

University of Bremen
Faculty of Mathematics and Computer Science

Doctor of Engineering Dissertation

Semantic Interaction in Web-based Retrieval Systems:

**Adopting Semantic Web Technologies and Social Networking
Paradigms for Interacting with Semi-structured Web Data.**

by

Hidir Aras

Supervisor: Prof. Dr. Rainer Malaka

2nd Supervisor: Prof. Dr. Iryna Gurevych

January, 2017

Dedicated to Lare and Roan

Contents

| | |
|---|-----------|
| Contents | i |
| I Foundations | 3 |
| 1 Introduction | 5 |
| 1.1 Semantic Interaction with the Web | 6 |
| 1.2 Aim and Research Questions | 8 |
| 1.3 Structure of the Thesis | 10 |
| 1.4 Publications | 11 |
| 2 Basics | 13 |
| 2.1 Web Basics | 13 |
| 2.1.1 Characteristics of Web Data | 14 |
| 2.1.2 Data Representation Models | 17 |
| 2.2 Knowledge Representation | 24 |
| 2.2.1 RDF | 25 |
| 2.2.2 Ontologies | 28 |
| 2.3 Emergent Semantics | 32 |
| 2.3.1 Social Networks and Folksonomies | 32 |
| 2.4 Metrics for Semantic Analysis | 36 |
| 2.4.1 Basic Metrics | 36 |
| 2.4.2 Graph-based Metrics | 37 |
| 2.4.3 Semantic Analysis of Folksonomies | 38 |
| 2.4.4 Other Approaches | 40 |
| 2.5 Machine Learning | 40 |
| 2.5.1 Supervised Learning | 41 |
| 2.5.2 Unsupervised Learning | 42 |
| 2.5.3 Instance-based Learning | 44 |

| | | |
|-----------|---|------------|
| 2.6 | Evaluation Methods for IR | 44 |
| 2.7 | Summary | 46 |
| 3 | Semantic Retrieval and Visual Exploration | 49 |
| 3.1 | Information Retrieval Models | 50 |
| 3.1.1 | Information Seeking | 51 |
| 3.1.2 | Browsing vs. Search | 52 |
| 3.2 | Structured Data Extraction | 53 |
| 3.2.1 | Supervised Wrapping | 56 |
| 3.2.2 | Unsupervised Wrapping | 58 |
| 3.2.3 | Discussion | 60 |
| 3.3 | Semantic Layering | 60 |
| 3.3.1 | Ontology-based Annotation | 61 |
| 3.3.2 | Social Semantic Tagging | 62 |
| 3.3.3 | Exploiting Tag Semantics for Retrieval | 63 |
| 3.4 | Visual Information Exploration | 64 |
| 3.4.1 | 2D Semantic Retrieval Interfaces | 66 |
| 3.4.2 | Retrieval Interfaces for Folksonomy Systems | 67 |
| 3.4.3 | Visual Exploration in 3D | 70 |
| 3.4.4 | Search UI Concepts and Information Bias | 72 |
| 3.5 | Conclusion | 74 |
| 4 | Semantic Interaction in Web Retrieval | 79 |
| 4.1 | Interaction Metaphors | 79 |
| 4.2 | Semantic Interaction Tasks in IR | 82 |
| 4.2.1 | Task 1: Semantic Layering | 83 |
| 4.2.2 | Task 2: Semantic Mediation | 84 |
| 4.2.3 | Task 3: Semantic HCI | 87 |
| 4.3 | SI Framework for Web IR | 88 |
| 4.4 | SI with Wrappers in a Dialogue System | 91 |
| 4.5 | SI in 2D/3D Visual Retrieval Interfaces | 95 |
| 4.6 | Summary and Roadmap | 99 |
| II | Case Studies and Applications | 101 |
| 5 | Collaborative Semantic Layering | 103 |
| 5.1 | Semantic Layering of Structured Data | 104 |
| 5.1.1 | State of the Art | 105 |
| 5.1.2 | tag2Wrap - Workflow | 107 |

