complex queries that enclose multiple concepts, modifiers and conjunctions, including comparatives, superlatives and negations, which are out of PowerAqua coverage. The cognitive process of the user for this kind of task, which requires them to formulate various queries in order to get an answer, can't be captured in terms of precision and recall. For instance, to find an answer to the comparative query “Which rivers in Arkansas are longer than the

Allegheny river?”, the user can ask “What is the length of the Allegheny river?” and “What is the length of rivers in Arkansas?”. Also, PowerAqua's mean query input time was significantly lower than any of the other tools’ and the overall time to answer an entire question was less than half its nearest competitor.

	k-search	ginseng	nlp-reduce	poweraqua
Mean experiment time (s)	4313.84	3612.12	4798.58	2003.9
Mean SUS (%)	44.38	40	25.94	72.25
Mean extended questionnaire (std)	47.29	45	44.63	80.67
Mean number of attempts	2.37	2.03	5.54	2.01
Mean answer found rate	0.41	0.19	0.21	0.55
Mean execution time (s)	0.44	0.51	0.51	11
Mean input time (s)	69.11	81.63	29	16.03
Min input time (s)	0.5	0.5	0.5	0
Max input time (s)	300.17	300.16	278.95	202.82
Mean overall question time (s)	257.25	216.19	246.95	109.51
Mean precision	0.44	0.32	0.16	0.57
Mean recall	0.61	0.32	0.55	0.68
Mean F-measure	0.46	0.27	0.21	0.57

Figure 9.2: Comparative of results from the usability phase⁵⁸

9.4.3 Discussion and Conclusions

The results are positive, giving an important insight into the usability of PowerAqua. PowerAqua is the system with the highest SUS score, which is consistent with the f-measure, the number of attempts the users required to formulate the query, and their degree of satisfaction.

Users of the NL tools often reported that they liked the flexibility of being able to pose NL queries to the system, compared with the rigid framework of form-based or graphic interfaces. Furthermore, they trusted the system, once: (1) they were informed about the limitations on the coverage of the system, i.e., the system not being able to handle superlatives, comparatives, negations, or very long conjunctive queries; and (2) after they had checked through the interface the

⁵⁸ Due to a bug in the PowerAqua SEALS wrapper, precision and recall had to be recalculated separately with a representative sample of queries pose by the users in the evaluation.

validity and provenance of the answers for the first queries. One subject stated that she liked the fact that such an input style "offered me alternative ways to discover the answer". In contrast to the other NL tools above, PowerAqua was capable of understanding quite complex queries, something which was appreciated by the subjects: "it can go beyond simple queries, like actual systems such as Google, to more complex combinations of queries". This preference for NL, in comparison to keyword-based tools, was also specified by one subject stating that she believed that "PowerAqua gave more precise and concise answers". Also, subjects liked the ability to check the provenance of the answers and the fact that they could see all the merged results generated by the various ontological translations.

On the negative side, NL QA systems are still affected by the habitability problem, namely that the user does not know what is possible to ask or why the system occasionally failed. In common with most of the other tools, it was noted that it was difficult to gain an understanding of exactly what information was available in the ontology. This was evident on the example "What are the states that border the state with the smallest population?", most users ask first "What is the population of all states?", to find that the least populated of all 51 states is Alaska, and then, they ask "What states border Alaska", which produced all kinds of different partial information regarding Alaska but not the states that border it. It took a while for the users to realize that either the ontology does not contain that information or that actually Alaska doesn't have any borders with other US states.

Also, users needed more than one attempt to figure out how to reformulate a query if it could not be understood in the first try, as stated in (Wrigley et al., 2010): "how NL QA systems can cope with the vast range of possible ways in which a query can be formulated is a significant challenge". We also found that since a number of the tested questions had a high complexity level and needed to be turned into two or more partial queries "subjects reported that they would have liked to have either used previous results as the basis of the next query or to have simply temporarily stored the results to allow some form of intersection or union operation with the current results set".

Our last observation is that some users are biased by the way they perform searches in Google. Although all of them liked the expressivity of NL for queries such as “What are the capitals of the states that border Montana?”, for other queries they preferred to use keywords, e.g. for the query “Tell me which rivers do not traverse the state with the capital Nashville?”, one user typed “Nashville city” and then “all state rivers”, and then she browsed through the ontological mappings to find her answer. Also, 8 out of 10 users participating in this evaluation are ontology engineers or experienced users in SW technologies, and we are aware that this fact may bias the results. At this stage, the PowerAqua interface assumes background knowledge from its users with regard to ontologies, rather than being designed for general web users (e.g. by hiding URIs and simplifying the amount of information presented to the users, etc.).

There is one main aspect that makes this particular evaluation of search tools more complicated than already existing benchmarks (like the ones developed for the TREC IR community, or the evaluation contests for mapping and ontology alignment algorithms⁵⁹), which is that different tools exhibit a wide range of search approaches (keyword-based, form-based, NL). In fact, the diversity of semantic technologies and the lack of uniformity in the construction and exploitation of data sources are some of the main reasons why there is still no general adoption of standard evaluation methods. In this sense, this evaluation has some disadvantages with respect to previous evaluations designed to evaluation PowerAqua or a NL QA system:

- The evaluation focuses on solving complex tasks, such as “Which states have a higher point than the highest point of the state with the largest capital in the us?”, however, they were not regarded as real or representative NL user queries. Queries are formulated similarly to the way the information is structured in the database and using the same vocabulary. Thus, it

⁵⁹ <http://oaei.ontologymatching.org/>

does not evaluate the mapping and disambiguation capabilities that any non-trivial QA system should have.

- It can only elicit usability measures of a system in a controlled closed scenario, rather than measuring the system's ability to answer user queries in an open-domain scenario.
- Some queries were misinterpreted, in particular at the end of the experiment (the duration of the experiment takes over 1 hour) or were considered ambiguous, e.g., for the query “Tell me which rivers do not traverse the state with the capital Nashville”, most of the users searched for rivers in all states, while one user searched only just in the states that are bordering Tennessee (whose capital is Nashville).

Summing up, the first two evaluations presented in this Chapter measured PowerAqua's effectiveness and performance in an open scenario, through user-centred experiments (Sections 9.2 and 9.3). The most critical aspect of the third experiment is benchmarking the user experience with the tool (Wrigley et al., 2010), thus this evaluation gives us a new and important insight into the usability of QA systems in general, and PowerAqua, in particular. The results from the SUS questionnaires and the feedback that we directly gathered from the users when carrying out the experiments are positive, and consistent with previous usability studies (Kaufmman et al., 2007) in which users prefer to use NL interfaces to search and query the knowledge stored in the repositories to interfaces using keywords, partial sentences and a graphical interface.

Chapter 10 Evaluation of PowerAqua with respect to Query Expansion on the Web (IR)

The evaluation presented in this chapter has been published as part of a paper at the International Conference on Semantic Computing in 2008 (Fernandez et al., 2008).

This is a collaborative work with the *Autonoma University of Madrid* (UAM) to evaluate PowerAqua as a QA interface to an IR system developed by the UAM. The work of the author is presented in sections 10.1, first step in Section 10.2, Section 10.4 and 10.6. The collaborative setup and results are presented in Section 10.3 and 10.5 respectively. For extended details on the collaborative benchmark produced for this evaluation refer to (Fernandez, M. et al., 2009).

10.1 Motivating scenario

The emergence of the SW makes it possible to begin experimenting with open semantic search systems that are not constrained by specific organizational ontologies, as is often the case today, but are able to exploit in an integrated way the combination of the information spaces defined by the SW and by the (non-semantic) Web. The research presented as part of this evaluation focuses on the added value of PowerAqua, as a NL interface to the SW, to contribute to the design of semantic retrieval technologies which can scale up to the open Web and are capable of: (1) bridging the gap between users and SW data, and (2) bridging the gap between the SW data and unstructured, textual information available on the Web.

In this evaluation, we report experiments with a new approach that builds upon PowerAqua and a pre-existing semantic IR system with complementary affordances. PowerAqua can search heterogeneous knowledge bases and generates answers to NL queries, thus, addressing the aim to bridge the gap between users and the SW. The second system, developed by the *Autonoma University of Madrid* (UAM) group and reported in (Castells et al., 2007)⁶⁰, supports semantic search, based on formal domain knowledge, over non-semantic World Wide Web documents.

⁶⁰ This tool is kindly provided by the UAM (Universidad Autonoma de Madrid: Nets.ii.uam.es) for our experimental research purposes, but its research and development is not part of this thesis.

Hence, it addresses the second aim by making ordinary Web pages open to semantic search. By coupling both systems we build a new system that:

a) Provides the user with the capability to query SW information using NL. PowerAqua's ability to answer NL queries makes the user interface more attractive than that of several search prototypes which rely on more complex ways to specify an information need (e.g., SPARQL queries). In addition, PowerAqua can retrieve a concrete answer when the appropriate semantic data is available.

b) Complements the specific answers retrieved during the QA process with a list of semantically ranked documents from the Web. The document-ranking module reported in (Castells et al., 2007) complements PowerAqua in two ways. If relevant ontologies exist and PowerAqua can provide an answer, it provides documentary evidence to help the user judge the validity of the answer. Alternatively, if PowerAqua cannot provide an answer, for example, because semantic data is not available for the topic, the user can still get an answer in the form of relevant documents.

Indeed, we are not aware of any system that provides these functionalities in an open scenario. Furthermore, this integration has allowed us to evaluate this semantic search system by reusing and adapting traditionally well-known IR evaluation benchmarks, in particular the TREC 9 and TREC 2001 (<http://trec.nist.gov/>). Our goal in selecting an IR collection is twofold. On the one hand, we aim to evaluate the query results retrieved by PowerAqua with an independently produced set of queries and document judgements. On the other hand, this allows us to evaluate the advantages of using the semantic information, retrieved by PowerAqua, for document retrieval in terms of precision and recall, defined as the fraction of the relevant documents that has been retrieved (recall) and the fraction of the retrieved documents that are relevant (precision).

In addition to evaluating the value of PowerAqua when coupled with a semantic IR tool, we have also analysed in Section 10.7 the feasibility of integrating PowerAqua with a standard Web search engine. The approaches proposed in this chapter take a step forward towards finding practical ways

to exploit the growing amount of semantic data that is available on the SW, and to potentially enhance current search technology on the World Wide Web.

The results show that ontology-based semantic QA capabilities can be used to complement and enhance keyword search technologies. The most important features of the proposed solution are:

- It uses both relevant semantic data drawn from the SW, when it is available, and non-semantic information found in standard Web pages, to answer user queries.
- It provides an innovative and flexible solution to the problem of integrating data found in these two sources by dynamically creating links between Web pages and semantic data, which keeps the two information spaces decoupled.
- It degrades gracefully to behave as an IR system when semantic data is not available or incomplete.

10.2 PowerAqua as part of an advanced IR system: overview.

Figure 10.1 depicts the two main steps of the overall retrieval process. The first step aims to bridge the gap between the users and SW data by retrieving answers from a set of ontologies as a reply to a user query expressed in NL. The second step aims to bridge the gap between the SW data and unstructured, textual information available on the Web by augmenting the answer retrieved from ontologies with relevant Web documents. PowerAqua addresses the first aim by providing a NL interface onto heterogeneous semantic data. The work reported in (Castells et al., 2007) addresses the second aim by making ordinary Web pages open to semantic search. Exploiting a large amount of metadata brings the advantage of retrieving Web documents without any potential domain restriction. The key step to achieve this aim lies in linking the semantic space to the unstructured content space by means of the explicit annotation of documents with semantic data, in such a way that the two information sources remain decoupled (without hardwiring the links between Web

pages and semantic data and without the aim to populate the ontologies, but rather to identify already available semantic knowledge within the documents).

Step1- Understanding the natural language user request and retrieving an answer in the form of pieces of ontological knowledge: the user's NL query is processed by PowerAqua. This component operates in a multi-ontology scenario where it translates the user terminology into the terminology of the available ontologies and retrieves a list of ontological entities as a response. This step has two main advantages. First, user interaction is eased by allowing NL queries, increasing the usability of the system. Second, the response is obtained from an unlimited number of ontologies which cover an unrestricted set of domains. As we have seen through this thesis, given a user query, PowerAqua is able to: 1) select the ontologies potentially relevant to the user query; 2) choose the appropriate ontology(ies) after disambiguating the query using its context and the available semantic information and; 3) extract from this ontology(ies) an answer in the form of ontological entities or individuals.

Step2- Retrieving and ranking relevant documents based on the previously retrieved pieces of ontological knowledge: once the pieces of relevant ontological knowledge have been returned as an answer to the user's query by PowerAqua, the system reported in (Castells et al., 2007) performs a second step to search for documents associated to those ontological entities retrieved as answers. This phase retrieves documents without any domain restriction, and ranks them using a semantic ranking model which scales up to large document repositories. Similarly to an annotation process, in Castells et al. the document collection is automatically indexed in terms of the ontology entities prior to the use of the system (for scalability reasons inverted indexes are used and not embedded annotations). The ranking algorithm, as described in (Castells et al., 2007), is based on an adaptation of the traditional vector-space IR model where documents and queries are represented as weighted vectors (annotations are assigned a weight that reflects how relevant the semantic entity is considered to be for the document - weights are computed based on the frequency of occurrence of the semantic entity in each document).

Both steps are carried out using four main architectural components (Figure 10.1): (1) the ontology indexing module, which pre-processes (online) available semantic information; (2) the PowerAqua module, which answers the NL query in the form of ontology triples; (3) the annotator module, which generates a concept-based index between the semantic entities and documents; and (4) the document retrieval and ranking module, which retrieves and ranks documents relevant to the ontology triples obtained by PowerAqua. The output of the system consists of a set of ontology elements that answer the user's question and a complementary list of semantically ranked relevant documents.

In what follows we explain the experimental setup and provide a brief summary of the evaluation results. For more details of the work and architecture of the proposed system refer to (Fernandez et al., 2008).

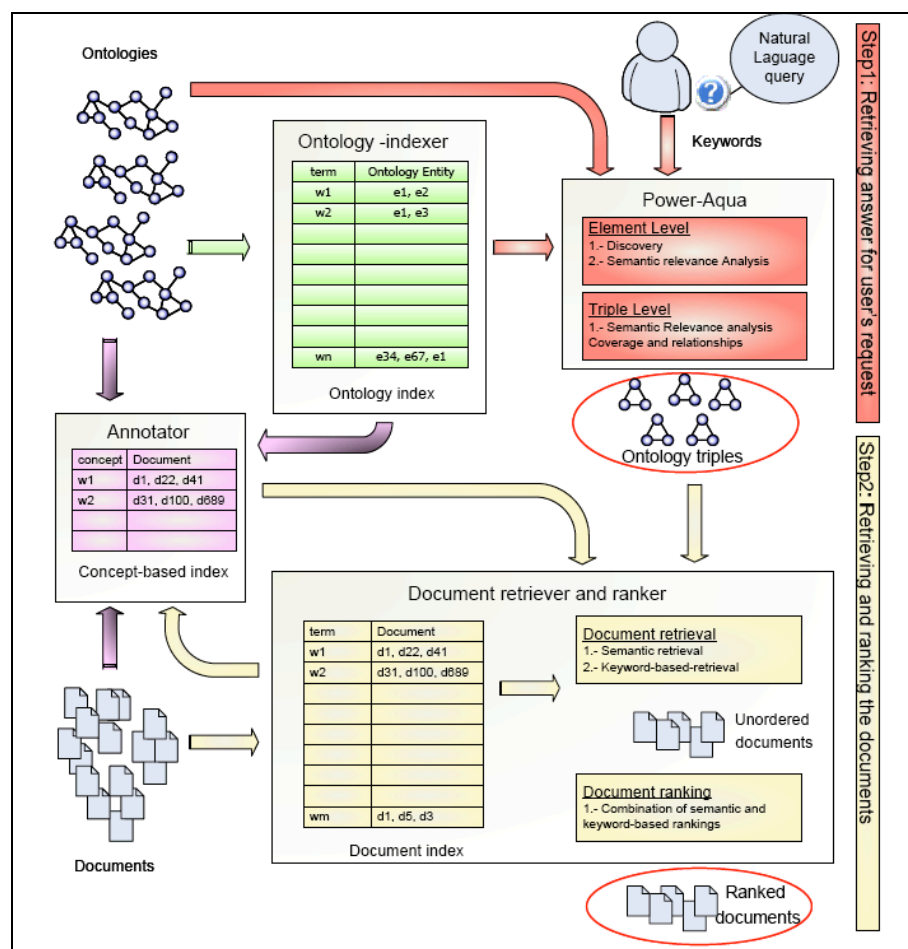


Figure 10.1 Architecture for the integrated system

10.3 Experimental Setup

Here we summarize the constructed evaluation benchmark, developed in collaboration with the UAM group. More details on this novel evaluation benchmark for cross-comparison between classic IR and ontology-based IR models, can be found in (Fernandez, M. et al., 2009).

To evaluate our semantic search system we required: a text collection, a set of queries and corresponding document judgments, ontologies that cover the query topics, and knowledge bases that populate the ontologies. The source of this semantic data should be independent of the query and text collection. The document collection and the set of queries and judgments are taken from the datasets used in the TREC IR tracks (<http://trec.nist.gov/>). The semantic data is obtained from the publicly available SW data on the Web.

The Documents Collection and Queries: The evaluation benchmark is constructed by taking the TREC 9 and TREC 2001 test corpora as a starting point, because these provide us with an independently produced set of queries and document judgments. This IR collection comprises: 10 GB of Web documents (containing 1.69 million Web pages) known as the TREC WT10G collection; 100 queries, corresponding to real user logs requests; and the list of document judgments related to each query (Bailey et al., 2003). These judgments allow the quality of the information retrieval techniques to be calculated using standard precision and recall metrics.

The Ontologies: Because the SW is still sparse and incomplete (Sabou et al., 2007), many of the query topics associated with WT10G were not covered by it at the time we performed this evaluation. Indeed, we only found ontologies covering around 20% of the query topics, the ones used for this evaluation. In the remaining cases, ontology-based techniques cannot yet be used to enhance traditional search methodologies. We identified 40 public ontologies on the SW, through Watson and Swoogle, which potentially covered a subset of the TREC domains and queries. These ontologies are grouped in 370 files comprising 400MB of RDF, OWL and DAML. In addition to these ontologies, to approximate a fair scenario, our experiments also accessed a bigger set of

random ontologies, specifically the 3GB of semantic data stored in Sesame and indexed with PowerMap's indexing structures that were used in the PowerAqua evaluation presented in Chapter 9.2.

The Knowledge Bases (KBs): Sparseness is even a bigger problem for KBs than for ontologies (KB often refers to the collection of instances of the concepts defined in the ontology, where the ontology provides the structure of the knowledge stored in the KB). At the time of this evaluation, current publicly available ontologies contained significant structural information in the form of classes and relations. However, most of these ontologies were barely populated. As a result these available KBs were still not larger enough to support large-scale semantic search testing. To overcome this limitation and provide a medium-scale test experimentation of our algorithms, some of the 40 selected ontologies have been semi-automatically populated from an information source which is independent from the document collection: Wikipedia. Wikipedia is a public encyclopaedia comprising knowledge about a wide variety of topics. In this way, we endeavour to show how semantic information, publicly available on the Web, can be applied to enhance keyword search over unstructured documents. The semi-automatic ontology population algorithm, explained in (Fernandez, M. et al., 2009), generated around 20,000 triples distributed along the 40 pre-selected ontologies.

The experiments were designed to compare the results obtained by three different search approaches with increasing levels of semantic awareness: three traditional keyword-based systems (Lucene, the best TREC automatic, and the best TREC manual) and PowerAqua both as a simple query expansion module and as part of the semantic search engine described in the previous section. The aims are twofold. On the one hand, we are able to evaluate the results retrieved by PowerAqua. On the other hand, we evaluate the advantage of semantically processing documents, rather than just using the semantic information to complement user queries. PowerAqua results are then evaluated in relation to their use on:

(1) Keyword Search: This type of search is performed with the widely used text search engine Lucene.

(2) Semantic query expansion: Semantic information is used just to expand the user query. PowerAqua processes the user query and extracts a list of semantic terms as an answer to this query. This list is added to the original query and used to perform a traditional keyword search.

(3) Semantic retrieval: In this search approach PowerAqua is integrated with a semantic retrieval system (Castells et al, 2007) that uses the semantic information retrieved by PowerAqua for its semantic document retrieval and ranking subsystem.

(4) Best TREC automatic search: the approach used by the best TREC search engine, which automatically uses as query just the title section of each topic in the TREC collection, as seen in Table 10.1.

(5) Best TREC manual search: the approach used by the best TREC search engine, which manually generates the queries using information from the title, the description and the narrative for each topic (described in Section 10.4).

The two semantic experiments 2) and 3) are compared with the results obtained in the experiment 1). The best TREC search results (title only and manual) 4) and 5) correspond to the best search engines of the TREC 9 and TREC 2001 Web track competitions. They are used as a reference, as it is not our goal to compare ourselves with the best TREC search engine. The document retrieval and ranking algorithm used in (Castells et al, 2007) depend on the quality of the index keyword search mechanism used during the annotation process (Lucene).

10.4 Adapting the TREC queries and list of query topics

The selection of the TREC queries was constrained in two ways: a) the queries must be formulated in a way suitable for QA systems; this means queries like “discuss the financial aspects of retirement planning” (topic 514) cannot be tackled because they are navigational and not query response

searches (Guha et al., 2003); b) ontologies must be available for the domain to test the algorithms. The second point is a serious constraint. In the end, we considered only 20 queries out of the 100 queries, or topics, from the TREC9 and TREC 2001 Web tracks.

The original TREC queries are described by: a) a title, which is the original user query extracted from users' logs, b) a description, which can be considered the NL interpretation of the query, and c) the narrative, which explains in more detail the relevant information that the user is looking for. We added, for the queries we used: d) a detailed request, suitable for a QA approach, e) notes on available ontologies covering that query. The complete list of our TREC topics selection is shown in Table 10.1, where each topic has associated a set of basic NL questions (obtained from the title, description and narrative provided by TREC), and ontologies that contains the answer.

Table 10.1. List of queries used in PowerAqua's second evaluation

Topic Number and Title: 451 - What is a Bengal cat?	
Description:	Provide information on the Bengal cat breed.
Narrative:	Item should include any information on the Bengal cat breed, including description, origin, characteristics, breeding program, names of breeders and catteries carrying bengals. References which discuss bengal clubs only are not relevant. Discussions of bengal tigers are not relevant.
Questions:	Provide information on the bengal cat breeders
Ontologies:	tapfull (animals)
Topic Number and Title: 452 – Do beavers live in salt water?	
Description:	Describe the normal habitat for beavers; note exceptions, if any.
Narrative:	Relevant documents describe the habitat range as well as references to specific areas and bodies of water.
Questions:	Describe the habitat for beavers.
Ontologies:	tapfull (animals)
Topic Number and Title: 454 – Parkinson's disease	
Description:	What are the symptoms and treatment of Parkinson's Disease, and what segments of the population have this disease?
Narrative:	Documents discussing research projects and funding for research projects were considered relevant only when clinical trials were included. Documents regarding legislation which discussed funding and programs were considered irrelevant.
Questions:	What are the symptoms of Parkinson?, What is the treatment for Parkinson?
Ontologies:	tapfull (diseases)
Topic Number and Title: 457 – Chevrolet trucks	
Description:	Find documents that address the types of Chevrolet trucks available.
Narrative:	Relevant documents must contain information such as: the length, weight, cargo size, wheelbase,

Questions:	horsepower, cost, etc. Find chevroleets.
Ontologies:	tapfull, autos (autos) <i>[both ontologies contained similar answers]</i>
Topic Number and Title: 465 – Deer	
Description:	What kinds of diseases can infect humans due to contact with deer or consumption of deer meat?
Narrative:	Documents explaining the transference of Lyme disease to humans from deer ticks are relevant.
Questions:	What deer diseases can infect humans? What human diseases are transferred by deers?
Ontologies:	tapfull (diseases)
Topic Number and Title: 467 – Dachshund dachshunds "wiener dog"	
Description:	Identify documents that contain information on buying and owning dachshund dogs.
Narrative:	Documents that discuss general dog information which is directly applicable to buying and owning dachshunds (i.e., how to chose a breeder) are relevant. Documents that list names of dachshund breeders and names of clubs for dachshund owners are relevant.
Questions:	Show me all information about dachshund dog breeders
Ontologies:	danchundogs, tapfull (animals) <i>[both ontologies contained similar answers]</i>
Topic Number and Title: 476 – Jennifer Aniston	
Description:	Find documents that identify movies and/or television programs that Jennifer Aniston has appeared in.
Narrative:	Relevant documents include movies and/or television programs that Jennifer Aniston has appeared in.
Questions:	Show me the movies of Jenifer Aniston.
Ontologies:	movie_database (cinema)
Topic Number and Title: 484 – Auto skoda	
Description:	Skoda is a heavy industrial complex in Czechoslovakia. Does it manufacture vehicles?
Narrative:	Relevant documents would include references to historic and contemporary automobile and truck production. Non-relevant documents would pertain to armament production.
Questions:	Show me the auto production of Skodas
Ontologies:	auto (autos)
Topic Number and Title: 489 – Calcium	
Description:	How do members of the medical profession view the effectiveness of calcium supplements?
Narrative:	Any document which cites the benefits of humans using calcium supplements or advises how calcium supplements should be used are relevant. A relevant document must establish that the information comes from a qualified medical source and not from the claims of a manufacturer or vendor of calcium supplements or from the opinion of anyone not recognized by the medical profession.
Questions:	What is the effectiveness of calcium supplements? What are the benefits of calcium?
Ontologies:	fungaiv2 (medicine)
Topic Number and Title: 491 – Japanese Wave	
Description:	Identify occurrences in which a Japanese wave or tsunami caused loss of life or damage.
Narrative:	Any reports that describe the occurrence of a Japanese wave or tsunami causing loss of life or damage are relevant. A relevant report must describe an actual event occurring at any location.
Questions:	Show me all tsunamis. Describe disasters produced by tsunamis.

Ontologies:	phenomenon (natural disasters)
Topic Number and Title: 494 – Nirvana	
Description:	Find information on members of the rock group Nirvana.
Narrative:	Descriptions of members' behavior at various concerts and their performing style is relevant. Information on who wrote certain songs or a band member's role in producing a song is relevant. Biographical information on members is also relevant.
Questions:	Show me all members of the rock group nirvana. What are the members of nirvana?
Ontologies:	tapfull, music (music)
Topic Number and Title: 504 – Information about what manatees eat	
Description:	Find documents that describe the diet of the manatee.
Narrative:	Relevant documents will identify any foods providing sustenance to the manatees.
Questions:	What is the diet of the manatee?
Ontologies:	tap (animals)
Topic Number and Title: 508 – Hair loss is a symptom of what diseases?	
Description:	Find diseases for which hair loss is a symptom.
Narrative:	A document is relevant if it positively connects the loss of head hair in humans with a specific disease. In this context, "thinning hair" and "hair loss" are synonymous. Loss of body and/or facial hair is irrelevant, as is hair loss caused by drug therapy.
Questions:	Of what diseases hair loss is a symptom Find diseases for which hair loss is a symptom. What diseases have symptoms of hair loss?
Ontologies:	biomedical(medicine)
Topic Number and Title: 511 – Diseases caused by smoking.	
Description:	What diseases does smoking cause?
Narrative:	A relevant document must describe smoking tobacco products as a cause of a disease. Diseases caused by second-hand smoke and smokeless tobacco are not relevant.
Questions:	What diseases does smoking cause? What diseases are caused by smoking?
Ontologies:	biomedical (medicine)
Topic Number and Title: 512 – How are tornadoes formed?	
Description:	How are tornadoes formed?
Narrative:	A relevant document will provide the meteorological and atmospheric conditions necessary to create a tornado and explain how the conditions interact to form the funnel-shaped cloud.
Questions:	How are tornadoes formed Describe the formation of tornadoes
Ontologies:	phenomenon (natural disasters)
Topic Number and Title: 513 – Earthquakes	
Description:	What causes earthquakes, and where do they occur most often?
Narrative:	A relevant document will discuss scientific causes of earthquakes or tremors and/or report geographic areas where earthquake activity occurs most frequently.
Questions:	What causes earthquakes? Where do earthquakes occur?

Ontologies:	phenomenon (natural disasters)
Topic Number and Title: 516 – Halloween?	
Description:	When, where, and how did Halloween evolve?
Narrative:	A relevant document will discuss the origin of Halloween and the original customs of Halloween. Modern day trick-or-treating stories are not relevant.
Questions:	What is the origin of halloween? What are the original customs of halloween?
Ontologies:	stconcepts (festivities)
Topic Number and Title: 519 – Info on where frogs live	
Description:	Find documents that describe the habitat of frogs.
Narrative:	A relevant document will identify the natural habitat of any type of frog. A frog's diet is not relevant.
Questions:	Where do frogs live? Describe the habitats for frogs.
Ontologies:	animals-wh (animals)
Topic Number and Title: 523 – Facts about the five main clouds	
Description:	How are the five main types of clouds formed?
Narrative:	A document that explains the process of cloud formation for any of the five main types of clouds is relevant. A document that discusses clouds, but does not explain their formation processes is not relevant.
Questions:	How are the clouds formed? Describe the formation of clouds. Explain the process of cloud formation
Ontologies:	phenomenon (natural world)
Topic Number and Title: 524 – How to erase a scar?	
Description:	What methods are used for removal of scar tissue?
Narrative:	A relevant document must disclose the name of a procedure or describe it, or identify the instrument used to remove scar tissue or skin defects. Mere references to "surgical removal" are insufficient.
Questions:	How to erase a scar? How to remove a scar?
Ontologies:	galen (medicine)
Topic Number and Title: 526 – bmi	
Description:	What does BMI stand for?
Narrative:	Any document that gives defines or explains BMI is relevant.
Questions:	What is BMI?
Ontologies:	form_demo (medicine)

10.5 Results

Table 10.2 contains the results of the performed evaluation. The first column contains the set of topics evaluated while the following columns contain the results for the five evaluation methodologies presented in Section 10.3, using two standard TREC metrics: average precision **and**

P@10 (precision at 10). The average precision metric gives an idea of the overall performance of the search engine, whereas the P@10 metric is restricted to the 10 first documents, giving an idea of how the search engine performs for what could be the “first page” of results. Numbers in bold correspond to maximal results for the current topic under the current metric, excluding the Best TREC manual approach, which outperforms the other approaches significantly for both metrics, most likely because the query is manually constructed introducing information from the title, the description and the narrative. The other three methodologies construct the query either by using just the title, in the case of the best TREC automatic approach, or by slightly modifying the title to fulfil its corresponding input format in the case of the ontology-based search engine. For this reason, we will exclude Best TREC manual from the rest of our analysis.

Table 10.2 Average Precision/P@10 metrics evaluation

Topic #	Lucene	Query expansion	Semantic retrieval	TREC automatic	TREC manual
451	0.2850/0.5	0.3970/0.7	0.4161/0.7	0.58/0.9	0.54/0.8
452	0.0292/ 0.5	0.0383/0.2	0.0383/0.2	0.2/0.3	0.33/0.9
454	0.2569/0.8	0.0281/0.4	0.2644/0.8	0.56/0.9	0.48/0.8
457	0.0000/0.0	0.0512/0.1	0.0486/0.1	0.12/0.3	0.22/0.8
465	0.0017/0.0	0.1414/0.2	0.1322/ 0.3	0/0	0.61/0.9
467	0.1241/0.4	0.1225/ 0.5	0.0984/0.4	0.09/0.3	0.21/0.8
476	0.2820/0.3	0.1954/0.2	0.1265/ 0.5	0.41/0.1	0.52/1
484	0.1230/ 0.3	0.1564/0.2	0.1916/0.2	0.05/0	0.36/0.3
489	0.1078/0.3	0.0346/0.1	0.0881/0.2	0.06/0.1	0.41/0.4
491	0.0794/0.3	0.4077/0.9	0.0770/0.2	0/0	0.7/0.9
494	0.2158/0.8	0.1427/0.2	0.4078/0.9	0.57/1	0.57/1
504	0.0755/0.2	0.1474/ 0.5	0.1349/0.2	0.38/0.5	0.64/1
508	0.0345/0.1	0.0732/0.4	0.1474/0.5	0.06/0.3	0.1/0.3
511	0.1543/0.5	0.3476/0.5	0.0733/0.4	0.23/ 0.7	0.15/0.2
512	0.1165/0.2	0.0640/0.1	0.2505/ 0.4	0.3/0.3	0.28/0.3
513	0.0602/ 0.4	0.0700/0.0	0.0786/0.1	0.12/0	0.11/0.4
516	0.0323/0.0	0.0755/0.4	0.0702/0.1	0.07/0	0.74/0.9
523	0.0000/0.0	0.2728/ 0.9	0.2860/0.9	0.23/0.4	0.29/0.9
524	0.0000/0.0	0.1853/0.4	0.1081/0.2	0.01/0	0.22/0.4
526	0.0596/0.0	0.1680/0.6	0.0863/0.1	0.07/0	0.2/0.5

Using as baseline the Lucene index, the results presented here show that using both the semantic query expansion and the semantic retrieval approach can improve the precision of the keyword-based approach for 85% of the evaluated queries⁶¹. In particular, for queries 457: “Chevrolet trucks”, 523: “How are clouds formed?”, and 524: “How to erase a scar?” the use of semantic results allowed valuable documents to be returned even if the keyword-search did not return any relevant result. Furthermore, for 80% of the queries the quality of the first 10 results is better when using semantic information than simple keyword ranking. Moreover, the semantic query expansion and the semantic retrieval approach together outperform TREC automatic in 55% of the queries for the average precision, and 70% for the P@10⁶². This is a strong indication that the semantic data provided by PowerAqua can help to enhance the quality of the results. For more details on how the document annotation process used for the semantic retrieval approach can be improved please refer to (Fernandez et al., 2010).

The performance of the semantic search system in terms of query response time, varied between a maximum time of 18.32 seconds for query 523: “How are clouds formed?”, and a minimum time of 1.63 seconds for query 491: “Japanese Wave”, thus resulting in an average of 5.37 seconds per query.

For the queries that have not outperformed the keyword baseline, such as query 467: “Show me all information about dachshund dog breeders”, a common reason is the scarceness of the semantic information obtained for the query even though, in general, such queries perform no worse than the keyword baseline. The exception is query 489: “What is the effectiveness of calcium supplements”. In this case, even though the semantic information retrieved by PowerAqua is relevant and focused on the benefits of calcium like: bone_strength, muscle_mass, etc., the precision of the semantic search is worse than the keyword search. The TREC evaluation states: “a relevant document must

⁶¹ There is not qualitative improvements of the semantic retrieval over the semantic query expansion

⁶² A future goal in our approach is to use better keyword-based index tools than Lucene to improve the results.

establish that the information comes from a qualified medical source”. Since our algorithms only focus on content and do not analyse the linking structures between Web pages to evaluate the quality of the source, they do less well than the baseline in this case.

Note that early attempts at building IR test collections exhaustively judged the relevance of every document to every query. However, for large collections and large numbers of queries (needed to achieve stable and statistically significant measures), providing complete relevance judgements is not feasible. A widely used alternative is *pooled assessment*, in which top-ranked documents from many systems are judged, and unjudged documents are treated as if they were not relevant. This may have adversely affected the quality of the results retrieved by ontology-based search, as more than half of the documents retrieved by the ontology-based search approach were not evaluated in the TREC collection. Therefore, our metrics marked them as irrelevant, when, in fact, some of them are relevant. In (Fernandez, M. et al., 2009) it is estimated that only 44% of the results returned by the ontology-based search approach had been previously evaluated in the TREC collection. The unjudged documents, 66% of the total, are therefore considered irrelevant. According to (Fernandez, M. et al., 2009) from the unevaluated documents 31.5% turn out to be relevant, and this figure increases to 50% if only the top 10 documents retrieved, which are not evaluated by TREC, are considered. This shows that, in most cases, the ontology-based search is obtaining new relevant documents when the query involves a class-instance relationship in the ontologies, such as specific symptoms and treatments of Parkinson disease, specific movies or TV programs, etc.

10.6 Discussion

We have constructed a complete semantic search approach that covers the entire IR process, from the NL query to the ranked set of documents. This approach has proved the feasibility of applying ontology-based retrieval models in unrestricted environments where an unlimited set of domains are covered. The initial results of the comparative evaluation are promising, showing that, when enough semantic information is available, the precision, and the average performance of the proposed

semantic search techniques only do worse than keyword search in rare cases and can often enhance the performance of current keyword search approaches.

During our work we noted the lack of standard evaluation benchmarks to formally judge the quality of ontology-based search approaches at a large scale. As discussed in Chapter 2, other ontology based QA systems, such as GINGSENG (Bernstein et al., 2006), NLP-Reduce (Kauffmann et al., 2007) and PANTO (Wang et al., 2007) make use of the independent Mooney dataset and queries in order to evaluate single ontology based interfaces. This is the first independent common dataset we are aware of to compare existing ontology based QA systems over a single source. Earlier systems, such as ORAKEL (Cimiano et al., 2007), AquaLog (Lopez et al., 2007) used their own dataset. The SEALS project is currently developing general evaluations for semantic technologies, we have presented in Chapter 9.4 a brief summary of PowerAqua results in the first 2010 SEALS evaluation campaign, which was based on the *Mooney* dataset and focused on usability. These are plausible evaluations if one considers SW applications based on using a single ontology at a time, whose role is to support the integration of all the available data, rather than trying to exploit the SW as a large-scale source of information.

In contrast with traditional IR approaches, which use the same type of inputs (keywords) and outputs (ranked documents) and are systematic, easily reproducible and scalable, there is no standard model of ontology-based search. As a result it is difficult to create independent sets of tests, questions and gold standards, in which semantic systems could be compared in terms of retrieving the best answer given a user information need. Aiming to address this issue, we built an ontology-based evaluation benchmark, departing from traditional IR datasets, to evaluate the multi-ontology capabilities of PowerAqua, used as a NL interface to a semantic IR system that retrieves documents as answers. Its practical advantage was that we were able to use an independent set of test questions and a gold standard from the IR community, the TREC WT10G collection, to judge the improvement in performance of the semantic search tool over a baseline. For data retrieval approaches like PowerAqua, which returns direct answers instead of documents, one way to

evaluate them is to consider their answers as a kind of query expansion. The expanded query is then used on the document space within a traditional keyword-based search. It can be argued that, apart from the limitations on the benchmark when applied to ontology-based search systems, the evaluation of data retrieval approaches using this methodology does not evaluate their real contribution, as we do not get direct results on their performance, but rather on how they perform to support another task. In this sense the comparison of ontology based search systems against QA models, such as those used in the IR QA track, could be more adequate to consider this type of query expressivity. For instance, the queries selected for TREC 9 and 2001 are extracted from real Web search engine logs, meaning that, the queries are generated in a suitable way for traditional keyword-based search engines and they do not exploit and evaluate the capabilities of ontology-based models to address more complex queries (in terms of relationships). Nevertheless, this is the first time an externally produced gold standard has been used to perform an evaluation over heterogeneous ontologies.

As stated in (Uren et al., 2010) “from an IR perspective this experiment may be seen as a ‘failure’ because our system could only cover 20% of the queries. However, in the context of the growth of SW, this experiment can be judged as a real milestone”. For the first time an evaluation has been performed over heterogeneous independent existent ontologies, while the queries and success criteria are externally sourced and use an externally produced gold standard.

10.7 Feasibility study: integration with Yahoo search engine

Our previous experiments highlighted the improvements derived from using PowerAqua as a query expansion component to progress beyond the capabilities of keyword-based IR search technologies. Therefore, it makes sense to explore whether the answers to a question returned by PowerAqua in the form of a list of entities can be augmented by open Web content retrieved by standard Web search engines, in response to queries automatically derived from the answers returned by PowerAqua.