| | | |
|----------|---|------------|
| 5.1.3 | Design and Tagging User Interface | 110 |
| 5.1.4 | Consolidation of Tag Structures | 112 |
| 5.1.5 | Evaluation and Results | 114 |
| 5.1.6 | Conclusion | 118 |
| 5.2 | Semantic Layering with HC | 118 |
| 5.2.1 | Related Work | 119 |
| 5.2.2 | Playing and Tagging using Binary Verification | 123 |
| 5.2.3 | Webpardy: Harvesting QA by HC | 126 |
| 5.2.4 | Evaluation and Results | 134 |
| 5.3 | Conclusion | 139 |
| 6 | Semantic QA in a Dialogue System | 141 |
| 6.1 | Background: The SmartWeb project | 143 |
| 6.1.1 | System Architecture and Dialogue Interaction | 144 |
| 6.1.2 | Speech Recognition and Semantic Queries | 146 |
| 6.1.3 | Knowledge Representation and Ontology | 147 |
| 6.2 | Semantic Wrapper Agents for QA | 150 |
| 6.2.1 | State of the Art | 150 |
| 6.3 | Semantic Wrapper Generation | 154 |
| 6.3.1 | Introduction | 154 |
| 6.3.2 | State of the Art | 155 |
| 6.3.3 | Visual Interaction for Creating Wrappers | 156 |
| 6.3.4 | JEFF - System Design | 174 |
| 6.3.5 | Evaluation and Results | 178 |
| 6.3.6 | Conclusion | 184 |
| 6.4 | Semantic Transformation | 187 |
| 6.4.1 | State of the Art | 187 |
| 6.4.2 | Approach | 188 |
| 6.4.3 | Data Representation: Input/Output | 189 |
| 6.4.4 | Transformation Engine *2RDF | 190 |
| 6.4.5 | Conclusion | 195 |
| 6.5 | Semantic Scoring of (Q,A)-Pairs | 197 |
| 6.5.1 | State of the Art | 198 |
| 6.5.2 | Approach | 199 |
| 6.5.3 | Evaluation | 202 |
| 6.5.4 | Conclusion | 207 |
| 6.6 | Question-Answering Workflow and Deployment | 209 |
| 6.6.1 | Deployment in SmartWeb | 210 |
| 6.7 | Conclusion | 215 |

| | |
|---|------------|
| 7 Semantic HCI for Visual Exploration | 219 |
| 7.1 Case Study: Semantic Tag Cloud | 222 |
| 7.1.1 State of the Art | 223 |
| 7.1.2 Approach | 224 |
| 7.1.3 Semantic Cloud User Interface | 231 |
| 7.1.4 User Study | 234 |
| 7.1.5 Results | 236 |
| 7.1.6 Conclusion | 237 |
| 7.2 Case Study: Semantic Browsing in 3D | 239 |
| 7.2.1 State of the Art | 240 |
| 7.2.2 Approach | 240 |
| 7.2.3 User Study | 248 |
| 7.2.4 Results | 252 |
| 7.2.5 Conclusion | 257 |
| 7.3 Case Study: Navigating Large Information Spaces | 259 |
| 7.3.1 State of the Art | 259 |
| 7.3.2 Approach | 260 |
| 7.3.3 User Study | 262 |
| 7.3.4 Conclusion | 265 |
| 7.4 Conclusion | 266 |
| 8 Conclusions | 269 |
| 8.1 Aim and Contribution | 270 |
| 8.1.1 Collaborative Semantic Layering | 270 |
| 8.1.2 Semantic QA in a Dialogue System | 272 |
| 8.1.3 Semantic HCI for Exploring Information Spaces | 274 |
| 8.2 Outlook and Future Work | 277 |
| Bibliography | 279 |
| A Appendix of Chapter 5 | 303 |
| A.1 Evaluation of the Tagging Method | 303 |
| B Appendix of Chapter 6 | 305 |
| B.1 Example HTML Page: Research Group Digital Media News. | 305 |
| B.2 Sample Structure for the Semantic Class "footballMatch" . . | 306 |
| B.3 Visual Wrapper Tool JEFF | 307 |
| B.4 Rule Naming Conventions | 307 |
| B.5 SeRQL query example | 308 |
| B.6 Sample s2rdf Rule File (.s2r) for the FIFA corpus. | 309 |