Thus, the research question is: if the semantic data to answer a user query is available, can this semantic data dynamically obtained from distributed sources be used to expand the query and improve the precision of the documents retrieved when querying the Web?

To answer this question, we describe informal experiments we have carried out, to investigate whether it is feasible or not to use the answers obtained from user queries to PowerAqua, to successfully perform query expansion and improve the precision of Yahoo! web searches on the first 10 results. Here, we also investigate which ranking mechanism on the semantic results returned by PowerAqua is most efficient to automatically elicit the most accurate results on web searches.

We report promising results, even in the current semantic scenario, defined by large amounts of heterogeneous semantic data of varying quality. In addition, if semantic data is not available (knowledge incompleteness), the model can degrade gracefully to perform standard web searches using only the terms in the query.

10.7.1 Evaluation set up

We selected a sample of 20 queries, obtained from the PowerAqua website and the evaluation presented in Chapter 11⁶³, whose answers are encoded totally or partially in DBpedia and in a set of medium size ontologies used in the PowerAqua evaluations presented earlier (Section 9.2.1). PowerAqua's ranking criteria are used to sort and cluster the list of answers as detailed in Chapter 8. Answers cluster at position one (@1) represents the best subset of results according to the chosen ranking method and are the ones used to perform query expansion. In the evaluation presented in Chapter 9.3, the best ranking algorithm, based on the confidence on the quality of the ontological translation (q@1), was able to obtain an average of 96-99% of precision, depending on the type of fusion applied to the answers (union or intersection). The combined algorithm (c@1) has lower precision than q@1 (86-96%) but it is able to improve the precision and recall ratio. Here, we study

⁶³ The selected queries and results are in: <http://kmi.open.ac.uk/technologies/poweraqua/yahoo-evaluation.html>

which strategy, favouring precision or favouring recall, works better to filter the semantic data used for searching relevant documents on the Web.

We manually evaluated the effectiveness of the query expansion. The keywords of a user query were used to retrieve documents in Yahoo! and a set of judgments over the retrieved answers was performed to calculate precision@10. We compared these results with the ones obtained by PowerAqua after performing query expansion with the set of selected answers. An example query “List me all films with Brad Pitt and Angelina Jolie” is presented in Table 10.3. In this example Yahoo! retrieves many pages with news about the couple’s relationship (hot topic), like their appearance at the Cannes festival. In total 4 out of 10 (4/10) pages make a mention to their movies together. When extending the query with the semantic data (i.e. names of the films) the precision increases to 6/10.

Table 10.3. Example query for the evaluation task

<p><question> List me films with Brad Pitt and Angelina Jolie </question> <topic> Actors and films </topic> <keywords> films, Brad Pitt, Angelina Jolie </keywords> <evaluation_criteria> Pages about a film(s) by Angelina Jolie and Brad Pitt, or about one of the actors, explicitly mentioning film(s) were they appear together (as actors or producers), are valid. Pages about their relationship, without mentioning their movies are not relevant. </evaluation_criteria> <comments> Relationship is not specified. Brad Pitt is the producer of “A mighty heart”, starred by Angelina Jolie. Both are starring “Mr & Mrs Smith.”</p>
--

10.7.2 Results

The average results presented in Table 10.4 show that query expansion improves the quality of the keyword-based approach in more than half of the evaluated queries. This means that the semantic data can help to enhance the quality or precision of the results the user is likely to read (@10), in particular for queries that can be answered with a list (or by merging lists) of entities, such as “Which books Stephen King wrote?”.

All ranking algorithms outperformed the keyword baseline. Interestingly, the popularity algorithm p@1, which is the one with highest precision but lowest recall in the evaluation performed in Chapter 9.3, performed slightly better than the combined algorithm c@1. Thus, the best ranking algorithms are those favouring precision (the confidence/quality algorithm q@1) and reducing noise,

as good results can be still obtained after a loss in recall or in the presence of knowledge incompleteness (the set of answers is not complete). Only in one case, the precision of the semantic search is slightly worse than the keyword search, for the query “Which islands belong to Spain”, where articles about renting properties, ferries or offering sports in a given island were retrieved, rather than articles mentioning or listing the various islands in Spain. When no documents could be retrieved because the set of answers was too large, as in “Give me films about Buenos Aires”, the system degraded gracefully to use only the query terms. Nonetheless, even if there is no improvement over the keyword baseline the semantic sources provide a list of answers, as it is the case for “Give me all British actors in Titanic”, for which no relevant document could be found, before or after the query expansion. Note that PowerAqua can only answer factual queries, without comparatives or quantifications; therefore we could only evaluate the effectiveness of query expansion on those types of queries.

Table 10.4. Average evaluation results (precision@10)

P@10	Yahoo	q@1	p@1	s@1	c@1
Avg.(20)	38%	56.5%	54.5%	51.5%	54%

The preliminary results of this feasibility study on the first implemented prototype are promising. In particular, if enough semantic data is available, the answers retrieved from PowerAqua appear to improve the quality of the results returned by the Yahoo! search engine. The novelty of this approach is that the source of the semantic data (the SW) is independent from the document collection (the Web itself). Therefore, maintaining the two information sources decoupled, i.e., with no links or annotations between the semantic search space (the ontologies and KBs) and the unstructured search space (the documents on the Web). The quality of the final search depends both on the quality of the ontologies and the quality of the query expansion terms discovered by PowerAqua.

10.8 Conclusions

With respect to the output, PowerAqua's answers, being primarily built for semantic QA, are constructed directly from structured data. The user-centred studies, presented in Chapter 9, focused on assessing firstly PowerAqua's ability to answer questions by identifying relevant information spread in multiple ontologies, and secondly, the degree of user satisfaction with respect to the range of questions the system should be able to answer. In addition to discovering structured answers, in this chapter, we have investigated the value of PowerAqua as a practical way to meaningfully enhance keyword search technologies on the Web by providing the semantic context to meaningfully extend an IR query. This is being done by: i) using PowerAqua as the query expansion component for an existing IR tool able to find documents semantically annotated with PowerAqua answers, and ii) using PowerAqua results to automatically trigger contextualized searches in standard search engines, such as Yahoo!.

Therefore, by integrating our QA approach with traditional IR under a common model, the ontological answers can be complemented with unstructured documents. The queries that give particularly good results, improving the precision of keyword-based techniques, are those that are translated into a class-instance relationship, where a list of individuals or instances is returned as an answer. Consequently we are bridging the gap between semantic search technologies that execute the user query on a KB, returning tuples of ontology values which satisfy the information request, and the IR document search space or the unstructured textual information available on the Web.

Several issues remain open nonetheless. One of the distinctive features of our system is its openness to the number of topic domains. Indeed, our system can potentially cover an unlimited set of domains by making use of ontologies provided in the SW scenario. Our experimental evaluation has shown however that the potential of our system is overshadowed by the sparseness of the knowledge currently available on the SW. Indeed, by the time of this study, although the SW has expanded dramatically, only about 20% of the queries (topics) in the TREC dataset were covered to

some extent by online ontologies. Further, these ontologies were only populated at the class level or weakly populated with instance data, so that we had to include in our experimental setup a semi-automatic population step from Wikipedia.

Nevertheless, while this status of the SW caused a suboptimal behaviour for our system, any extension of the critical mass in ontologies and semantic data available online will result in a direct performance improvement of the proposed approach. Meanwhile, a potential way to cope with semantic incompleteness and sparseness is by means of graceful degradation to a classic IR system, which gets by without semantics when they are insufficient.

Despite the limitations the results from these evaluations are promising. Hence, we regard these findings encouraging to further developing PowerAqua as part of a search engine integration, and as part of an important challenge for future research work: the integration of semantic and non-semantic data.

Chapter 11 Scaling up to Linked Data: issues, experiments and lessons learned

The solutions and experiments on scaling up Question Answering to Linked Data presented in this chapter have been published in the Knowledge Engineering and Knowledge Management Conference in 2010 (Lopez et al., 2010).

Furthermore, with regards to scalability, PowerAqua has participated in the billion triple challenge hold in the International Semantic Web Conference in 2008 (d'Aquin et al., 2008b)

11.1 Motivations

One of the main issues for PowerAqua is to maintain real time performance in a scenario of perpetual change and growth. The SW has expanded rapidly, a prominent example is the amount of Web data in the Linked Data cloud (Bizer et al., 2009a). Till recently, most KBs covered specific domains and were created by relatively small groups. Yet, this is starting to change and we are reaching the critical mass required to realise the vision of large scale, distributed SW, with real-world datasets, representing real community agreement. For example, the DBpedia project (Bizer et al., 2009b) is increasingly becoming the central interlinking hub for the emerging Linked Data.

Ultimately, by integrating and connecting data on the Web, Linked Data “can be used to answer quite surprising queries about a wide range of topics” (Kobilarov et al., 2009), as we hypothesized in our work. PowerAqua’s outputs are answers constructed directly from the open and freely available structured semantic data. Hence, it is worth investigating whether it can be successfully used to query Linked Data content on the SW.

However, while Linked Data datasets literally may contain the answers to millions of questions, locating and exploiting the relevant information to extract these answers from them is a major challenge. Back-end technologies and semantic applications that can be robust at small or medium scale, may not be suitable when applying them to a real-world scale of heterogeneous Web data. In fact, most Linked Data end-user tools analysed in Section 11.2 only perform a shallow exploitation

of these data. On the contrary, as we have seen throughout this thesis PowerAqua is able to integrate, on the fly, statements drawn from different heterogeneous semantic sources to generate integrated answers to user questions. Thus, in this chapter we analyse the feasibility of this ambition, by looking at the scalability issues which arise when integrating PowerAqua with the largest general purpose dataset offered by the Linked Data community: DBpedia.

In particular, querying Linked Data brings up a new scenario, whose differentiating characteristics (further detailed in Section 11.3) are:

- **Scalability.** As a result of the LOD initiative, scale is not only related to the number of ontologies on the SW, but also to their size. These large datasets can potentially cover a wide range of user queries, thus making it more difficult for PowerAqua to focus quickly on a few ontologies with high discriminatory power.
- **Increased Heterogeneity.** The LOD initiative has also caused a shift from the exploitation of small domain ontologies to the exploitation of large generic ontologies covering a variety of domains. As a result, heterogeneity is not only arising from the use of different ontologies, but also within the same ontology. As argued in (Mollá, and Vicedo, 2007) ontology-based QA systems in restricted domains can tackle the answer-retrieval problem by means of an internal unambiguous knowledge representation. However, in open-domain scenarios, or when using open-domain large ontologies, as is the case of DBpedia, systems face the problem of polysemous words and concept ambiguity (one query term can have multiple interpretations within the same ontology), which are usually unambiguous in restricted domains.
- **Dealing with noisy and incomplete data.** Ontologies are decentralized, containing heterogeneous terminology and modelling errors: datasets often lack semantics or a formal ontology, e.g., missing domain and range information for properties, undefined entity types,

complex semantic entity labels, redundant entities within the same dataset (e.g., birthplace and placeofbirth), etc.

In what follows, we look at the abilities of existing tools that handle the sheer amount of multi-domain data offered by the SW and Linked Data to provide easy access to the end user (Section 11.2). Then we present the major issues (Section 11.3) that we faced to scale up PowerAqua to take advantage of Linked Data's potential to answer queries. The feasibility of the solutions presented (in Section 11.4) to scale up to this new semantic information space is assessed through experiments. These experiments, based on the same evaluation set up presented in Chapter 9, measure the QA performance before and after using DBpedia (Section 11.5). In addition, we present the latest PowerAqua evaluation, focused on assessing the performance of its algorithms using different semantic storage platforms, in particular Virtuoso and Watson (Section 11.6).

Our aim is to analyse the main issues that currently overshadow Linked Data potential in the QA context and to present the lessons learned, gained through the experience of adapting the PowerAqua NL interface, to scale to some of the available Linked Data. We believe that the learned lessons obtained with our experiments can be extrapolated to a large proportion of semantic tools that wish to retrieve, use and combine these large, multi-domain semantic data on the fly.

11.2 Current interfaces to Linked Data and limitations

The database and SW communities had developed back-end technologies for managing large amounts of Web data. Various RDF stores can scale to large amounts of data originating from different sources, such as Virtuoso or the Talis platform⁶⁴. Search engines such as Watson (d'Aquin et al., 2007) and Sindice (Oren et al., 2008) come also with features for indexing and querying data from the SW.

⁶⁴ Open Link Virtuoso: <http://virtuoso.openlinksw.com> and Talis platform: <http://www.talis.com/platform/>

Linked Data sources usually offer a SPARQL endpoint for their dataset(s)⁶⁵. Alternatively, they also provide RDF data dumps to build and query your own store⁶⁶. However, users can hardly know which identifiers and properties are used in the KBs and hence can be used for querying. Consequently, they have to be guided when building queries, e.g., through the suggestion of reasonable alternatives. Creating innovative ways to interact with Linked Data and the SW data is crucial and even envisioned as a potential “killer app”.

Nonetheless, to find a trade-off between the complexity of the querying process and the amount of data ontology-based approaches can use and integrate is still an open problem. Semantic search models that have proved to work well in specific domains still have to undertake further steps towards an effective deployment on a decentralized, heterogeneous and massive repository of content about a potentially unlimited number of domains. Here we present a state of the art on user interfaces that can, in principle, scale sufficiently to explore the large amounts of semantic data, such as those available as Linked Data:

Triple query builder interfaces. A Query Builder allows users to query the KB by means of multiple triple patterns (Auer and Lehmann, 2007). For each triple pattern variable, identifiers or filters for the subject, predicate and object can be defined. The user needs to follow the terminology and structure of the ontology to pose queries, e.g., the DBpedia Leipzig query builder⁶⁷. However, for each typed identifier name a look ahead search proposes suitable options in a drop down menu that helps the user to create complex queries, e.g.: `<?x, rdf:type, db-ont:Person> <?x, notablePrize, Nobel_Peace_Prize>`.

⁶⁵ A more complete list of SPARQL Endpoints at: <http://esw.w3.org/topic/SparqlEndpoints>

⁶⁶ Jena <http://jena.hpl.hp.com/wiki/TDB>; Sesame <http://www.openrdf.org>; 4store <http://4store.org>;

⁶⁷ <http://querybuilder.dbpedia.org/>

Relationship finder. The DBpedia relationship finder (Lehmann et al., 2007) explores connections between objects. DBpedia is treated as an undirected graph and, given two objects, the relationship finder looks for a path between them (not necessarily the shortest).

Keyword lookup: The DBpedia URI Lookup Index and OpenLink Software⁶⁸ find the most likely matches (URIs) for a given term. The service combines Lucene's string similarity based ranking with a relevance ranking similar to PageRank. The OpenLink Software builds a text, label and URI lookup service upon a larger collection of sources, but it limits the number of results to only those containing an exact mapping of the input keyword and the search can be refined by specifying URIs of classes, properties or values. However, expressivity and usability are limited (e.g., 1358 classes are associated to the keyword "actor").

Data aggregators and mash-ups. In Sig.ma (<http://sig.ma/>) the user enters a keyword and is able to explore all the aggregated data coming from the search engine Sindice (including synonyms). Although mash-up technologies provide support for large-scale indexing and for aggregating heterogeneous information, they do not attempt to automatically disambiguate or rank between different interpretations, even though, in Sig.ma the user can filter out the irrelevant sources.

Linked Data browsers. These provide a way to browse RDF data on the Web. Examples are Tabulator (Berners-Lee, 2006) and Disco⁶⁹. Given a dereferenceable HTTP URI, these browsers render all information that they can find about that URI. All aggregated data across sources is viewed in a tabular form and the user can navigate through interlinked sources. However, it does not aggregate unconnected entities with different URIs representing the same individual.

DBpedia Faceted search. This allows the user to easily ask queries like "recent films about Buenos Aires", by typing the keyword "Buenos Aires" and then applying an intelligent filtering on

⁶⁸ <http://lookup.dbpedia.org/> and <http://lod.openlinksw.com>

⁶⁹ <http://sites.wiwiiss.fu-berlin.de/suhl/bizer/ng4j/disco/>

the underlying ontology, in this case by selecting the item type “Film”⁷⁰. The interface guides the user to filter objects according to facets (properties) and range of values. Nevertheless, the user needs to familiarize herself to some extent with the vocabulary and the structure of the KB - e.g., lexically related words like “Movie” are not understood. This applies not only to undefined terms, but also when there is not a straight mapping between the user query and the way the knowledge is structured in the ontology. For example, the query “Give me the husbands of Elizabeth Taylor” cannot be easily formulated if the user does not know that the relevant relation “spouse” is defined only for the entities representing Elizabeth Taylor’s husbands, whose ontological type is “Actor”, and it is neither defined for the instance “Elizabeth Taylor”, nor for the types “Person” or “Artist”. As the system is unable to map “husband” to the relation “spouse”, the only solution left is to manually explore among all the results that contain a match for “Elizabeth Taylor”. However, filters may impose the limitation of searching within a particular type. For instance, in the DBpedia faceted search, if the user wants an answer to a query like “Give me actors that appear in films directed by Francis Ford Coppola” the user first needs to get all films directed by Francis Ford Coppola, and then search the actors for each film.

Research on user interfaces for Linked Data is a open-ended area and the above paradigms provide different solutions to handling the sheer amount of multi-domain data. We distinguish the following limitations according to the criteria presented in Table 11.1:

- **Usability:** if the input of the system is a URI or is some of the components of a triple, usability is limited to knowledgeable users, familiar with semantic source’s contents.
- **Expressivity:** if the system builds upon keywords, then it provides limited capabilities to grasp and exploit the conceptualizations involved in user needs, with limitations such as the inability to account for relations between terms, or to cope with complex queries.

⁷⁰ <http://dbpedia.neofonie.de/browse>

- **Scalability:** if the system is restricted to one or a set of domains to maintain performance then it cannot scale up to a truly open environment.
- **Mapping:** the vocabulary that the system can understand is limited to that used in the ontology, the input is controlled and if a term has more than one sense, disambiguation is done manually by the user. Although guided interfaces solve the old habitability problem (a mismatch between the user expectations and the abilities of the system), the burden to formulate the queries is shifted from the system to the user.
- **Fusion:** most systems have limited ability to merge heterogeneous facts or multiple different answers (representing the same individual) across different semantic sources.
- **Ranking:** most systems lack ranking algorithms to cope with large-scale information sources.

Table 11.1. Limitations of the existent paradigms according to 6 criteria.

Criteria	Query Builder	Rel. Finder	Term Lookup	Mash-ups	Browsers	Facets
Usability	Yes	Yes	Yes	Yes	--	Yes
Expressivity	Yes	--	No	No	No	Yes
Scalability	Yes	Yes	Yes	Yes	Yes	Yes
Disambiguation	No	No	--	--	No	No
Fusion	No	No	No	--	--	No
Ranking	No	No	--	No	No	No

Among the existing Linked Data querying approaches presented, only facets and query builder interfaces provide an efficient way to pose complex and expressive queries to a large repository. These allow users to query the KB by mean of multiple triple patterns, either supporting the definition of variables for the elements of the triple (query builders) or a filtering process based on the type and properties of the underlying ontology (facets). In fact, a common drawback of these systems is that the user, not the application, is the one who has the responsibility to reformulate the query in a way that can be understood, following the terminology and structure of the ontology.

Another open issue concerns the usability of menu views and facets over multiple heterogeneous sources. Here the varying levels of usability depend on the query complexity and the number of filtered options presented on the drop menus, as end-users can get lost in large-scale information spaces.

Furthermore, only keyword-based mash-ups (and lookup services to some extent) can aggregate information across sources. However, they do not attempt to fuse similar results, nor do they have enough context to interpret and elicit the best answers. What is achievable on small/medium scale data by querying interfaces (in particular sophisticated NL ones) has until now not been achieved on large Linked Data datasets. In this chapter, we aim to go beyond the state of the art, by adapting the mapping and fusion techniques required by PowerAqua’s NL interface to the Linked Data scenario.

11.3 Before and After Linked Data: a new dimension in QA

In this section we discuss three crucial challenges that Linked Data pose to systems such as PowerAqua.

11.3.1 Scaling to highly populated and dense ontologies

A well-known limitation of the SW when compared to the Web is its sparseness (Polleres et al., 2010), not only on the reduced number of topics covered by existing ontologies (domain sparseness) but also at the level of instances and relations, as pointed out in (Fernandez et al., 2008) (d’Aquin et al., 2007). As stated in (Uren et al., 2009), without a well populated SW, developing semantic search systems is only an intellectual exercise. However, Linked Data initiatives are producing a critical mass of semantic data, and it is likely that soon we will have so much data that the core issues would be less related to sparseness than to scalability and robustness.

Furthermore, as reported in an analysis of 25,500 ontologies and semantic documents collected by Watson (d’Aquin et al., 2007) in 2007: “the SW is characterized by a large number of small, lightweight ontologies and a small number of large-scale, heavyweight ontologies”, the biggest one

at that time containing more than 28,000 entities. By contrast, an obvious characteristic of Linked Data is that it is very large, DBpedia alone consists of more than 103 million RDF triples, and describes more than 2.9 million entities⁷¹. Also, as analysed in (Lehmann et al., 2007) the DBpedia dataset is densely connected. As ever, the links between entities are more important than the entities themselves because that's where the context lives.

In particular, the time needed to answer a query by PowerAqua depends on: (1) the number of required (SPARQL-like) calls to the Watson API or to online repositories to explore relevant connections between entities, and (2) the response times to these calls. The total number of calls depends directly on the number of semantic sources and mappings that take part in the answering process, e.g., too many irrelevant ontological mappings collected by PowerMap inevitably affect the overall performance of the system. The response times to these calls depend directly on the complexity of the (SPARQL-like) query and the size of the ontology, in particular when a query involves one or more classes with lots of instantiated data.

Thus, scale matters since the response time of the calls to query the ontologies depends directly on the number of potential mappings and the size of the ontology. Therefore, the algorithms for querying Linked Data need to be able to explore the relevant connections while trying to avoid expensive computations.

11.3.2 Mapping query terms to large generic ontologies

The really challenging aspects of these Linked Datasets appears to be not only their scale but also their heterogeneity: new challenges are introduced by simultaneously querying not only a large number of domain specific ontologies but also a few very large, populated ontologies covering several domains.

⁷¹ As November 2009

The first version of PowerAqua was developed to work on an initial, sparsely populated SW. Therefore, PowerAqua's algorithms were designed to maximize recall in order to bridge the gap between user terminology and the terms used in the ontologies. Nevertheless, to keep up with the continuous and rapid growth of the SW this approach is no longer effective, and heuristics that balance precision and recall are now used to be able to prune the search. To avoid analysing an unfeasibly large space of solutions, the algorithms are iterative, and try to find an answer by augmenting the search space at each iteration, until either an answer is found or all possibilities have been analysed. The algorithm is based on the assumption that the ontologies that better cover a query (i.e., contain matches for most of the elements on a QT) are likely to represent better the domain(s) of the query and contain an answer. The algorithm uses this coverage criterion to restrict the mappings to be analysed to those in the covering ontologies, extending the search space only if no answers are found.

However, the main distinctive feature of DBpedia, apart from its size, is that it is a repository about a wide range of domains. Therefore, even in the cases where the answer to the user query is not contained in DBpedia, this dataset is frequently selected as relevant, and often contains a huge number of potential ontological hits, from a large number of domains, for one or more of the terms in the user query. Consequently, the coverage criterion previously used to filter ontologies, ergo candidate mappings, becomes insufficient when most of the lexical matches for a user query belong to just one large open domain KB and often have different meanings than the one intended by the user. For instance, DBpedia alone contains more than 1000 mappings for an apparently unambiguous keyword like "Russian" - e.g., the exact mappings *Russia* (instance) & *russia* (property); the synonyms *Soviet_Union* (instance) & *USSR* (instance); the hypernym *country* (class); the approximate instances: *Russian_empire*, *president_of_russia*, *MTV_Russia*, *Rocket_to_Russia*, *Russia_Today*, *Anastasia_of_Russia*, etc. Analysing the ontological context of all potentially relevant hits, to select the ones containing an answer, would result in unacceptably slow response time for a run time algorithm. Therefore, new filtering heuristics are needed.

11.3.3 Fusion across heterogeneous and decentralize ontologies

When searching multiple collections together, knowledge needs to be shared and reused through fusion techniques. Knowledge can be aggregated to complete information partially presented in single sources, fusing redundant or similar answers from different sources together and filtering irrelevant ones, as described in Chapter 8.

Fusion requires matching at the schema level as well as entity reconciliation at the data level; it assigns the individuals returned as answers from different ontologies into subsets of answers that represent identical entities. A decision about the equivalence of two answers is made based on string similarity metrics applied to their labels, local names, and, in case of uncertainty, other datatype attributes. As explained in Chapter 8, pairwise comparison of entities would make the complexity of the procedure N^2 with respect to the input set size. In order to avoid this, candidate matches are selected using a search over the indexes and the comparison focuses only on the entities that appear among the search results. This makes the complexity linear with respect to the answer set size, but in cases where the answer set is formed by thousands of partial answers, further heuristics to improve efficiency are needed.

11.4 Solutions to challenges and lessons learned

Next, we report on the solutions adopted by PowerAqua to solve the challenges outlined above. Our experiences also give us an insight into the quality of the datasets and a better understanding of the concrete issues encountered when handling large-scale data.

11.4.1 Large scale data: Shifting focus onto precision for mapping and fusion

DBpedia contains a large number of instances across domains, and as said before, it can produce a large number of diverse mappings for a single query term. Strategies to select the most likely mappings to answer a query and filter the least promising ones are crucial to ensure an acceptable run time performance when querying a huge amount of semantic data.

These strategies start with a quick filtering mechanism based on scores, returned by Lucene **string similarities**, to ensure that the mappings for a given term, within one single repository, exhibit a minimum degree of quality and their number does not exceed a given threshold. This favours precision and can negatively affect recall; however, it is a necessary measure to ensure that all questions can be answered in real time. Secondly, the number of mappings is reduced according to **heuristics to filter out** the least promising individual mappings. These heuristics are not based on a PageRank-like algorithm (i.e., the popularity of the entity within the ontology)⁷² but rather on the context of the query. The reason behind this is that we do not want to penalize searches in which the user is interested in the unique meaning (not the popular meaning) of the word. For instance, the different meanings of the word “Turkey” are clear when asking for “Give me books written in Turkey” or “Which wine goes well with a braised Turkey?”. Based on the **coverage criterion** ontologies about geography or food would be selected in the two examples. However, an ontology like DBpedia contains both meanings of Turkey (*Turkey_(bird)* and *Turkey*) plus several mappings for “books” and “wine”. Before the time consuming process of analysing the relationships between the mappings to obtain the interpretation that better translates the query in a given ontology, further heuristics to reduce the number of mappings within an ontology are applied. These heuristics are **based on the syntactic relevance** of the mappings: (1) the quality: exact vs. approximate; (2) the semantic relation: equivalent, synonyms, hyper(hypo)nymy, or meronymy. Next, more expensive semantic mechanisms based on the **ontology taxonomy** are applied, i.e. to discard redundant mappings by selecting those that are higher in the same ontology hierarchy (e.g., the class “wine” selected over its subclasses “rose-wine”, “white-wine”, “sugary-wine”, etc). Also, **unconnected mappings** with no ontological context (isolated entities), in the form of domain or is-a relations, are discarded.

⁷² Popularity measures are used only to rank the final answers obtained across ontologies.

These strategies applied by PowerMap's mapping algorithms to increase precision, and consequently performance, require making certain assumptions about the quality of the semantic sources, without making any unreasonable or a-priori assumption. As explained in Section 11.4.2, if the heuristics are too strict, recall is affected and valid answers are missed, in particular for the heterogeneous and general datasets such as DBpedia. The mapping process finishes with the Triple Similarity Service iterative algorithms (detailed in Chapter 7) that exploit the **ontological relationships** between the candidate entities to instantiate the user query (starting with the most straightforward solutions and executing expensive steps last, if no solution is found).

In order to improve the efficiency of the fusion procedure, our approach is to balance the quality of the resulting set of the answers and their expected duration. When the number of answers, which have to be processed by the fusion module, is large, our procedure tries to minimize the number of index search calls. The input of the fusion procedure is a set of answers retrieved from one or more ontologies $\mathcal{O}_i, i=1\dots n$. Let A_i denote a set of answers $\{a_{i1}\dots a_{im}\}$ coming from the ontology \mathcal{O}_i . By default, for each a_{ij} we searched the index using as keywords the label and local name of a_{ij} together with their synonyms extracted from WordNet. Instances a_{kx} which were retrieved by the search, and at the same time belonged to the original set of query answers, were considered as candidates for fusion. In the new version of the algorithm, we introduced two thresholds for $|A_i|$. The first regulates the use of WordNet synonyms in search: if $|A_i| > \lambda$, then WordNet synonyms are not used when searching for instances similar to $a_{ij} \in A_i$. The second one excludes the whole set A_i from search: if $|A_i| > \mu$, then the module does not try to search for instances similar to any $a_{ij} \in A_i$. Thus, if there is an instance a_{kx} belonging to an ontology \mathcal{O}_k such that $a_{kx} \equiv a_{ij}$, they can only be merged if a search for a_{kx} returns a_{ij} , which potentially leads to lower recall of the fusion procedure. However, this potential loss of recall is justified when we are dealing with very large answer sets.

11.4.2 Higher heterogeneity and duplicated terms: filtering heuristics based on quality and semantics

In DBpedia, the most valuable contents extracted from Wikipedia are the *infoboxes*. However, different infobox templates use different names for the same attribute (e.g., *birthplace* and *placeofbirth*). The DBpedia project deals with this situation by using two different extraction approaches in parallel: a *generic* one that aims at a high coverage, and a *mapping-based* one that aims at high data quality by mapping Wikipedia terms to a manually created ontology (Bizer et al., 2009b)⁷³.

One of the filtering heuristics to favour precision is to consider the quality and semantic relation of the mappings. However, the presence of duplicated entities within the same semantic source limits the effectiveness of this criterion. Consider the example: “Give me the husbands of Elizabeth Taylor”, the keyword “husbands” produce several mappings in DBpedia, including: the approximate equivalent instances: *Clifford_Husbands*, *Commuter_Husbands*, *Dead_Husbands*, *Husbands_and_Wives*, *Young_Husbands*, etc; the exact equivalent properties: *husbands*, *husband*, etc; and the exact hypernyms: *spouse*, *spouses*, *partner*, etc., some of them with the same semantics (duplicated entities) although only very few are connected to the relevant information. The equivalent properties representing “husband(s)” do not produce any valid Onto-Triples (OTs) with the entity “Elizabeth Taylor”. The answer is encoded in the hypernym property “spouse”. Therefore, hypernyms can be as relevant as equivalent mappings. With this kind of dataset, in which different terms have the same semantics, heuristics need to be flexible.

First, within an ontology, exact mappings, if any, are selected over approximate mappings for the same query term and entity type, e.g., the exact class *Film* is preferred over the class *FilmFestival* as a matching for the synonym term *Film*. Furthermore, exact mappings are a requirement in cases

⁷³ The DBpedia ontology contains 170 classes, which form a shallow subsumption hierarchy, and more than 900 properties. Nevertheless, there is also a dataset of approximately 8000 different property types for which there is no formal ontology.

where the type of the mapping is not the expected one. For instance, in “Who won a Nobel prize” the linguistic relation “won” is mapped to the instance “Won_James_Won”, this approximate mapping is discarded before analysing how it relates to the subject and object of the triple.

Ultimately, the semantic validity of the candidate mappings is established by analysing the ontological context, which is translated in many SPARQL-like queries to find the OTs. However, for queries that are particularly expensive, heuristics based on the semantic relation (synonym, hyper(hypo)nym) together with the quality of the mapping (exact, approximate) are also applied to minimize the number of those expensive queries. In large ontologies like DBpedia, searching for both *domain relationships* (1:1) between two highly populated classes and *indirect relationships* with one mediating concept (1:2) are expensive computations. In these two cases, to favour precision when looking for relationships among the mappings, only pairs in which at least one candidate mapping is exact and equivalent are analysed (singulars and plurals are considered exact mappings). Hence, for the query “Give me actors starring in movies directed by Clint Eastwood”, the DBpedia mapping for the term *actor* is selected over the DBpedia mapping obtained with its hypernym *person* to look for relations between *actor* (exact) and *movie* (synonym).

Finally, heterogeneity is not only present in the vocabulary but also in the granularity of the data (i.e., entities modelled with different degrees of richness). For instance, the depth that characterizes the YAGO hierarchy (Suchanek et al., 2008) and its conjunctive schema classes (used in DBpedia classification), which encode too much information in one class, make the processing of the labels too difficult for automatic QA understanding, as in the case:

“MultinationalCompaniesHeadquarteredInTheNetherlands”.

11.4.3 Lack of semantics and incomplete data: light-weight reasoning

PowerAqua is able to scale thanks to the various strategies and filtering heuristics that keep the number of mappings and queries to the semantic sources more or less constant, even when adding large semantic sources or a large number of them. However, as said in Section 11.3.1, performance

also depends on the size of the ontologies, which influences the response time of the calls (SPARQL or SeRQL) to query them. The effectiveness of these queries, which use the ontology semantics to perform basic light-weight inferences based on the taxonomy and relationships, also relates to the quality of the sources they are querying.

In DBpedia, the properties defined in the namespace <http://dbpedia.org/ontology/> (*dbpedia-owl*) belong to the data generated by DBpedia *mapping-based* approach. In this approach, fine-grained rules are applied to define the target datatype and ignore additional text that may be present in the attribute value. In the DBpedia *generic approach*, <http://dbpedia.org/property/> (*dbpprop*), the coverage of all infoboxes is complete but synonymous attribute names are not resolved, and there is a high error rate to determine the datatype of a value. Although the percentage of properties pointing to other DBpedia entities is much higher in the mapping-based dataset (53%) than in the generic dataset (25.6%) (Bizer et al, 2009b), the coverage (in terms of instances) is lower (843,000 compared to 1,462,000 entities). The effectiveness of finding answers in the generic approach is limited with respect to the mapping approach and it has an important impact on query performance, as described next.

a) Domain and range: In the DBpedia mapping approach OTs can be extracted with reasonable performance by querying the schema (domain and range). However the answer can be encoded in a property defined within the generic approach, where properties do not map to a schema. The lack of domain and range information results in either sending expensive (time consuming) SPARQL/SeRQL queries or missing connections between the analysed entities. To balance performance and recall, domain and range information is crucial in cases where we are looking for indirect relationships, or domain relations between two classes (or a class and a literal). For example in “Give me languages used in Islamic countries?” the query is translated into OTs representing “languages spoken in a country” and “countries that are Islamic”. Domain and range information is used to get all possible relations among the two classes “language” and “country”, because it is not feasible to check all the instances of languages to find domain relations with any instance of country

in real time, particularly for highly populated classes. Therefore, the answers encoded in the OTs formed with the KB relation “<http://dbpedia.org/property/states>” are not found; however PowerAqua finds answers for the schema relation: `<language, http://dbpedia.org/ontology/states, country>`. Furthermore, domain and range information is also needed in order to complete a triple when the relation is mapped but the subject of the triple is not. For instance, the query “In which region is Cantonese spoken?” is mapped to the OT: `<PopulatedPlace (domain of region), region, Cantonese>` with 12 answers (Hong Kong, Macau, etc.). Because an instance can instantiate a relation with thousands of other instances, schema information is needed to model the OT.

b) Inference of relationships: When looking for relationships between two entities, the ontology is treated as an indirect graph, and all direct and inverse relations are retrieved. Inheritance of relations is also considered. If the ontological platform does not offer schema inferencing (as it is the case for Postgres repositories in Sesame 2), then complex *SeRQL* queries need to be generated to consider the relationships defined for the superclasses of the classes involved. In our previous example, domain and range information is used to find instantiated relations between the classes “language” and “country”, or any of their superclasses. If looking for indirect relations, inheritance of relations is also considered and the search is restricted to candidate mediated concepts with relations defined in the schema. Looking for indirect inverse relations is avoided as it is computationally expensive to search for relations to and from highly populated mediated concepts. Also, inferences cannot be carried out with instances whose type is defined in another ontology.

c) Literals and meaningful labels: a literal has no structure and the meaning is given just by its label. For instance, the unprocessed value for the property “dbpprop:states” in the instance “Tamil language” (“*India, Sri Lanka and Singapore, where it has an official status; with significant minorities in Canada, {...}*”) is too complex to automatically infer answers by using NLP techniques - e.g., to obtain all official languages spoken in India.

11.5 Experiments with DBpedia and Discussion

Despite using community driven large scale knowledge obtained from sources that are heterogeneous, redundant and not always complete or well formed, the SW technology is mature enough to interpret and answer NL user queries. In this section, we present some example queries to justify our claim that we can obtain answers to queries directly from DBpedia – even in its current form. The solutions in Section 11.4 allowed us to improve the mapping and fusion algorithms used by PowerAqua to exhibit better performance. This is achieved by shifting the focus on precision while minimizing the loss in recall. We have made an initial comparative study between PowerAqua’s efficiency before and after adding DBpedia dataset and the heuristics to favour precision, by using the semantic data and a subset of queries from the previous evaluations presented in Chapter 9. This semantic data consists of 700 semantic documents distributed in 130 repositories, 3GBs data in which the biggest source, SWETO, is no more than 1GB (over 3 million triples, around 800,000 entities and 1,600,000 relations). The only change was the addition of more than 13 GBs of semantic data from DBpedia (in a Sesame repository) to the semantic search space, as a representative LOD dataset. A sample of 16 queries can be seen in Table 11.2, where the last 6 queries can only be answered if DBpedia is included in the query dataset. As shown in Table 11.2, the average time for the mapping algorithms to translate a query increases from 32 to 48 seconds when using the same queries and datasets but adding DBpedia, even with the use of filtering heuristics. However, the resulting number of valid answers obtained after applying the fusion algorithm (which has a precision of 94% according to the evaluation presented in Chapter 9.3) rises from 64 to 370 on average. The average mapping time with DBpedia increases to 52 seconds when adding complex queries like Q_{11} and Q_{12} that require fusing partial translations from DBpedia and

other datasets. While this is acceptable for a research demo, work still has to be done to improve the speed. In any case, it shows that semantic data can be handled in modest projects⁷⁴.

The reasons behind the decrease in speed are not so much because of the increase of the number of resultant hits obtained when querying more and larger repositories. Heuristics that balance precision and recall reduce *SeRQL* calls by more than 40% (352/587) and even keep them lower (540) when mapping complex queries into multiple facts. However, speed falls because of a suboptimal performance at the back-end, where the response times to calls to the repositories increase for single large datasets. In particular this is the case for expensive queries to find: (1) relationships between instances of highly populated classes (domain-range information is limited and inheritance is taken into account); (2) indirect relationships; (3) relationships involving literal values. The first fact explains that Q_1 is executed faster than Q_{12} , even if it implies twice as many (*SeRQL*) calls, because there are 47,821 actors starring in films in DBpedia while there are just 3,224 languages related to a country. The second and third facts explain why Q_9 is the slowest query, because answers are obtained in DBpedia through 21 indirect OTs, with mediating concepts (such as airlines, company, person, military conflict, etc.) which relate to both DBpedia entities “island” and “Spain”. In addition the SWETO ontology contains multiple literal values for “Spain” (corresponding to instances of Spanish cities) that need to be analysed.

Table 11.2: Examples of queries after and before using the DBpedia dataset + heuristics (precision)

(Q_i) NL Query: after / before DBpedia	N°Ont	Answers	Seconds	Calls
Q_1 : How many languages are used in Islamic countries?	2/ 2	170/ 0	95.2/ 34.5	1078/ 419
Q_2 : Which Russian rivers end in the Black Sea	3/ 1	4/ 1	41.3/ 27.3	639/ 428
Q_3 : Who lives in the white house	4/ 3	12/12	17.9/ 13.7	310/ 144
Q_4 : Give me airports in Canada	2/ 1	156/155	23/ 14.22	157/ 40
Q_5 : List me Asian countries	6/ 6	64/ 72	15.3/ 67.4	298/ 1308

⁷⁴ The experiments were performed on a 3GHz Intel Pentium dual core processor with 19GB RAM, Redhat Enterprise, Linux, 374GB local disk.