| | |
|--|------------|
| B.7 ProperScore Evaluation | 311 |
| B.8 Example of a (q,a) pair in RDF | 315 |
| C Appendix of Chapter 7 | 317 |
| C.1 Semantic Cloud Evaluation | 317 |
| List of Abbreviations | 319 |
| List of Figures | 321 |
| List of Tables | 324 |

Acknowledgements

First, I want to thank my family for their support, patience and encouragement during the time of writing this thesis.

Next, I want to thank my supervisor Rainer Malaka for his support and confidence over the years and his vital enthusiasm for research. I would also like to thank my second supervisor Iryna Gurevych. I also want to thank my colleagues H.-P. Zorn, R. Porzel and C. Pretzsch for their ideas and contributions in the SmartWeb project. Last but not least my other colleagues at the European Media Lab, the University of Bremen and my students for their contributions in experiments and case studies.

I also want to thank the KTS and the BMBF for sponsoring and funding the SmartWeb project.

Part I

Foundations

Chapter 1

Introduction

When the World Wide Web was born in the early 1990s allowing access to static hypertext documents worldwide, only few people understood its immense potential impact for new forms of communication, business, and use of knowledge. Since then, the web has evolved into a global information space which has passed through several phases from published static documents to dynamic interactive sites, web services, the ‘social’ web, and other enhanced applications and services for storing, exchanging, and processing various types and large amounts of information.

Due to the decentralized nature and former presentational purpose of the web, published contents and related data structures are heterogeneous and lack semantic information. It is generally difficult to automatically understand the meaning of the underlying information structures, as well as interlinking them or interacting with them in a reasonable manner without high intellectual effort.

An initial serious answer targeting the elimination of this so-called “semantics bottleneck” of the web – i.e. adding meaning to information encoded in web pages – was given by the Semantic Web initiative and a vision formulated by Tim Berners-Lee in 2001¹. However, adopting the “semantic” Web on a large scale proved difficult, given that many of the proposed technologies – such as creating knowledge representations – require expert knowledge and are time-consuming, difficult to maintain, and costly.

It is precisely for this reason, that popular social networking or crowd-based applications provide a new source for gathering semantic information and meta-data from users in the form of user-assigned tags attached to

¹ “bringing meaning to the Web making it possible for machines, i.e. software agents to understand and reason about data that is distributed and available worldwide” (Berners-Lee et al., 2001)

user-generated content. The user-created vocabularies (folksonomies) which result from a social or collaborative tagging process based on the *Wisdom of the Crowds* principle, allow for emergent semantics to be extracted, e.g. concepts, by statistically analyzing tags and their relations.

Similar forms of collective intelligence has been employed systematically in Crowdsourcing and Human Computation (HC) systems which are designed to exploit human intelligence and skills for (re)solving computationally complex tasks, e.g. image recognition and understanding, semantic labeling, classification, etc. It has been shown that these tasks can also be embedded as part of the playing strategy of a computer game, as pioneered by Von Ahn et al. (2008) in the “Games With A Purpose” approach.

In general, the semantic enrichment or transformation pipeline for the web can be described as a process that turns data into knowledge the users can interact with, or explore in order to fulfill their goals, e.g. finding relevant answers to specific questions. In the best case, the process results in an increase of valuable information for the users, while much simpler motivations such as fun or entertainment can be imagined as well.

1.1 Semantic Interaction with the Web

The vast majority of web documents are created by tools that do not allow for adding formal and explicit semantics to information or if so, only in a restricted form. For this reason, existing information needs to be transformed to a semantic representation applying increasingly automatic methods. A widely adopted technique for this task is information extraction which can be integrated with methods for semantic layering, e.g. using controlled vocabularies, knowledge bases, etc.