Q ₆ : Give me the main companies in India	2/ 2	710/ 386	17.4/ 43.9	298/ 588
Q ₇ : Give me movies starring Jennifer Aniston	3/ 2	28/ 23	10.7/ 4.5	94/ 22
Q ₈ : Which animals are reptiles?	9/ 8	2518/ 23	42.8/ 7.1	165/ 49
Q ₉ : Which islands belong to Spain	3/ 3	13/ 7	206/ 104	387/ 2617
Q ₁₀ : Find all the lakes in California	2/ 2	37/ 2	13.6/ 12.9	103/ 258
Average (10 queries) – after / before DBpedia	3.6/ 3	371/68	48.3/32	352/587
Q ₁₁ : Find me university cities in Japan	7/-	19/-	68/-	1087/-
Q ₁₂ : Tell me actors starring in films directed by Francis Ford Coppola	3/-	135/-	120/173	574/-
Q ₁₃ : Show me Spanish films with Carmen Maura	1/-	2/-	30.5/-	477/-
Q ₁₄ : Give me English actors that act in Titanic	1/-	4/-	144/-	3340/-
Q ₁₅ : Give me tennis players in France	1/-	29/-	14.7/-	113/-
Q ₁₆ : Television shows created by Walt Disney	1/ -	8/ -	9.4/ -	137/ -
Average (16 queries) – after DBpedia	3.1	244.3	54.3	578.5

Table 11.3: Fusion with DBpedia – after precision heuristics / before precision heuristics

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Avg
Secs	16.5/516	6.7/25	1.8/2	0.3/77	50.9/30	55.7/219	0.1/6.5	74.43/255	3.6/2.3	1.6/3	126/940	30.7/188
Calls	11/771	16/177	14/16	1/311	131/138	370/1148	44/53	19/2803	14/26	38/40	579/3792	112/843

We do not always have enough ontological context to focus on precision when, because of heterogeneity, there are many alternative translations. Take the query Q₁₄: *Give me English actors that act in Titanic*. Although DBpedia contains several mappings for “English actors”, “act” and “Titanic” an ontological translation for the user query can only be found by splitting the compound English actors into two⁷⁵. This query requires an unusual number of calls (3340) to find the OTs that instantiate the QTs <actors/ English?, act, Titanic> and <English, ?, actors>. There are 25 OTs in DBpedia for the first QT linking the class “Actor” to various instances of “Titanic” (Cinematic_Titanic, Titanic_1943_film, Titanic_1953_film, Titanic_1997_film, etc.), through several domain relations (starring, director, producer, academyawards, etc), because the matches for the linguistic relation “act” (the ontological property “act” and various instances, such as “The act” or “Sister act”) turn out not to be relevant for the query. Similarly, the second QT is mapped to 26

⁷⁵ “English actor” is the exact label for several DBpedia instances of actors, none of them starring in Titanic.

DBpedia OTs formed with various ad_hoc relations (residence, ethnicity, location, hometown, etc, and the duplicated properties: birthplace, birthPlace and born) between the class “Actor” and the instance “England” and “English_people”. Moreover, the keyword “English” alone produces several mappings that had to be analysed to determine or not their relevance (e.g., English language, English people, English channel, English football, England, etc). PowerAqua combines the partial answers to extract the final set of answers (the English actors: Bernard Hill, Ian Holm and Kate Winslet starring in Titanic 1997, Brian Aherne and Ian Holm starring in Titanic 1953 and S.O.S. Titanic respectively).

Yet, the needed filtering heuristics impose a toll on recall - e.g., new DBpedia answers are obtained for Q_5 but the answers from one ontology (KIM) are missed, producing a loss in total answers (64/72). Moreover, in Q_{13} , relevant films are missed because the mapping “spanish_language” was not selected and only OTs formed with the mapping “Spain” were retrieved. Notwithstanding, generally, we cannot see any negative effect on recall if we compare the total number of final answers with the previous version of PowerAqua but quite the contrary (371 answers on average after including DBpedia with respect to 68 in the previous version).

Table 11.3 compares the average fusion times before and after applying the fusion heuristics to reduce the number of calls to the indexes, and therefore improve efficiency. The average fusion times have been reduced from 188 to 30.7 seconds. We could not see any loss in recall due to the new fusion heuristics in this query sample.

Nevertheless, there are many open grounds for exploration of techniques which can: (1) lead to better mappings, e.g., “British actors in Titanic” translates into multiple OTs but with no answers after fusion, because British is not mapped to England, and the resultant answers like “Kate Winslet” are related to England but not to Britain; (2) improve the selection of mappings by trust mechanisms and user feedback; (3) exploit the explicit linkage for bridging data in the Linked Data world. For instance, if no ontology offers a complete translation of a single QT, instead of obtaining

partial translations, explicit connections stated across different sources (*owl: sameAs*) can be used to efficiently search for cross ontology connections across entities as if they were part of the same graph.

11.6 Evaluating PowerAqua's response time when using different semantic storage platforms

To address the suboptimal performance of semantic storage, PowerAqua has been integrated with the Virtuoso semantic storage platform. The evaluation of this solution is reported in the following subsection 11.6.1.

The SW community has yet to propose standardized benchmarks to evaluate cross-ontology open-domain QA systems. Nonetheless, we have tested our algorithms with a significant amount of distributed semantic metadata of varying levels of quality and trust and different domains. However, we also report in subsection 11.6.2 on a small-scale test to measure the performance of PowerAqua using Watson to access online semantic data.

11.6.1 Using Virtuoso as a semantic storage platform

Aiming to assess the time performance of PowerAqua when introducing Virtuoso as a new semantic storage platform, we have re-run the evaluation presented in Section 11.5. The results of this evaluation can be seen in Table 11.4. The first column shows a subset of 16 queries used to test the system. The second column shows the performance of PowerAqua before DBpedia was integrated within the semantic search space and using Sesame as the main semantic storage platform. The third column shows the performance of PowerAqua when integrating DBpedia as part of the search space and, the last column shows the performance of PowerAqua when integrating Virtuoso as the main semantic storage platform. As we can see in the table, the average query response time has diminished considerably, from 54.3 to 18.5 seconds, i.e., a 65% reduction.

Table 11.4. Different performance times before and after adding DBpedia and filtering heuristics, and after the Virtuoso integration

NL Query:	Before DBpedia	After DBpedia.	After DBpedia
-----------	----------------	----------------	---------------

		(Sesame)	(Virtuoso)
How many languages are used in Islamic countries?	34.5	95.2	30
Which Russian rivers end in the Black Sea	27.3	41.3	13
Who lives in the white house	13.7	17.9	15
Give me airports in Canada	14.22	23	16
List me Asian countries	15.3	67.4	25
Give me the main companies in India	43.9	17.4	17
Give me movies starring Jennifer Aniston	4.5	10.7	5
Which animals are reptiles?	7.1	42.8	16
Which islands belong to Spain	104	206	33
Find all the lakes in California	12.9	13.6	13
Tell me actors starring in films directed by Francis Ford Coppola	120	173	22
Find me university cities in Japan	-	68	35
Show me Spanish films with Carmen Maura	-	30.5	10
Give me English actors that act in Titanic	-	144	26
Give me tennis players in France	-	14.7	11
Television shows created by Walt Disney	-	9.4	10
Average response time	32	54.3	18.5

Lessons and open issues: This huge decrease in the query response time obtained with the use of Virtuoso is a positive sign, indicating that the latest solutions for semantic storage can efficiently handle the growth of semantic resources, thus increasing the potential of applications that rely on them, such as PowerAqua. As public SPARQL end points⁷⁶ are also based in Virtuoso, we also implemented a plugin to query them. However, they could not be used because currently SPARQL end points do not expose the SQL port to the public. As a result they have to be accessed by HTTP services⁷⁷, rather than through the SQL interface (JDBC) which provided better performance. In addition, network delays and tighter constraints on the web services (e.g., query timeouts, number of users) make the QA process slower.

11.6.2 Using the Watson SW gateway

As said before, despite having tested our algorithms with a significant amount of distributed semantic metadata of varying levels of quality, an issue remains nonetheless open in all previous

⁷⁶ <http://dbpedia.org/sparql>

⁷⁷ Accessed using Jena arq libraries: <http://jena.sourceforge.net/ARQ/cmds.html#arq.remote>

evaluations: the use of our own collected datasets to perform the experiments. Here, we report on a small-scale test to measure the performance of PowerAqua using Watson to access online semantic data.

The performance was tested with a set of 27 queries, presented in Table 11.5, answered by one or more ontologies in Watson. Total recall cannot be measured, therefore, we obtained the following averages in the aggregated results: 0.77 for precision (p): which results are valid from all set of results; 0.83 for precision@1 (p@1): which results are valid from all the results ranked in first position; and 0.69 for recall@1 (r@1): which of the valid results are ranked in first position with respect to the total of all valid results. The ranking criterion used here is the combination ranking. It took an average of 27.7 seconds to translate the NL query into the OTs and 5.43 seconds for fusion, a total 33.1 seconds. Thus, we achieved similar response times with Watson as in previous experiments with online repositories.

Table 11.5. Results on Watson evaluation

NL Query	n° sources	p	p@1	r@1	secs
Which Russian rivers flow into the Black sea?	2	1/1	1/1	1/1	21.6 + 0
Which animals are reptiles?	12	13/13	13/13	13/13	5.7+15.5
Where is Kazakhstan?	13	1/19	1/2	1/1	45.4+4.3
What borders Croatia?	19	6/20	1/1	1/6	11+7.4
Where is Islam practiced?	10	6/20	6/20	6/6	14.7+22.2
Who knows Enrico Motta?	27	24/24	2/2	2/24	88.9+9.3
Who has an interest in ontology evaluation?	11	2/2	2/2	2/2	31.2+2.76
Who works at AIFB?	37	30/31	3/4	3/30	74.0+28.2
Which organizations are working on the Semantic Web	13	17/17	3/3	3/17	31.2+8.0
Who attended both at ISWC2006 and ESWC2006?	53	72/72	5/5	5/72	169.3+8.0
Which countries are in Europe?	3	7/9	7/7	7/7	12.5+8.6
Does Jones Marion takes steroids?	2	1/1	1/1	1/1	7.1+0
Who attended ESWC2007?	27	25/25	1/1	1/25	30.2+8.9
Where is Tom Heath working?	30	7/32	0/2	0/25	36.4+8.1
What is Sierra Leone?	11	1/15	1/2	1/1	8.4+4.32
In which city is Barajas international airport?	1	1/1	1/1	1/1	3.7+0
What airport is in Washington?	1	1/1	1/1	1/1	4.8+0
Who works at the Open University?	3	5/5	5/5	5/5	28.0+0.8
Which sea is close to Volgograd?	4	2/2	1/1	1/2	4.4+0.3
What is the religion in Russia?	2	3/3	3/3	3/3	4.6+0
Which are the universities based in Madrid?	1	2/2	2/2	2/2	3.9+0

Where is Odessa?	2	1/1	1/1	1/1	2.8+0
Find me all cities in California	11	6/20	6/18	6/6	14.7+6.2
In which countries are earthquakes?	4	1/39	1/39	1/39	5.9+2.2
Name the president of Russia	2	1/1	1/1	1/1	4.8+0
Give me oil industries in Russia	2	1/1	1/1	1/1	12.8+0
Give me all oceans and seas	10	11/11	6/6	6/11	23.8+1.7

Lessons and open issues: PowerAqua selects the ontologies relevant for a user query on the fly as part of the querying process. The main advantage of Watson is that it provides an infrastructure to automatically discover ontologies in the SW with zero cost (PowerAqua can find answers from any of the datasets crawled by Watson without being previously aware of them). However, semantic sources in the open Web appear to have many quality issues. The size and quality of ontologies found in Watson, which includes a large number of small, lightweight ontologies (often not populated and not fit for QA purposes) and foaf files, is lower than those added in our repositories. The semantic data is often duplicated, noisy, or it does not have a schema associated to it (an ontology split into different files that are not recognized as part of the same graph). These quality issues hamper the performance of the system to find answers.

11.7 Summing up and conclusions

We are quickly reaching the critical mass required to enable a true vision of large scale, distributed SW with real-world RDF datasets. This transition from restricted domains to real world scale structure data, in terms of number and size, stimulated by Linked Data initiatives, adds a new dimension of scalability both for applications and back-end technologies that aim to exploit the SW. The emergence of Linked Data semantic sources, in particular DBpedia, with millions of RDF statements extracted from Wikipedia, leads to new query answering possibilities, beyond prototypes and proof of concepts. However, exploiting this relevant information, characterized by openness, heterogeneity and scale, to extract answers to user queries is a major challenge. In fact, as analysed in Section 11.2, most current scalable Linked Data tools for end-users, which attempt to hide the complexity behind SW formalisms, have limited reasoning capabilities and only perform a shallow

exploitation of these data. In particular, mappings are either found a priori or the user is asked to disambiguate between competing interpretations. Moreover, because keywords have limited expressivity, these systems do not perform semantic ranking and fusion of data across different semantic sources. Thus, formulating more complex queries requires the use of formal languages, which hinder the widespread exploitation of the web of data for the non-expert user.

Here we looked at the challenges that need to be addressed by the complex QA task in order to scale to billion of triples of Linked Data content, to efficiently extract answers from large and highly heterogeneous community-driven open data. We discussed the issues and lessons learned from our experience of integrating PowerAqua as a front-end to a subset of Linked Data sources, in particular DBpedia. As such, PowerAqua aims to go one step beyond the state of the art on user-centric interfaces for Linked Data by introducing mapping and fusion techniques needed to answer a user query by means of multiple sources, and applying filtering heuristics and ranking algorithms to cope with large-scale information sources, using the context of the user query to elicit the best semantic answers. Our first informal experiments support our claim that, in fact, it is feasible to obtain answers to user queries by composing information across the huge amount of semantically rich information extracted from semantic sources and Linked Data sources such as DBpedia, even in its current form, where the strength of Linked Data is more a by-product of its size than its quality. We believe that the challenges we have faced are useful learned lessons that are applicable to the wider SW community and our experiences can be extrapolated to a variety of end-user applications that wish to scale up to what possibly is the greatest wealth of structured data on the Web.

Based on this analysis, PowerAqua provides a step towards the realization of scalable and effective SW applications, able to deal with the new layers of complexity introduced by the continuous growth of the SW. Although more work still needs to be carried out on our back-end infrastructure to cover not just a subset but all Linked Data available, we believe the effectiveness and the accuracy of the system are likely to improve as the SW grows and semantic search engines mature. As more information becomes available and the quality of the data improves, it will become

possible for PowerAqua to focus primarily on precision rather than recall, thus leading to better accuracy and speed.

Meanwhile, in order to interface PowerAqua to new large-scale repositories, we have experimented with different ontological platforms in the context of large-scale QA, i.e. Virtuoso and Watson, to provide PowerAqua with more efficient access to large amounts of semantic data. On the basis of these experiments, we are optimistic that PowerAqua (and other query-intensive interfaces) can scale to a huge amount of semantic information as long as the semantic software it is based on can efficiently keep up with the growth of the semantic sources.

Chapter 12 Conclusions and future directions

A description of the PowerAqua system has been published in the Semantic Web Journal: Report on Tools and Systems (Lopez et al., 2011a).

W: This is fantastic! It is like in this story, “A Logic Named Joe” (Leinster, 1946): a tool exploiting a network of resources containing the answers to all your questions.

S: It is even better as it constantly evolves from the contributions of thousands of people around the world.

W: Does it really work that smoothly?

S: To be honest with you W, there is still a lot of work to do to make it fully operational, the foundations are basically here and it proves that it is feasible. Not only that, it also opens the way to new perspectives.

W: Like what?

S: Like the possibility to link it to classical Web navigation, providing the user not only with the straight answer to her question, but also with a link to the biography of Kurt Cobain, or to an online store to listen to, and buy, Nirvana’s albums.

W: Hey S, you are getting passionate again! Now, you will think I am an annoying imaginary person, but there is one thing that still bothers me: Nirvana does not exist anymore, and some of the group members have only been part of it for a time. How is that handled?

S: This is actually an excellent question W, but it is late already, so I guess it will have to be another story

Dialog extracted from the Nodalities Magazine (d’Aquin and Lopez, 2008).

12.1 Conclusions and contributions

The research presented here tackles the problem of supporting users in locating and querying information on the Semantic Web (SW). In doing so, we have sought an answer to the following question: *“how can we support end users in querying and exploring this novel, massive and heterogeneous structured information space?”*.

Ontology search engines and gateways to access online semantic information, such as Swoogle, Watson or Sindice are based on keywords and have limited expressivity to represent user needs. Typically they are unable to express relationships between search terms, to cope with ambiguous terms or to describe complex queries. The novelty of our approach is in the integration of ideas from traditional QA towards the realization of scalable user-friendly interfaces for the SW, to provide an open Natural Language (NL) Question Answering (QA) interface for the SW. The result is

PowerAqua, the only QA system we are aware of, which is able to take advantage of any semantic data available on the open SW to interpret, integrate and answer user queries posed in NL.

In this chapter, we present the conclusions and contributions related to the research questions addressed by this work (introduced in Chapter 1.5) and discuss future work and open issues.

12.1.1 Contributions of semantic QA to the state of the art

A **first main contribution** of this thesis is to show that there is a good evidence for open semantic QA as a novel approach to push the boundaries and address the issues raised by the shortcomings of current work (presented in Chapter 2).

In Chapter 3-4, we present the requirements and design of PowerAqua, as an ontology-based QA system that aims to go beyond the state of the art by addressing the limitations of current approaches:

1) PowerAqua is not limited by the single-ontology assumption and it does not impose any pre-selection or pre-construction of semantic knowledge (closed-domain limitation), but rather explores the increasing number of multiple, heterogeneous sources created on the Web⁷⁸ based on NL.

2) PowerAqua can answer queries by composing information from multiple heterogeneous semantic sources of varying quality and domains. To this purpose, PowerAqua has developed syntactic, semantic and contextual information processing mechanisms to support query disambiguation, knowledge fusion (to aggregate similar or partial answers) and ranking mechanisms.

In the context of the SW, QA over semantic data distributed across multiple sources has been introduced as a new paradigm, which integrates ideas from traditional QA research into scalable SW

⁷⁸ PowerAqua retrieves information that has been crawled and indexed by Watson or exists in specific online repositories.

tools, aiming to master scalability and heterogeneity together with user-friendliness and expressivity. Table 12.1 extends Table 2.3 by comparing PowerAqua to the systems reviewed in Section 2.6. As illustrated in this table, semantic open QA goes beyond other methods in supporting end-users in querying and exploring the heterogeneous SW.

Table 12.1. Comparison of PowerAqua with respect to querying approaches classified according to different search criteria (\checkmark = yes, \emptyset = no, +/- = limited)

Criteria	Input		Scope			Search environment (research issues)				Sources (scalability)	
	Expressivity	Reasoning services	Portability	Open Domain	Heterogeneity	Ranking	Disambiguation	Fusion	Sources on-the-fly	Scale SW	Scale Web
NLIDB	\checkmark	\checkmark	+/-	\emptyset	\emptyset	\emptyset	\checkmark	\emptyset	\emptyset	\emptyset	\emptyset
QA-Text	\checkmark	\emptyset	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\emptyset	\checkmark	\emptyset	\checkmark
Ontology-QA	\checkmark	\checkmark	\checkmark	\emptyset	\emptyset	+/-	\checkmark	\emptyset	\emptyset	+/-	\emptyset
Commercial QA	\checkmark	\checkmark	\checkmark	\checkmark	\emptyset	\checkmark	\checkmark	\emptyset	\emptyset	\emptyset	+/-
Keyword search	+/-	\emptyset	\checkmark	\checkmark	\checkmark	\checkmark	+/-	\emptyset	\checkmark	\checkmark	+/-
Mashups	\emptyset	\emptyset	\checkmark	\checkmark	\checkmark	+/-	\emptyset	\checkmark	\checkmark	\checkmark	\emptyset
Facets	\checkmark	\emptyset	\checkmark	\checkmark	\checkmark	\checkmark	\emptyset	\emptyset	\emptyset	\checkmark	\emptyset
PowerAqua	\checkmark	\emptyset	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	+/-	\checkmark	+/-

12.1.2 Our proposal towards Semantic Question Answering

The **second main contribution** of this work explores to what extent we can provide correct answers to questions, without the need for engineering our own data, by taking the SW as a given resource. In this thesis, we report on the complete implementation of the PowerAqua system and each component in its pipeline architecture, to give a comprehensive account of the way the system returns answers to queries (Chapters 5-8).