For example, simple data extraction techniques have been used in shop bots that extract and evaluate product data for the purpose of price comparison (Doorenbos et al., 1997). The *DBpedia*² example shows the great benefit of applying methods of structured information extraction for generating linked data and semantic knowledge bases from WikiPedia³ sources.

However, complex user-machine interactions beyond search and retrieval of web documents – such as question answering – have higher requirements for extracting and analyzing web data. Therefore, relevant and valuable information structures must be identified, extracted, transformed, and relevant answers returned to the users of the system nearly at query time.

²<http://dbpedia.org/page/Bremen>

³<https://de.wikipedia.org/wiki/Bremen>

Question answering systems play an important role in enhanced information systems such as dialogue systems and hereby, answering the user's natural language questions from relevant information in web documents was identified as an important challenge.

Natural language questions such as "Who won the federal election in Germany?" or "Who scored for Bayern Munich against Inter Mailand in the first half?", etc. require the identification and extraction of facts found in web pages on-the-fly. If possible, a question answering system must also deal with follow-up questions, which could either be answered through inference based on formal semantics by e.g. using ontologies or via crowd-based approaches exploiting collective intelligence.

Furthermore, the vast amounts of data on the web produced thus far confronts research and industry with the famous "needle in a haystack" problem, which web search engines try to resolve by employing statistical methods and by exploiting the web hyperlink structure (Brin and Page, 1998). Most of the time, the users have no transparent insight into how the returned results have been obtained and ranked. Another fact is that users are only able to interact with a minimal percentage of the available information directly and via limited "non-semantic" retrieval interfaces based on the old classic four-phased framework or a variation of this approach (Shneiderman et al., 1997). As a consequence, there is an urgent need for extending the existing models and researching new forms of "semantic" interaction approaches increasingly from a human-computer interaction (HCI) perspective. Therefore, new methods, modalities, and interaction metaphors, e.g. based on direct manipulation, must be explored in order to enhance user experience for searching and exploring web content efficiently.

Frohlich (1997) reaffirms the fact that improving the graphical representation for directly interacting with an interface is not sufficient, stating that the representation of the results and the interaction itself must be "meaningful and accurate" to users. In particular, in scenarios for searching and exploring personally relevant information, retrieval and search systems could benefit from semantic information obtained from a crowd or social network of users.

Held and Cress (2008), Sheth and Ramakrishnan (2007) showed that the interaction behavior of users relies on associative semantic network models. For example, visual retrieval interfaces could appropriately integrate and exploit associative semantics derived from a community-driven approach. Hence, there is a need to revisit and enhance existing interaction metaphors and develop new "semantic" interaction metaphors that can be implemented

in enhanced user interfaces for retrieval, considering the user's background knowledge as well as cognitive aspects.

1.2 Aim and Research Questions

The aim of this thesis is to investigate new forms of interaction with web data based on semantic information in use case scenarios of web retrieval. Therefore, an inherently dynamic view beyond single concepts and models from semantic information processing, information extraction and human-computer interaction is adopted.

The described research focuses on two use case scenarios; semantic question-answering and semantic exploration of information spaces on the web employing 2D/3D visual retrieval interfaces. The main research questions are summarized as follows:

1. Is collaborative tagging a reasonable approach for adding semantics to structured data or information fragments in a web page? How can the human computation paradigm be exploited for creating semantic annotations for information structures in web pages?
2. How can semantic wrapping technology be employed for answering natural language questions of users by extracting the answers from semi-structured web pages at query time? How can an agent-based semantic wrapper system be deployed and integrated into a question answering pipeline in a complex real-world dialogue system?
3. How can associative semantics from folksonomies and dedicated semantic interaction metaphors be employed in 2D/3D visual retrieval interfaces in order to explore and navigate (large) information spaces on the web?

In order to provide a basis for resolving the listed challenges, fundamental concepts, methods, and a semantic interaction (SI) framework are elaborated on in Chapter 4. Based on these, enhanced tasks for semantic layering, retrieval/extraction and human-computer interaction (S-HCI) for the focused use cases are defined and investigated.

Collaborative Semantic Layering

The first case studies focus on researching social web applications and human-computation games (Chapter 5) for creating semantic annotations

for structured data or informative fragments in web pages employing a social network or crowd of users.

An initial contribution in this regard is the application of a visual annotation method for creating conceptual structures utilizing a collaborative tagging environment. It will be shown that it is possible to generate appropriate semantic annotation structures, while challenges exist due to the influence of the selection metaphor used on tagging behavior, the consolidation method for disambiguating conceptual tags and the problem of user motivation for contributing to the collaborative tagging process.

In follow up research game-centric approaches employing the human computation paradigm for annotating information structures and informative fragments in web pages are investigated. In an initial proof of concept study it will be shown that semantic annotation tasks can be integrated successfully into an HC game based on the concept of binary verification.

In the second case study, the HC paradigm will be applied for the task of question answering. The Webpardy game – based on the famous TV quiz – helps to associate questions with web fragments in order to collect pairs of questions and resources/answers to be exploited in question answering applications.

Both case studies will show that the tripartite model of collaborative tagging (user-tags-resources) can be successfully employed in human computation games for semantic annotation.

Semantic QA in a Dialogue System

The case study presented in Chapter 6 aims at researching and implementing a workflow and appropriate methods for answering natural language question of users from semi-structured web documents in the context of a knowledge-based real-world dialogue system.

Therefore, how to employ a semi-automatic visual wrapper generation approach for semantic labeling, extracting and semantic querying of complex information structures from web pages is investigated. Herewith, semantic access to the online state of frequently changing web pages can be realized at query time, which is a fundamental requirement for web information extraction systems in order to be deployed in a real world dialogue system. Moreover, as dialogue systems use a conversational metaphor for realizing user-machine interaction, a semantic question answering pipeline based on an expert-created complex ontology is employed.

Semantic HCI for Exploring Information Spaces

The case studies in Chapter 7 investigate the question of how implicit semantics, e.g. semantic relatedness of user tags and dedicated semantic interaction metaphors, e.g. based on direct manipulation can be employed in order to enhance the (semantic) interaction capabilities of web-based (visual) retrieval interfaces, for exploring resources in a folksonomy system.

First, new metaphors for semantic arrangement and visualization in 2D/3D are examined. Second, enhanced interaction metaphors for semantic exploration of search results, e.g. based on direct manipulation are explored.

The Semantic Cloud study investigates the task for semantic exploration of the folksonomy tag space and related resources employing a hierarchically organized tag cloud metaphor as a visual retrieval interface.

The second case study investigates the task of semantic exploration of results of folksonomy retrieval systems by exploiting relatedness of tags in 3D visualizations and for navigation in 3D space. Therefore, the first person shooter perspective based on the direct manipulation metaphor is employed.

In order to enable an efficient and immersive navigation for large 3D semantic information spaces, a navigation technique based on direct manipulation that scales well from low distance interaction to navigating large distances using continuous gestures for mouse and touch input with visual feedback on direction and speed is researched and evaluated.

1.3 Structure of the Thesis

In Chapter 1 the thesis motivation, aims and research questions are described, in addition to the structure of the thesis.

Chapter 2 introduces basic concepts and models of the Web, knowledge representation models, methods and metrics for semantic analysis of data, machine learning and retrieval evaluation. Chapter 3, then, introduces the basics of information seeking, web information extraction and presents approaches for semantic layering of data and state of the art of 2D and 3D visual retrieval interfaces that exploit semantic information to some extent.

In Chapter 4, a conceptual framework for enabling semantic interaction in web-based retrieval systems is described. The introduced framework allows for implementing and integrating the basic building blocks for exploring and answering the principal research questions of the thesis.