As a design feature, the architecture includes a plugin specification mechanism that supports a common API to manipulate content independently of the storage platform. As a result, it keeps the query processing independent of the underlying infrastructure and has allowed us to integrate more efficient query platforms into PowerAqua over time (Chapter 4).

After a linguistic analysis the NL query is translated into a triple-based representation (Query-Triples), where the interdependences between query terms are identified to facilitate the exploitation of the query context by the subsequent components (Chapter 5). The PowerMap ontology discovery component dynamically identifies those semantic sources and element mappings that may be

relevant for the user query. Thus, addressing the first research challenge, that is, resource discovery and information focusing. This initial match is performed by means of syntactic techniques and WordNet lexically related words that, in many cases, generate several possible candidate semantic entities, which may provide potential alternative interpretations for a query term. To check the soundness and the semantic nature of the previously identified individual mappings, the PowerMap semantic validation component builds on well-founded ideas from the Word Sense Disambiguation community. PowerMap exploits the background knowledge provided by WordNet and the ontological hierarchy of the candidate entities to identify and disambiguate between the different possible interpretations of the same query term across ontologies, by acquiring, when possible, the WordNet sense of the matched concept (Chapter 6).

By exploring the context and ontological relationships surrounding the candidate entity mappings obtained by PowerMap, the TSS assembles the element level matches and identifies the ontological triples (OTs) from different ontologies that better match with the set of Query-Triples, which represent the user information needs (Chapter 7). Thus, addressing the second research challenge, that is, mapping user terminology into ontology terminology. Obtaining the set of OTs from which answers can be derived is a complex and costly procedure. In order to maintain real time performance in a scenario of perpetual change and growth, the TSS steps are executed following increasingly complex and expensive stages.

As a side effect of using multiple ontologies a query can have similar, partial or alternative translations derived from different ontological interpretations in different ontologies. To address the third research challenge, that is integrating information from different information sources, in this thesis we also proposed collective methods for combining and ranking those answers obtained from multiple distributed heterogeneous data sources, which at the same time, can aggregate composite answers, filter irrelevant answers, fuse similar answers, and elicit the most accurate answer(s) (Chapter 8). PowerAqua includes three different ranking algorithms, which sort the final answers according to different criteria based on: (a) the confidence of the mapping algorithm on the

ontological facts from which the answer is derived (b) the confidence of the disambiguation algorithm on the interpretation of the answer, and (c) the *popularity* of the answer across ontologies after applying the merging algorithm.

Taking a step back from the details of the algorithms, we believe that exploiting the heterogeneous SW is essentially about discovering interesting connections between items in a meaningful way. PowerAqua provides a NL front end, which makes it possible to perform QA on the SW, retrieving the exact answers to users' requests in the form of pieces of ontological knowledge, hence supporting such a discovery process across multiple heterogeneous sources. This contrasts sharply with formal query languages for the SW, such as RDQL or SPARQL, which not only can be used solely by experts but, in addition, are unable to perform data integration across heterogeneous ontologies and cannot be used to support such process of discovering and linking information spread across multiple sources.

12.1.3 Evaluation

There are not yet any standards to evaluate systems like PowerAqua. Therefore, our **third main contribution** is the design and the results from five evaluations, which in addition to providing data about PowerAqua's competence also provide important insights into the issues, current strengths and limitations related to using the SW as the target answer set in QA. Because of their nature, our experiments can be used as means to assess the quality of the SW itself.

The conducted experiments, based on reusing existing benchmarks and through ad-hoc, user-centric evaluations, have shown the feasibility of the approach, and the ability of PowerAqua to provide accurate responses to users' requests from distributed SW content. Each of the five evaluations presented here allow us to extract useful lessons and open issues for developers in the wider SW community:

- 1) In the evaluation of PowerAqua's ability to map a NL query into several ontologies (Chapter 9.2), we found that PowerAqua was able to answer correctly more than half of the queries (a

positive result considering the openness of the scenario). The evaluation highlighted that most of the failures were due to lexical level issues, which originated as a consequence of the high levels of heterogeneity combined with poorly modelled and incomplete, or barely populated, ontologies.

- 2) The merging and ranking evaluation showed an improvement in the quality of answers with respect to a scenario where the merging and ranking algorithms were not applied (Chapter 9.3). Besides obtaining more accurate integrated answers to questions by exploiting the increasing amount of collectively authored, highly heterogeneous, semantic data, it allows PowerAqua to answer users' requests that extend beyond the coverage of single datasets and build across ontological statements from different sources. The best ranking measure was the one based on the confidence (quality) of the mappings, while semantic similarity and popularity ranking measures were hampered by the sparseness and incompleteness of the data on the SW. Nonetheless, we believe that the success of ranking strategies based on popularity across ontologies goes hand in hand with the growth of the SW.
- 3) The usability study showed that despite PowerAqua's still limited linguistic coverage and the habitability problem typical of NL interfaces (the user requires a bit of familiarization with the system to know what is possible to ask), users like the flexibility of being able to pose NL queries and PowerAqua obtained a good usability score (Chapter 9.4).
- 4) PowerAqua has also been evaluated as a query expansion module in a more complex semantic IR experiment, reusing an IR evaluation benchmark and an independent set of questions (the TREC WT10G collection) to assess whether this system provides an improvement in precision/recall over an IR baseline (Chapter 10). This evaluation highlighted the potential of using SW information to enhance searches on the Web, but also the sparseness and incompleteness of the SW when compared to the Web. This second evaluation is therefore an example of how to re-use an IR gold standard to evaluate semantic search.

- 5) The evaluation on scalability shows that PowerAqua's response time increases when a large Linked Data source such as DBpedia is added (Chapter 11). The combination of scale, heterogeneity and quality issues became a bigger challenge with the appearance of Linked Data sources. In particular the decrease in speed was due less to the increase in the number of resulting hits than to the increase in query response times in the semantic storage for large datasets. This shows that improvements in PowerAqua's will also be linked to the evolution of semantic storage and querying platforms.

Our evaluations, tested on a corpora of significant scale, show promising results proving that it is feasible to answer questions with not just one but many heterogeneous ontologies selected on the fly in a reasonable time, despite the major issues that currently hamper the system, such as sparseness and incomplete conceptualizations. Indeed, the effectiveness and accuracy of the system are likely to improve with the growth of the SW. As more and more information becomes available on the SW, it will become possible for this kind of systems to focus primarily on precision rather than recall, thus leading, in principle, to better accuracy and faster performance. Nevertheless, a major conclusion of our research is that the SW is already a powerful source of background knowledge, which can be exploited to successfully tackle real world complex tasks, such as QA, in particular to support queries that can only be answered by combining and adding information present in different sources.

12.1.4 Scalability towards a Web environment

Finally, the fourth research challenge, that is, scalability to large amounts of heterogeneous data, is addressed to a significant extent by the last main contribution of this work. We have presented and tested in Chapter 11 how our research model has been adapted to be able to scale to an open, large and highly heterogeneous environment, not only on the number of ontologies but also on their size, such as the KBs offered by the Linked Data initiative.

Following an ambitious future research direction and as part of this last **fourth contribution**, although the system is primarily built for QA, in addition to its core functionality, we have also

experimented with embedding it into an advanced IR system (Chapter 10) and to trigger contextualized searches in a popular search engine (Chapter 10.7). In these settings, PowerAqua allows users to phrase their questions in NL and attempts to find an answer by performing QA on online ontologies. Then, the answers found are returned to the user and also redirected as keywords to an IR system that returns relevant documents, thus enhancing searches on the Web with semantic information.

Balancing the complexity of the querying process in an open-domain scenario such as the Web (e.g., to be able to handle complex questions requiring making deductions on open-domain knowledge, or to capture the interpretation of domain-specific adjectives and superlatives) and the amount of semantic data is still very much an open problem and research in this area is crucial for the realization of the SW. During the development of PowerAqua the SW has evolved continuously. PowerAqua was first envisioned in 2006 in the context of a paradigm shift from the first generation of classic semantic systems, characterized by being closed-domain, KB-centric semantic systems, to the next generation of open SW applications. Our vision was that on the one hand, a QA system could take advantage of the vast amount of heterogeneous semantic data and, on the other hand, that the SW could benefit from a user-friendly NL interface. As we predicted, recent years have witnessed the rise of ontology-based QA as a new paradigm of research and the appearance of proprietary QA systems able to obtain structured answers in homogeneous large KBs. It can also be argued that the availability of large scale Linked Data defines a turning point in the evolution of the SW, signalling the transition from restricted domains to real world, large scale structured data sources. At the same time, Linked Data added a new layer of complexity, requiring applications to leverage precision and recall to scale up to large amounts of data. Thus, to support large scale QA and to confront the high levels of heterogeneity, quality and noise present in some of the available semantic resources, in addition to its iterative algorithms, PowerAqua has implemented a set of filtering heuristics to limit the space of solutions provided by these resources.

Publishing errors and inconsistencies arise naturally in an open environment like the Web. Thus, imperfection (gaps in coverage, redundant data with multiple identifiers for the same resource, conflicting data, undefined classes and properties without a formal RDFS or OWL description, invalid or literal datatypes, etc.) can be seen as an inherent property of the Web of data. As such, the strength of the SW will be more a by-product of its size than its absolute quality. In other words, we believe that in large-scale semantic systems, intelligence becomes a side effect of a system's ability to operate with large amounts of data from heterogeneous sources, rather than being primarily defined by its reasoning ability to carry out complex tasks.

The overarching contribution of this thesis is one step towards the vision of the SW becoming a reality through the convergence of user-friendly traditional QA approaches and open interfaces and search models for the SW. This ability to extract knowledge from the SW without any domain restrictions according to a user query is also the first step towards the integration of SW models with scalable IR models, and as such, the combination of semantic spaces provided by the SW and non-semantic spaces in the Web.

12.2 qFuture work and extensions

Important research topics still lie ahead, not fully addressed in this thesis or in close relation to the ones we have addressed. In this section we discuss unsolved limitations, further incremental improvements, as well as new interesting research lines that can be pursued to enhance the current model:

1) Open QA on the SW still suffers from the well-known problems of knowledge sparseness and incompleteness, together with the cost of building and maintaining rich semantic sources. For these reasons, open QA cannot yet compete with major search engines, like Google, Yahoo or Microsoft Bing. Added to this, balancing the complexity of the querying process, performance and scalability is still an open problem. The major challenge is, in our opinion, the combination of scale with the considerable heterogeneity and noise intrinsic to the SW. As future work, basing our premises on the

continuous growth of semantic data, we aim to focus on the development of algorithms that help to improve the precision of answers retrieved by PowerAqua, leaving recall as a secondary goal since, as indicated in our experiments, we expect recall to grow in line with the growth of available semantic content. Of course, as the size of the SW increases, additional experimental evaluations will be needed to identify the optimal trade-off between recall and precision.

2) Because of the openness of the scenario, our research has focused on factual QA (such as in TREC), as a first step to research on more ambitious and complex forms of QA. Future work should focus on extending the range of queries the system is able to handle, in response to new scenarios or user needs, for example, to support complex factual queries with negations, comparatives, superlatives, and temporal reasoning, among others.

3) Improvements in scalability and efficiency of the underlying storage platforms will allow us to further improve the search performance of the PowerAqua algorithms, for example, by being able to explore longer paths (relations) between two given entities within the same or across ontologies.

4) Information on the SW originates from a large variety of sources and exhibits differences in quality, and therefore, as the data is not centrally managed or produced in a controlled environment, trust becomes an issue. The development of trust mechanisms in the SW could open many interesting doors for future research especially for ranking mechanisms that cope with large-scale information sources (see Chapter 8.5).

5) Finally, we also aim to carry out further experiments in integrating PowerAqua with standard IR approaches, to complement the answers given by PowerAqua with Web pages and enhance the expressivity and performance of traditional search engines with semantic information (as detailed in Chapter 10). As the number of annotated sites increases, the answers to a question extracted by PowerAqua, in the form of lists of entities, can be used as a valuable resource for discovering classic Web content that is related (annotated) to the answers given as ontological entities. PowerAqua aims

to use the data on the SW and the Web to provide a service that extends capabilities from querying a large number of unconnected sources to more interlinked ecosystems of data.

In conclusion, PowerAqua provides the most ambitious system currently available to find information on the SW. As the SW evolves, we expect PowerAqua and similar systems to eventually evolve into powerful and comprehensive semantic querying solutions, thus bringing home the vision of a SW providing precise answers to precise question, and supporting the dynamic integration of information drawn from different sources in response to user needs.

Bibliography

- Adams T., Dullea J., Clark P., Sripada S. and Barrett T. (2000): *Semantic Integration of Heterogeneous Information Sources Using a Knowledge-Based System*. In Proc. of the 5th International Conference on Computer Science and Informatics.
- Aleksovski, Z., Klein, M., ten Katen, W. and van Harmelen, F. (2006): *Matching Unstructured Vocabularies using a Background Ontology*. In Staab, S., Svátek, V., editors. In Proc. of the 15th International Conference on Knowledge Engineering and Knowledge Management, Podebrady, Czech Republic. Springer LNCS Vol. 4284.
- Aleman-Meza, B., Hakimpour, F., Arpinar, I.B. and Sheth, A.P. (2007): *SwetoDblp Ontology of Computer Science Publications*. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 5(3): p.151-155. Elsevier Science Publishers B.V. Amsterdam.
- Androutsopoulos, I., Ritchie, G.D. and Thanisch, P. (1993): *MASQUE/SQL - An Efficient and Portable Natural Language Query Interface for Relational Databases*. In Proc. of the 6th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Edinburgh, p.327-330. Gordon and Breach Publisher Inc.
- Androutsopoulos, I., Ritchie, G.D. and Thanisch P. (1995): *Natural Language Interfaces to Databases - An Introduction*. Natural Language Engineering, 1(1): 29-81. Cambridge University Press.
- Auer, S. and Lehmann, J. (2007): *What have Innsbruck and Leipzig in common? Extracting Semantics from Wiki Content*. In Franconi, E., Kifer, M., May, W., editors. In Proc. of the 4th European Semantic Web Conference, Innsbruck, Austria. Springer LNCS Vol. 4519.
- Baeza, R. A. and Raghavan, P. (2010): *Next generation Web search*. In Ceri, S. and Brambilla, M., editors. In Search Computing, LNCS Vol. 5950, p.11-23. Springer, Heidelberg.
- Bailey, P., Craswell, N. and Hawking, D. (2003): *Engineering a multi-purpose test collection for web*. Information Processing and Managment, 39(6): 853-871.
- Banerjee S. and Pedersen T. (2003): *Extended Gloss Overlaps as a Measure of Semantic Relatedness*. In Proc. of the International Joint Conference on Artificial Intelligence, p.805-810. Stockholm, Sweden.
- Bartell, B. T. (1994): *Optimizing Ranking Functions: A Connectionist Approach to Adaptive Information Retrieval*. PhD thesis, University of California, San Diego.
- Basili, R., Hansen, D. H., Paggio, P., Pazienza, M. T. and Zanzotto, F. M. (2004): *Ontological resources and question answering*. Workshop on Pragmatics of Question Answering, held jointly with HLT-NAACL 2004. Boston, Massachusetts.
- Basili, R., De Cao, D. and Giannone, C. (2007): *Ontological Modeling for Interactive Question Answering*. In Tari, Z., editor. In Proc. of the OTM confederated international conference on On the move to meaningful internet systems (1): 544-553. Springer-Verlag.
- Berners-Lee, T., Hendler, J. and Lassila, O. (2001): *The Semantic Web*. Scientific American, 284(5): 33-43.
- Berners-Lee, T. (2006): *Tabulator: exploring and analyzing linked data on the semantic web*. In Proc. of the 3rd International Semantic Web User Interaction Workshop. Athens, Georgia.
- Bernstein, A., Kaufmann, E., Kiefer, C. and Burki, C. (2005): *SimPack: A Generic Java Library for Similarity Measures in Ontologies*. Technical report, University of Zurich, Department of Informatics.
- Bernstein, A., Kauffmann, E., Kaiser, C. and Kiefer, C. (2006). *Ginseng: A Guided Input Natural Language Search Engine*. In Proc. of the 15th workshop on Information Technologies and Systems (WITS 2005), p.45-50. MV-Wissenschaft, Münster.
- Bhagdev, R., Chapman, S., Ciravegna, F., Lanfranchi, V. and Petrelli, D. (2008): *Hybrid Search: Effectively combining keywords and semantic searches*. In Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M., editors. In Proc. of the 5th European Semantic Web Conference. Tenerife. Springer LNCS Vol. 5021.
- Bilenko, M. and Mooney, R. J. (2003): *Adaptive duplicate detection using learnable string similarity measures*. In Proc. of the International Conference on Knowledge Discovery and Data Mining (KDD-2003), p.39-48, Washington.
- Bizer, C., Heath, T. and Berners-Lee, T. (2009a): *Linked Data – The Story So Far*. International Journal on Semantic Web and Information Systems, 5(3): 1-22. Elsevier.
- Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R. and Hellman, S. (2009b): *DBpedia: A Crystallization Point for the Web of Data*. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 7(3): 154-165. Elsevier.

- Bouquet P., Serafini L. and Zanobini S. (2003): *Semantic coordination: a new approach and an application*. In Sycara, K. and Mylopoulos, J., editors. In Proc. of 2nd International Semantic Web Conference, p.130-145, Sanibel Island, Florida. Springer LNCS, Vol. 2870.
- Brooke, J. (1996): *SUS: a quick and dirty usability scale*. In Usability Evaluation in Industry, p.189-194. Taylor and Francis.
- Budanitsky, A. and Hirst, G. (2006): *Evaluating WordNet-based measures of semantic distance*. Computational Linguistics, 32(1): 13-47.
- Buitelaar, P., Declerck, T., Calzolari, N. and Lenci, A. (2003): *Language Resources and the Semantic Web*. In Proc. of the ELSNET/ENABLER Workshop, Paris, France.
- Burger, J., Cardie, C., Chaudhri, V., Gaizauskas, R., Harabagiu, S., Israel, D., Jacquemin, C., Lin, C. Y., Maiorano, S., Miller, G., Moldovan, D., Ogden, B., Prager, J., Riloff, E., Singhal, A., Shrihari, R., Strzalkowski, T., Voorhees, E. and Weischedel, R. (2001): *Issues, Tasks and Program Structures to Roadmap Research in Question & Answering (Q&A)*. Tech Report, NIST, Gaithersburg, USA.
- Burke, R., D., Hammond, K., J. and Kulyukin, V. (1997) *Question Answering from Frequently-Asked Question Files: Experiences with the FAQ Finder system*. In Proc. of the World Wide Web Internet and Web Information Systems, 18(TR-97-05): 57-66. Department of Computer Science, University of Chicago.
- Castells, P., Fernández, M. and Vallet, D. (2007): *An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval*. IEEE Transactions on Knowledge and Data Engineering 19(2): 261-272. IEEE Educational Activities Department.
- Cheng, G., Ge, W. and Qu, Y. (2008): *Falcons: Searching and browsing entities on the Semantic Web*. In Wei-Ying, M., Tomkins, A., Zhang, X., editors. In Proc. of the 17th International Conference on World Wide Web, p. 1101-1101, Beijing, China. ACM.
- Cimiano, P., Haase, P. and Heizmann, J. (2007): *Porting Natural Language Interfaces between Domains -- An Experimental User Study with the ORAKEL System*. In Chin, D. N., Zhou, M. X., Lau, T. S. and Puerta A. R., editors. In Proc. of the International Conference on Intelligent User Interfaces, p. 180-189, Gran Canaria, Spain. ACM.
- Cimiano, P. and Minock, M. (2009): *Natural Language Interfaces: What's the Problem? - A Data-driven Quantitative Analysis*. In Proc. of the International Conference on Applications of Natural Language to Information Systems (NLDB 2009), p. 192-206.
- Clark, P., Thompson, J. and Porter, B. (1999): *A Knowledge-Based Approach to Question-Answering*. In the AAAI Fall Symposium on Question-Answering Systems, CA:AAAI, p.43-51.
- Cohen, W., W., Ravikumar, P. and Fienberg, S. E. (2003): *A Comparison of String Distance Metrics for Name-Matching Tasks*. IJCAI Workshop on Information Integration on the Web (IIWeb-03), p.73-78, Acapulco, Mexico.
- Copestake, A. and Jones, K. S. (1990): *Natural language interfaces to databases*. Knowledge Engineering Review. 5(4): 225-249.
- Croft (1986): *User-specified domain knowledge for document retrieval*. In Rabitti, F., editor. In Proc. of the 9th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR 1986), p. 201-206, Pisa, Italy. ACM.
- Cunningham, H., Maynard, D., Bontcheva, K. and Tablan, V. (2002): *GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications*. In Proc. of the 40th Anniversary Meeting of the Association for Computational Linguistics, 54: 168-175. Association for Computational Linguistics.
- D'Aquin, M., Baldassarre, C., Gridinoc, L., Angeletou, S., Sabou, M. and Motta, E. (2007): *Characterizing knowledge on the semantic web with Watson*. In Proc. of 5th EON Workshop at International Semantic Web Conference, p.1-10, Busan, Korea. Citeseer.
- D'Aquin, M., Motta, E., Sabou, M., Angeletou, S., Gridinoc, L., Lopez, V. and Guidi, D. (2008a): *Towards a new Generation of Semantic Web Applications*. IEEE Intelligent Systems, 23 (3): 20-28.
- D'Aquin M., Lopez V., Motta, E. (2008b): *FABiT – Finding Answers in a Billion Triples*. Billion triple challenge at International Semantic Web Conference.
- D'Aquin, M., Sabou, M., Motta, E., Angeletou, S., Gridinoc, L., Lopez, V. and Zablith, F. (2008c): *What can be done with the Semantic Web? An Overview of Watson-based Applications*. In Proc. of the 5th Workshop on Semantic Web Applications and Perspectives, SWAP, Rome, Italy.
- D'Aquin, M., Lopez, V. (2008): *Finding Answers on the Semantic Web*. Nodalities Magazine September / October 2008 issue: http://www.talis.com/nodalities/pdf/nodalities_issue4.pdf.
- Damljanovic, D., Tablan, V. and Bontcheva, K. (2008): *A Text-based Query Interface to OWL ontologies*. In Proc. of the 6th Language Resources and Evaluation Conference (LREC), p. 205-212, Marrakech, Morocco.
- Damljanovic, D., Agatonovic, M. and Cunningham, H. (2010): *Natural Language interface to ontologies: Combining syntactic analysis and ontology-based lookup through the user interaction*. In Aroyo, L., Antoniou, G., Hyvönen, E.,

- ten Teije, A., Stuckenschmidt, H., Cabral, L. and Tudorache, T., editors. In Proc. of the European Semantic Web Conference, Heraklion, Greece. Springer Verlag.
- De Boni, M. (2001): TREC 9 QA Track Overview.
- De Roeck, A., N., Fox, C., J., Lowden, B., G., T., Turner, R. and Walls, B. (1991): *A Natural Language System Based on Formal Semantics*. In Proc. of the International Conference on Current Issues in Computational Linguistics, Penang, Malaysia.
- Ding, L., Pan, R., Finin, T., Joshi, A., Peng, Y. and Kolari, P. (2005): *Finding and Ranking Knowledge on the Semantic Web*. In Gil, Y., Motta, E., Benjamins, V.R. and Musen, M., editors. In Proc. of International Semantic Web Conference, p.156-170, Galway, Ireland. Springer LNCS, Vol. 3729.
- Dittenbach, M., Merkl, D. and Berger, H. (2003): *A Natural Language Query Interface for Tourism Information*. In Proc. of the 10th International Conference on Information Technologies in Tourism (ENTER-03), p.152-162. Helsinki, Finland. Springer Verlag.
- Doan, A., Madhavan, J., Domingos, P. and Halevy, A. (2002): *Learning to Map between Ontologies on the Semantic Web*. In Proc. of the 11th World-Wide Web Conference, p.662-673. ACM.
- Ehrig, M. and Staab, S. (2004): *QOM: Quick ontology mapping*. In McIlraith, S., Plexousakis, D., van Harmelen, F., editors. In Proc. of the International Semantic Web Conference, p.683–697. Hiroshima, Japan. Springer LNCS Vol. 3298.
- Elmagarmid, A. K., Ipeirotis, P. G. and Verykios, V. S. (2007): *Duplicate record detection: a survey*. IEEE Transactions on Knowledge and Data Engineering 19(1): 1-16.
- Erling, O. and Mikhailov, I. (2009): *Faceted Views over Large-Scale Linked Data*. In Proc. of the Linked Data on the Web Workshop at the World Wide Web Conference (LDOW2009), Madrid, Spain.
- Euzenat, J., Stuckenschmidt, H. and Yatskevich, M. (2005): *Introduction to the Ontology Alignment Evaluation 2005*. In Proc. of the Integrating Ontologies Workshop.
- Fazzinga, B. and Lukasiewicz, T. (2010): *Semantic Search on the Web*. Semantic Web – Interoperability, Usability, Applicability, 1(1-2): 1-7.
- Fernandez, M., Lopez, V., Sabou, M., Uren, V., Vallet, D., Motta, E. and Castells, P. (2008): *Semantic Search meets the Web*. In Proc. of the IEEE International conference on Semantic Computing, p.253-260, Santa Clara, California. IEEE Computer Society.
- Fernandez, M., Lopez, V., Vallet, D., Castells, P., Motta, E., Sabou, M. and Uren, V. (2009): *Using TREC for cross-comparison between classic IR and ontology-based search models at a Web scale*. In Proc. of the Semantic Search 2009 Workshop at the 18th International WWW Conference, Madrid.
- Fernandez, M., Cantandor, I., Lopez, V., Vallet, D., Castells, P. and Motta, E. (2011): *Semantically enhanced Information Retrieval: an ontology-based approach*. Journal of Web Semantics: Science, Services and Agents on the World Wide Web. Special Issue on Semantic Search. In Press.
- Fernandez, O., Izquierdo, R., Ferrandez, S. and Vicedo J. L (2009): *Addressing Ontology-based question answering with collections of user queries*. Information Processing and Management, 45 (2): 175-188. Elsevier.
- Finin, T., Mayfield, J., Fink, C., Joshi, A. and Cost, R. S. (2005): *Information retrieval and the Semantic Web*. In Proc. of the 38th Hawaii International Conference on System Sciences (Hicss'05) and Management, 45 (2): 175-188. Elsevier.
- Forner, P., Giampiccolo, D., Magnini, B., Peñas, A., Rodrigo, A. and Sutcliffe, R. (2010). *Evaluating Multilingual Question Answering Systems at CLEF*. In Proc. of the conference on International Language Resources and Evaluation (LREC), Malta.
- Frank, A., Hans-Ulrich K., Feiyu, X., Hans, U., Berthold, C., Brigitte, J. and Ulrich, S. (2006): *Question answering from structured knowledge sources*. Journal of Applied Logic, Special Issue on Questions and Answers: Theoretical and Applied Perspectives, 5(1): 20-48. Elsevier.
- Giunchiglia F. and Yatskevich M. (2004): *Element Level Semantic Matching*. Meaning Coordination and Negotiation Workshop at the International Semantic Web Conference.
- Giunchiglia F., Shvaiko P and Yatskevich M. (2004): *S-Match: an algorithm and an implementation of semantic matching*. In Proc. of the 1st European Semantic Web Symposium.
- Gracia, J., Trillo, R., Espinoza, M. and Mena, E. (2006): *Querying the web: a multiontology disambiguation method*. In Wolber, D., Calder, N., Brooks, C. and Ginige, A., editors. In Proc. of the 6th international conference on Web Engineering. Palo Alto, California. ACM.
- Gracia, J., Lopez, V., d'Aquin, M., Sabou, M., Motta, E. and Mena, E. (2007): *Solving Semantic Ambiguity to Improve Semantic Web based Ontology Matching*. Workshop on Ontology Matching at the 6th the International Semantic Web Conference, p.1-12.

- Green, B. F., Wolf, A. K., Chomsky, C. and Laughery, K. (1961) *BASEBALL: An automatic question answerer*. Proceedings Western Joint Computer Conference, 19: 207-216. McGraw-Hill.
- Gruber, T. R. (1993). *A Translation Approach to Portable Ontology Specifications*. Knowledge Acquisition, 5(2): 199-220. Elsevier.
- Guarino, N., Masolo, C. and Vetere, G. (1999): *OntoSeek: Content-based Access to the Web*. IEEE Intelligent Systems, 14(3): 70-80.
- Gueret, C., Groth, P. and Schlobach, S. (2009): *eRDF: Live Discovery for the Web of Data*. Billion Triple Challenge at International Semantic Web Conference, Chantilly, VA, USA, Springer LNCS, Vol. 5823.
- Guha, R. V., McCool, R. and Miller, E. (2003): *Semantic Search*. In Proc. of the 12th International World Wide Web Conference (WWW 2003), 36(1): 700-709, Budapest. ACM Press.
- Guha, R. (2004): *Object Co-Identification on the Semantic Web*. In Feldman, S., Uretsky, M., Najork, M. and Wills C. E., editors. In Proc. of the 13th World Wide Web. New York, USA. ACM
- Gurevych, I., Bernhard, D., Ignatova, K. and Toprak, C. (2009): *Educational Question Answering based on Social Media Content*. In Proc. of the 14th International Conference on Artificial Intelligence in Education, 200: 133-140. IOS Press.
- Hallett, C., Scott, D. and Power, R. (2007): *Composing Questions through Conceptual Authoring*. Computational Linguistics, 33(1): 105-133. MIT Press.
- Harabagiu, S., Moldovan, D., Pasca, M., Mihalcea, R., Surdeanu, M., Bunesco, R., Girju, R., Rus, V. and Morarescu, P. (2000): *Falcon - Boosting Knowledge for Answer Engines*. In Proc. of the 9th Text Retrieval Conference (Trec-9), p.479-488.
- Hendler, J. (2010): *Web 3.0: The Dawn of Semantic Search*. IEEE Computer, 43(1): 77-80. IEEE Computer Society Press.
- Hildebrand, M., Ossenbruggen, J. and van Hardman, L. (2007): *An Analysis of Search-based User Interaction on the Semantic Web*. Information Systems Journal, INS-E0706, p.1386-3681. CWI, Amsterdam, Holland.
- Hirschman, L. and Gaizauskas, R. (2001): *Natural Language Question Answering: The View from here*. Natural Language Engineering, Special Issue on Question Answering, 7(4): 275-300. Cambridge University Press.
- Hovy, E.H., Gerber, L., Hermjakob, U., Junk, M. and Lin, C. Y. (2000): *Question Answering in Webclopedia*. In Proc. of the Ninth Text Retrieval Conference (TREC-9). NIST, p.655-664.
- Hunter, A. (2000): *Natural Language database interfaces*. Knowledge Management. (www.cs.ucl.ac.uk/staff/a.hunter/tradepress/nldb.html)
- Hyvonen, E., Makela, E., Salminen, M., Valo, A., Viljanen, K., Saarela, S., Junnila, M. and Kettula, S. (2005): *MuseumFinland – Finnish Museums on the SemanticWeb*. In Journal of Web Semantics: Science Services and Agents on the World Wide Web. Selected Papers from the International Semantic Web Conference 2004, 3(2): 224-241.
- Ide N. and Veronis J. (1998): *Word Sense Disambiguation: The State of the Art*. Computational Linguistics, 24(1): 1-40.
- Jian, N., Hu, W., Cheng, G. and Qu, Y. (2005): *Falcon-AO: Aligning Ontologies with Falcon*. In the Proc. of the K-CAP Workshop on Integrating Ontologies.
- Katz, B., Felshin, S., Yuret, D., Ibrahim, A., Lin, J., Marton, G., McFarland A. J. and Temelkuran, B. (2002): *Omnibase: Uniform Access to Heterogeneous Data for Question Answering*. In Proc. of the 7th International Workshop on Applications of Natural Language to Information Systems (NLDB), p. 230-234, Stockholm, Sweden.
- Katz, B. and Lin, J. (2003): *Selectively Using Relations to Improve Precision in Question Answering*. In Proc. of the EACL-2003. Workshop on Natural Language Processing for Question Answering.
- Kauffmann, E., Bernstein, A. and Zumstein, R. (2006): *Querix: A Natural Language Interface to Query Ontologies Based on Clarification Dialogs*. In Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M. and Aroyo, L., editors. In Proc. of the 5th International Semantic Web Conference, p.980-981, Athens, USA. Springer LNCS, Vol. 4273.
- Kauffmann, E., Bernstein, A. and Fischer, L. (2007): *NLP-Reduce: A “naïve” but Domain-independent Natural Language Interface for Querying Ontologies*. In Franconi, E., Kifer, M. and May, W., editors. In Proc. of the 4th European Semantic Web Conference, p.1-2, Innsbruck. Springer Verlag.
- Kaufmann, E. and Bernstein, A. (2007): *How Useful Are Natural Language Interfaces to the Semantic Web for Casual End-Users?*. In Aberer, K., Choi, K. S., Noy, N., Allemang, D., Lee, K. I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., and Cudré-Mauroux, P., editors. In Proc. of the 6th International Semantic Web Conference, p.281-294, Busan, Korea. Springer LNCS Vol. 4825.
- Kaufmann, E. (2009): *Talking to the Semantic Web - Natural Language Query Interfaces for Casual End-Users*. PhD thesis. University of Zurich, Switzerland.
- Klein, D. and Manning, C. D. (2002): *Fast Exact Inference with a Factored Model for Natural Language Parsing*. Advances in Neural Information Processing Systems. 15: 3-10. Citeseer.

- Kobilarov, G., Scott, T., Raimond, Y., Oliver, S., Sizemore, C., Smethurst, M., Bizer, C. and Lee, R. (2009): *Media Meets Semantic Web --- How the BBC Uses DBpedia and Linked Data to Make Connections*. In Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M. and Simperl E., editors. In Proc. of the 6th European Semantic Web Conference, Heraklion, Greece. Springer LNCS 5554.
- Kwok, C., Etzioni, O. and Weld, D. (2001): *Scaling question answering to the Web*. In Proc. of the 10th International Conference on World Wide Web, p.150-161, Hong Kong, China. ACM.
- Lee, J. and Goodwin, R. (2005): *The semantic webscape: a view of the semantic web*. In Ellis, A. and Hagino, T., editors. In the Special interest tracks and posters of the 14th International Conference on World Wide Web, Chiba, Japan. ACM.
- Lehmann, J., Schüppel, J. and Auer, S. (2007): *Discovering unknown connections – the DBpedia relationship finder*. In Proc. of the 1st SABRE Conference on Social Semantic Web.
- Lei, Y., Uren, V. and Motta, E. (2006): *SemSearch: A Search Engine for the Semantic Web*. In Staab, S. and Svátek, V., editors. In Proc. of the 15th International Conference of Knowledge Engineering and Knowledge Management, EKAW, p.238-245, Podebrady, Czech Republic. Springer Verlag.
- Leinster, M. (1946): *A Logic Named Joe*. Astounding Science Fiction, March.
- Levenshtein, V. I. (1996): *Binary Codes Capable of Correcting Deletions, Insertions, and Reversals*. Soviet Physics – Doklady, 10, p.707-710.
- Levy A., Y., Srivastava D. and Kirk T. (1995): *Data Model and Query Evaluation in Global Information Systems*. Journal of Intelligence Information Systems, JIIS, 5(2): 121-143. Springer.
- Linckels S. and Meinel, C. (2005): *A Simple Solution for an Intelligent Librarian System*. In Proc. of the IADIS International Conference of Applied Computing, p.495-503.
- Linckels, S. and Meinel, C. (2006): *Resolving ambiguities in the Semantic Interpretation of Natural Language Questions*. In Proc. of the Intelligence Data Engineering and Automatic Learning, IDEAL, p.612-619, Burgos, Spain, LNCS Vol. 4224.
- Litkowski, K. C. (2001): *Syntactic Clues and Lexical Resources in Question-Answering*. Information Technology: The Ninth Text REtrieval Conference (TREC-9), NIST Special Publication 500-249, Gaithersburg, MD: National Institute of Standards and Technology.
- Lopez, V. and Motta, E. (2004): *Ontology Driven question answering in AquaLog*. In Proc. of the 9th International Conference on Applications of Natural Language to Information Systems (NLDB 2004), Manchester, UK, and published on the book AKT: Selected papers 2004.
- Lopez, V., Motta, E. and Pasin, M. (2005): *AquaLog: An Ontology portable Question Answering Interface for the Semantic Web*. In Gomez-Perez A., Euzenet J., editors. In Proc. of the European Semantic Web Conference, p.546-562, Crete, Greece. Springer LNCS 3532.
- Lopez, V., Motta, E. and Uren, (2006a): *PowerAqua: Fishing the Semantic Web*. In Sure, Y., Domingue, John, editors. In Proc. of the European Semantic Web Conference, p.393-410, Budva, Montenegro. Springer LNCS 4011.
- Lopez, V., Sabou, M. and Motta, E. (2006b): *PowerMap: Mapping the Real Semantic Web on the Fly*. In Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M. and Aroyo, L., editors. In Proc. of the 5th International Semantic Web Conference, p.414-427, Athens, USA. Springer LNCS. 4273.
- Lopez, V., Uren, V., Motta, E. and Pasin, M. (2007a): *AquaLog: An ontology-driven question answering system for organizational semantic intranets*. Journal of Web Semantics: Science Service and Agents on the World Wide Web, 5(2): 72-105.
- Lopez, V., Fernandez, M., Motta, E., Sabou, M. and Uren, V. (2007b): *Question Answering on the Real Semantic Web*. In the demo and poster session at the International Semantic Web Conference (ISWC 2007), Korea.
- Lopez, V., Motta, E., Džbor, M., D'Aquin, M., Peroni, S. and Guidi, D. (2009a): *Final Version of the Question Answering System*. OpenKnowledge Deliverable 8.6 (<http://www.openk.org/>).
- Lopez, V., Nikolov, A., Fernandez, M., Sabou, M., Uren, V. and Motta, E. (2009b): *Merging and Ranking answers in the Semantic Web: The Wisdom of Crowds*. In Yong, Y. and Ding, Y., editors. In Proc. of the Asian Semantic Web Conference, p.135-152, Shanghai, China. Springer Berlin Heidelberg.
- Lopez, V., Sabou, M., Uren, V. and Motta, E. (2009c): *Cross-Ontology Question Answering on the Semantic Web – an initial evaluation*. In Gil, Y. and Noy, N., editors. In Proc. of the Knowledge Capture Conference, p.17-24, California, USA. ACM.
- Lopez, V., Nikolov, A., Sabou, M., Uren, V., Motta, E. and d'Aquin, M. (2010): *Scaling up Question-Answering to Linked Data*. In Cimiano, P. and Pinto, S., editors. In Proc. of the 17th Knowledge Engineering and Knowledge Management by the Masses, p.193-210, Lisbon, Portugal. Springer Berlin.
- Lopez, V., Fernandez, M., Motta, E. and Stieler, N. (2011a): *PowerAqua: Supporting Users in Querying and Exploring the Semantic Web*. In the Semantic Web - Interoperability, Usability, Applicability. To appear.