In Chapter 5, alternatives to expert-based approaches for labeling semi-structured data on web pages based on collaborative tagging and human

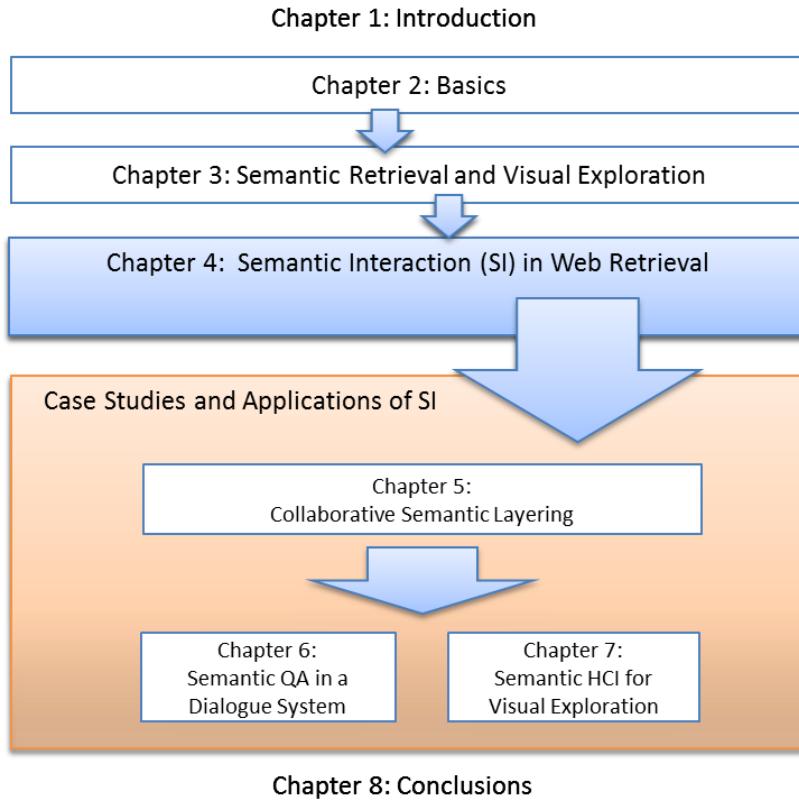


Figure 1.1: Thesis structure.

computation online games are explored in short studies via prototypes.

In Chapter 6 an ontology-based semantic layering and wrapping approach is researched and deployed in the SmartWeb multi-modal dialogue system, allowing natural language questions to be posed to the semi-structured web.

Chapter 7 focuses on visual retrieval interfaces for exploring data in folksonomy systems via suitable 2D/3D visualization and navigation metaphors and exploiting implicit semantics from folksonomies based on the concepts defined in Chapter 4.

The thesis concludes with Chapter 8, describing its core contributions and potential future work.

1.4 Publications

The research presented is partly a result of collaborations with colleagues in research projects, colleagues at the University of Bremen and some of my

students whose theses I supervised. In the following I would like to honor their contributions appropriately:

The work described in 5 was published at the IADIS WWW/Internet Conference in 2009 (Aras et al., 2009). The students Wenyu Cai and Julian Wiersbitzki contributed to the implementation of the tagging user interface and the consolidation method and its evaluation.

The FastTag idea resulted from a collaboration with Markus Krause and was published at the WWW Conference in 2009 (Krause and Aras, 2009). The Webparty Human Computation approach was developed in collaboration with my colleague Markus Krause as well and the student Andreas Haller, who implemented the game concept. The work was published as a paper at the KDD HCOMP Workshop in 2010 (Aras et al., 2010).

The Semantic Cloud interface described in Section 7.1 was implemented by Sandra Siegel in her bachelor's thesis which I supervised in 2009. The results of the Semantic Cloud project resulted in a joint paper published at the Visual Interfaces for the Social and Semantic Web Workshop at the IUI'2010 Conference⁴ in Hong Kong (Aras et al., 2010). Sandra Siegel was awarded the CONTACT Software Award⁵ for her bachelor's thesis in 2009.

The user interface of the 3D semantic browsing approach (Section 7.2) was implemented by Alena Penner in her diploma thesis, which I supervised in 2009.