- Lopez, V., Uren, V., Sabou, M. and Motta, E. (2011b): *Is Question Answering fit for the Semantic Web? A Survey*. In the Semantic Web - Interoperability, Usability, Applicability, 2(2): 125-155.
- Madhavan, J., Bernstein, P.A. and Rahm, E. (2001): *Generic schema matching with cupid*. The Very Large Databases Journal, p.49-58.
- Madhavan, J., Jeffery, S., Cohen, S., Dong, X., Ko, D., Yu, C., et al. (2007). *Web-scale data integration: You can only afford to Pay As You Go*. In Proc. of the 3rd Biennial Conference on Innovative Data Systems Research (CIDR), p. 342-350.
- Magnini B., Serafin L. and Speranza M. (2003): *Making Explicit the Semantics Hidden in Schema Models*. In Proc. of the Workshop on Human Language Technology for the Semantic Web and Web Services, held at ISWC-2003, Sanibel Island, Florida.
- Martin, P., Appelt, D. E., Grosz, B. J. and Pereira, F. (1985): *TEAM: An Experimental Transportable Natural-Language Interface*. IEEE Database Engineering. 8(3): 10-22.
- Mc Guinness, D. and van Harmelen, F. (2004): *OWL Web Ontology Language Overview*. W3C Recommendation 10 (2004) <http://www.w3.org/TR/owl-features/>.
- Mc Guinness, D. (2004): *Question Answering on the Semantic Web*. IEEE Intelligent Systems, 19(1): 82-85.
- McCool, R., Cowell, A. J. and Thurman, D. A. (2005). *End-User Evaluations of Semantic Web Technologies*. Workshop on End User Semantic Web Interaction. In Proc. of the International Semantic Web Conference. Springer LNCS.
- Meij, E., Mika, P. and Zaragoza, H. *Investigating the Demand Side of Semantic Search through Query Log Analysis*. (2009): In Proc. of the Workshop on Semantic Search at the 18th International World Wide Web Conference (WWW 2009), Madrid, Spain.
- Mena E., Kashyap V., Sheth A. and Illarramendi A. (2000): *OBSERVER: An approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies*. Distributed and Parallel Databases 8(2): 223-271.
- Mika, P. (2005): *Flink: SemanticWeb Technology for the Extraction and Analysis of Social Networks*. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 3(2). p.211-223. Elsevier.
- Miller, G. A. (1995): *WordNet: A Lexical Database for English*. Communications of the ACM, 38(11): 39-41.
- Minock, M., Olofsson, P. and Naslund, A. (2008): *Towards building robust Natural Language Interfaces to Databases*. In Proc. of the 13th international conference on Natural Language and Information Systems, London, UK.
- Minock, M. (2010): *C-Phrase: A System for Building Robust Natural Language Interfaces to Databases*. Journal of Data Engineering (DKE), 69(3): 290-302. Elsevier.
- Mithun, S., Kosseim, L. and Haarslev, V. (2007): *Resolving quantifier and number restriction to question OWL ontologies*. In Proc. of The First International Workshop on Question Answering at the International Conference on Semantics, Knowledge and Grid, Xi'an, China.
- Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Goodrum, R., Girju, R. and Rus, V. (1999): *LASSO: A Tool for Surfing the Answer Net*. In Proc. of the Text Retrieval Conference (TREC-8).
- Moldovan, D., Harabagiu, S, Girju, R., Morarescu, P., Lacatusu, F., Novischi, A., Badulescu, A and Bolohan, O. (2002): *LCC Tools for Question Answering*. In Proc. of the 11th Text Retrieval Conference TREC-11. NIST, Gaithersburg.
- Moldovan, D., Pasca, M., Harabagiu, S. and Surdeanu, M. (2003): *Performance issues and error analysis in an open-domain question answering system*. ACM Trans. Information Systems, 21(2): 133-154.
- Mollá, D. and Vicedo, J. L. (2007): *Question Answering in Restricted Domains: An Overview*. Computational Linguistics, 33(1): 41-61. MIT Press
- Motta, E. and Sabou, M. (2006): *Language technologies and the evolution of the semantic web*. In Proc. of the Fifth International Conference on Language Resources and Evaluation (LREC), Genoa, Italy. ELRA.
- Noy, N. and Musen, M. (2001): *Anchor-PROMPT: using non-local context for semantic matching*. In Proc. of the workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence (IJCAI), p.63-70.
- Oren, E., Delbru, R., Catasta, M., Cyganiak, R., Stenzhor, H. and Tummarello, G. (2008): *Sindice.com: A document-oriented lookup index for open linked data*. International Journal of Metadata, Semantics and Ontologies, 3(1): 37-52. Inderscience Publishers.
- Pasca M. (2003): *Open-Domain Question Answering from Large Text Collections*. CSLI Studies in Computational Linguistics, 29(4): 46. CSLI Publications.
- Pease, A., Niles, I. and Li, J. (2002). *The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications*. In the AAAI-2002 Workshop on Ontologies and the Semantic Web, Edmonton, Canada.
- Polleres, A., Hogan, A., Harth, A. and Decker, S. (2010): *Can we ever catch up with the Web?*. Journal of Semantic Web - Interoperability, Usability, Applicability, 1(1-2): 45-52. IOS Press.

- Popescu, A., M., Etzioni, O. and Kautz, H., A. (2003): *Towards a Theory of Natural Language Interfaces to Databases*. In Proc. of the International Conference on Intelligent User Interfaces, p. 149-157. ACM Press.
- Popov, B., Kiryakov, A., Kirilov, A., Mano, D., Ognyanoff and D., Goranov. (2004): *KIM – A semantic platform for information extraction and retrieval*. Natural Language Engineering, 10, p.375-392.
- Rahm E. and Bernstein P. A. (2001): *A survey of approaches to automatic schema matching*. The International Journal on Very Large Data Bases 10(4): 334-350.
- Resnik P. (1995): *Disambiguating noun grouping with respect to WordNet senses*. In Proc. of the 3rd Workshop on very Large Corpora. MIT, p.54-68.
- Sabou, M., d'Aquin, M. and Motta, E. (2006a): *Using the Semantic Web as Background Knowledge for Ontology Mapping*. In Proc. of the 1st International Workshop on Ontology Matching.
- Sabou, M., Lopez, V. and Motta, E. (2006b): *Ontology Selection on the Real Semantic Web: How to Cover the Queens Birthday Dinner?*. In Proc. of the Knowledge Acquisition Workshop at the Knowledge Engineering and Knowledge Management Conference (EKAW).
- Sabou, M., Gracia, J., Angeletou, S., d'Aquin, M. and Motta, E. (2007): *Evaluating the Semantic Web: A Task-based Approach*. In Aberer, K., Choi, K-S., Noy, N., Allemang, D., Lee, K., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudre-Mauroux, P., editors. In Proc. of the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference, p. 423-437, Korea. Springer LNCS 4825.
- Sabou, M., d'Aquin, M. and Motta, E. (2008): *Exploring the Semantic Web as Background Knowledge for Ontology Matching*. Journal of Data Semantics, 11, p.156-190.
- Schraefel, M.C., Shadbolt, N., Gibbins, N., Glaser, H. and Harris, S. (2004): *CS AKTive Space: Representing Computer Science in the Semantic Web*. In Feldman, S., Uretsky, M., Najork, M., Wills, C., editors. In Proc. of the International World Wide Web Conference, p.384-392, New York, USA. ACM Press.
- Shvaiko, P. and Euzenat, J. (2005): *A Survey of Schema-Based Matching Approaches*. Journal of Data Semantics IV, p.146-171.
- Srihari, K., Li, W. and Li, X. (2004): *Information Extraction Supported Question- Answering*. In Advances in Open-Domain Question Answering. Kluwer Academic Publishers.
- Staab, S. and Studer, R. (2004): *Handbook of Ontologies*. Springer, Berlin / Heidelberg / New York.
- Stuckenschmidt, H., van Harmelen, F., Serafini, L., Bouquet, P. and Giunchiglia, F. (2004): *Using C-OWL for the Alignment and Merging of Medical Ontologies*. In Proc. of the First International Workshop on Formal Biomedical K.R. (KRMed).
- Suchanek, F., Kasneci, G. and Weikum, G. (2008): *YAGO: A Large Ontology from Wikipedia and WordNet*. Web Semantics: Science, Services and Agents on the World Wide Web, 6(3): 203-217.
- Sure, Y. and Iosif, V. (2002): *First Results of a Semantic Web Technologies Evaluation*. Common Industry Program at the federated event: ODBASE'02 Ontologies, Databases and Applied Semantics. California, Irvine.
- Surowiecki J. (2004): *The Wisdom of Crowds*, Doubleday, New York.
- Tablan, V., Damjanovic, D. and Bontcheva, K. (2008): *A Natural Language Query Interface to Structured Information*. In Bechhofer, S., Hauswirth, M., Hoffmann, J. and Koubarakis, M., editors. In Proc. of the 5th European Semantic Web Conference, p.361-375, Tenerife, Spain. Springer Verlag.
- Tang, L. R. and Mooney, R. J. (2001): *Using multiple clause constructors in inductive logic programming for semantic parsing*. In Raedt, L., Flanch, P., editors. In Proc. of the 12th European Conference on Machine Learning (ECML-2001), p.466-477, Germany. Springer.
- Tartir, S. and Arpinar, I. B. (2010): *Question Answering in Linked Data for Scientific Exploration*. In the 2nd Annual Web Science Conference. Raleigh, North Carolina, USA. ACM.
- Thompson, C. W., Pazandak, P. and Tennant, H. R. (2005): *Talk to Your Semantic Web*. IEEE Internet Computing, 9(6): 75-78. IEEE Computer Society.
- Tran, T., Cimiano, P., Rudolph, S. and Studer, R. (2007): *Ontology-based interpretation of Keywords for Semantic Search*. In Aberer, K., Choi, K. S., Noy, N., Allemang, D., Lee, K. I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G. and Cudré-Mauroux, P., editors. In Proc. of the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference, p.523-536, Busan, Korea. Springer LNCS, Vol. 4825.
- Tummarello, G., Cyganiak, R., Catasta, M., Danielczyk, S., Delbru, R. and Decker, S. (2010): *Sig.ma: Live views on the Web of Data*. In Rappa, M., Jones, P., Freire, J. and Chakrabarti, S., editors. In Proc. World Wide Web Conference (WWW-2010), 8(4): 1301-1304, Raleigh, USA. Elsevier B.V.
- Uren, V., Lei, Y., Lopez, V., Liu, H., Motta, E. and Giordanino, M. (2007): *The usability of semantic search tools: A review*, Knowledge Engineering Review, 22 (4): 361-377.

- Uren, V., Sabou, M., Motta, E., Fernandez, M., Lopez, V. and Lei, Y. (2010): *Reflections on five years of evaluating semantic search systems*. In the International Journal of Metadata, Semantics and Ontologies. 5(2): 87-98. Inderscience Publishers.
- Van Hage, W., Katrenko, S. and Schreiber, G. (2005): *A method to Combine Linguistic Ontology-Mapping Techniques*. In Gil, Y., Motta, E., Benajmins R., Musen, M., editors. In Proc. of International Semantic Web Conference, p.732-744. Springer LNCS Vol. 3729.
- Wang, C., Xiong, M., Zhou, Q. and Yu, Y. (2007): *PANTO: A portable Natural Language Interface to Ontologies*. In Franconi, E., Kifer, M., May, W., editors. In Proc. of the 4th European Semantic Web Conference, p.473-487, Innsbruck, Austria. Springer Verlag.
- Wang, H., Tran, T., Haase, P., Penin, T., Liu, Q., Fu, L. and Yu, Y. (2008): *SearchWebDB: Searching the Billion Triples!*. Billion Triple Challenge 2008 at the International Semantic Web Conference, Karlsruhe, Germany.
- Winkler, W. (1999): *The state of record linkage and current research problems*. US Bureau of the Census Technical Report RR99/04.
- Woods, W. (1973): *Progress in natural language understanding - an application to lunar geology*. In Proc. of the American Federation of Information Processing Societies (AFIPS), 42: 441-450. AFIPS Press.
- Wrigley, S.N., Elbedweihy K., Reinhard, D., Bernstein, A. and Ciravegna, F. (2010): *Evaluating semantic search tools using the SEALS Platform*. In the International Workshop on Evaluation of Semantic Technologies (IWEST 2010) at the International Semantic Web Conference, China.
- Wu, M., Zheng, X., Duan, M., Liu, T. and Strzalkowski, T. (2003): *Question Answering by Pattern Matching, Web-Proofing, Semantic Form Proofing*. NIST Special Publication: The Twelfth Text REtrieval Conference (TREC), p. 500-255.
- Wu Z. and Palmer, M. (2004): *Verb Semantics and Lexical Selection*. In Proc. of the 32nd Annual Meeting of the Associations for Computational Linguistics, p.133-138, Las Cruces, New Mexico.
- Zenz, G., Zhou, X., Minack, E., Siberski, W. and Nejd, W. (2009): *From Keywords to Semantic Queries—Incremental Query Construction on the Semantic Web*. In the Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 7(3): 166-176. Elsevier.

Appendix A

A.1 Example of NL queries by topics

Currently, PowerAqua covers a wide range of topics, such as:

- **Movies and actors:** List me all films with Brad Pitt and Angelina Jolie, Who stars in Bruce Almighty?, Give me movies about Buenos Aires, etc.
- **Books:** Which books Stephen King wrote?, Where was Franz Kafka born?, etc.
- **Geography, airports:** How many airports exist in Canada?, Which sea is close to Volgograd?, Which airlines fly from London Luton to Paris Orly?, What state is Salem the capital of?, Which countries are members of the EU?, etc.
- **Languages:** How many languages are spoken in Afghanistan?, Give me languages in Islamic countries, In which region is Cantonese spoken?, etc.
- **Famous people:** Which prizes have been won by Laura Linney, Who are the husbands of Ava Gardner and Mia Farrow?, What is Steve Pizzati known for?, etc.
- **Universities:** Which organization employs Enrico Motta?, Which universities are in Pennsylvania?, What are the publications of Marta Sabou or Frank Van Harmelen?, etc.
- **Television, music, sports:** Television shows created by Walt Disney, Who presents Top Gear?, Who are the members of Prodigy?, Give me tennis players in France, etc.
- **Animals:** Describe the habitat for beavers, What is the diet of the manatee?, Give me types of birds, Which animals are reptiles?, etc.
- **Companies, organizations:** Which organizations participate in humanitarian aid?, What is the revenue of IBM?, Give me oil industries in Russia, Which RBA banks are situated in Switzerland?, etc.
- **Food, drinks, restaurants:** List all Mexican popular dishes, Give me Thai restaurants in Alameda, etc.
- **Diseases, medicine:** What diseases has symptoms of hair loss?, What enzymes are activated by adrenaline?, What are the symptoms and treatments for Parkinson?, Which drugs are popular for weight gain?, How to erase a scar?, etc.
- **Religion:** Which religion does Easter belong to?, Which are the fasting periods in Islam?, Who believe in the apocalypse?, etc.
- **Natural Disasters, Terrorism:** What are the terrorist organizations that are active in Spain? , Where do earthquakes occur?, Which are the main attacks that took place in the United Kingdom?, etc.
- **Cultural, Arts:** Who is the engineer of the Eiffel tower?, Name all bridges in Paris that are crossing the Seine, etc.
- **Cars, Jobs:** Give me all Toyota cars that are a convertible, What jobs does lcs recruit for?, What is the apex lab?, etc.

Appendix B

B.1 PowerAqua configuration

PowerAqua runs under Apache Tomcat and Java 1.5 or above. PowerAqua is based on a plug-in mechanism that allows the user to query ontologies in different knowledge representation languages and ontology platforms at run time. At start up PowerAqua loads all plugin files. Given a user query, each ontology used by PowerAqua is then assigned and queried through its own instantiation of the respective plug-in (initialized on demand). All plugins implement the generic interface defined in the class *OntologyPlugin*. The system administrator can configure PowerAqua through the following configuration files:

multi_index_properties.xml: the administrator can specify whether the Watson search engine, Virtuoso or PowerMap's own list of indexed ontologies are used (true), or not (false), as potential sources to answer user queries. The end user can also specify the optional use of the Watson Search Engine through the Web interface. If PowerMap is used, in this file the administrator can specify the location of the metadata table, which links each ontology with its index, and of the global location of the Lucene indexes

service_properties.xml (Figure B.1): it lists the location of all the ontologies (or graphs) to be used by PowerAqua (apart from the ones accessed through the Watson search engine), naming their respective plug-ins (and proxy information if required). There is an entry for each ontology (<REPOSITORY>). Also, at the beginning of the file, there is an entry (<PLUGIN_MANAGER>) to specify the global location of the different plugins.

index_properties.xml: it lists the name of all Lucene indexes and the location (local or remote) of the metadata tables (with login information if required). There is one metadata table per Lucene index, and each Lucene index indexes 1 or many ontologies. Metadata tables are created offline, at the same time as the indexes.

```

1. <CONFIGURATION>
2. <PLUGIN_MANAGER>/Applications/apache-tomcat-5.5.23/webapps/poweraqua/WEB-INF/aquaplugins</PLUGIN_MANAGER>
3. <REPOSITORY>
4.     <SERVER>http://kmi-web07.open.ac.uk:8080/sesame</SERVER>
5. <PROXY>wwwcache.open.ac.uk</PROXY>
6. <PORT>80</PORT>
7. <LOGIN></LOGIN><PASSWORD></PASSWORD>
8. <PLUGIN_TYPE>sesame</PLUGIN_TYPE>
9. <REPOSITORY_NAME>TAP</REPOSITORY_NAME>
10. <TYPE>RDF</TYPE>
11. </REPOSITORY>

```

Figure B.1. An example of the service_properties.xml file

PowerAqua also contains the following files and resources:

WordNet files and dictionaries: the location and version of the WordNet dictionaries is specified in the file_properties.xml file. The version currently in use is 3.0 for Mac and Linux platforms.

query_properties.xml : used by PowerAqua to associate linguistic terms to *wh-clauses* like “who/when/where”. For instance “who” is equivalent to “person, organization”, while “where” is translated into “location, state”.

jape_grammars (*.jape) and GATE files: used by the GATE platform and the Linguistic Component, they do not need to be modified unless the grammars are extended. Currently, PowerAqua uses GATE version 3 (<http://www.gate.ac.uk/>).

B.2 PowerAqua indexing mechanism

PowerMap provides indexing services to ontologies stored in platforms such as Sesame, which do not provide full text searches. The use of this off-line indexing engine (based on Lucene inverted indexes), together with database storage technologies, allows PowerAqua to scale and respond to searches quickly (in real time). It provides the following functionalities:

- Exact and approximate full text searches over ontological entities and literals: to find the candidate matches for a query term. It uses the localname and label, and optionally, other additional information that identifies the entity, i.e. alternative names. In the case of compound names, partial or incomplete mappings should not be returned as hits, e.g.:

“Enrico Franconi” is not a good mapping for “Enrico Motta”. The resultant entity hits should provide the following information: URI, label(s), type (class, instance, property or literal) and the identifier of the ontology they belong to (plugin ID). In the case of literals, PowerAqua requires to know the URI of the instance where the literal appears.

- Ranking of hits across indexes: ranking is necessary to limit the total number of candidate mappings across sources and within the same source. Ranking is based on setting up adequate thresholds over Lucene searches to obtain a good compromise between performance and recall.
- Indexes optimization: PowerMap indexing module creates separate Lucene indexes for the schema (classes, properties) and the KB (instances, literals). This allows us to optimize searches in some particular cases, e.g., the search of WordNet hypernyms (which normally relate to classes, not instances) can be restricted to the schema only.
- Fast access to metadata: PowerAqua semantic algorithms require to know all superclasses, subclasses and type of the entities returned as candidate matches. This taxonomical information is used to filter and compute the semantic meaning of hits. As the number of candidate hits can be high, this key information about the entities should be easily accessible for fast retrieval. PowerAqua indexing services create *mysql* metadata tables at indexing time to store all taxonomical information for each entity indexed, so that this information can be retrieved quickly, while at the same time the indexes are kept small (containing only the relevant information to locate and identify the entities).
- Multiple indexing: the generation and modification of indexes is allowed at different times.

The indexing module creates new indexes for all semantic sources specified by the administrator in the xml configuration files.