The concept of ElasticSteer presented in Section 7.3 was implemented in a prototype based on the 3D semantic browser re-implementation by Patrick Rodacker in his master's thesis. The work was developed in collaboration with my colleagues Benjamin Walther-Franks and Marc Herrlich and published at the Smart Graphics Workshop in 2011(Aras et al., 2011). The ideas researched in the ElasticSteer project and the Semantic Space Browser were also published partly at the CHI conference in 2012 (Döring et al., 2012).

⁴<http://ceur-ws.org/Vol-565/>

⁵<http://www.contact-software.com/de/engagement/contact-software-foerderpreis.html>

Chapter 2

Basics

This chapter introduces the basic concepts behind the World Wide Web, knowledge representation models for encoding web data, metrics and methods of semantic analysis, relevant machine learning algorithms and standard evaluation methods of information retrieval.

2.1 Web Basics

Web information sources in the context of this thesis may be classified into three categories; structured, semi-structured and unstructured. Structured information sources allow the data to be queried using a predefined query language. Data- and knowledge bases are typical representatives of structured information sources. Although semi-structured information sources also allow data to be queried, no predefined query language exists. Knoblock et al. (1998) consider an information source to be semi-structured if a formal grammar can be used to retrieve the information, e.g. HTML documents, LaTeX source, etc. In contrast, unstructured sources have neither a form of standardized organization nor it is possible to identify precise relations among the data. In general, natural language text is considered to be unstructured.

One the Web, most data is encoded in HTML where natural language parts are mixed with structured parts that have an inherent relational or semantic structure. Examples are tables that contain a detailed product list and other data that has been propagated from a database as HTML code.

2.1.1 Characteristics of Web Data

```
<html>
  <head>
    <title>Product Web Page</title>
  </head>
  <body>
    <h1>Available Books:</h1>
    <p>Reduced price for ...</p>
    <table>
      <tr>
        <td>A Brief History of Time
           <a href="http://... ">S.
           Hawking Web Page</a>
        </td>
        <td>EUR 9.45</td>
        <td>S. Hawking</td>
      </tr>
    </table>
  </body>
</html>
```

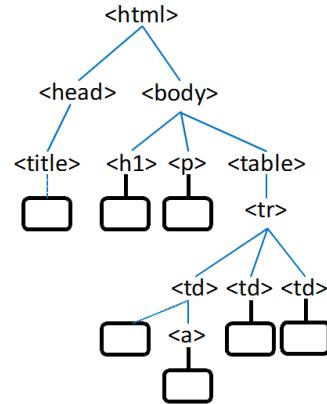


Figure 2.1: HTML document (left) and its tag tree based on DOM (right).

The Web consists of huge collections of interlinked documents encoded in HTML – the *HyperText Markup Language*⁶. In HTML documents (Figure 2.1, left), text and other elements are structured using designated markup elements i.e. HTML tags. These can be classified in *block level* and *inline or text level* elements (Figure 2.2).

Focusing on different levels of data in HTML can have an influence on retrieval, hence, the accuracy of the information obtained.

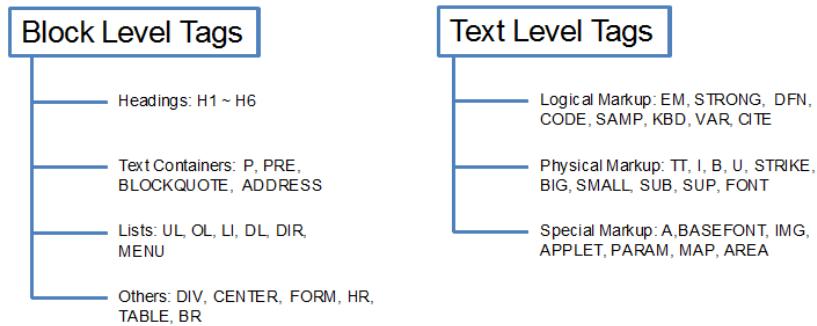


Figure 2.2: Classification of HTML elements.

⁶<http://www.w3.org/MarkUp/>

Each document on the Web has a unique identifier, the URL (*Uniform Resource Locator*⁷), that is used by web browser applications to identify and access the respective documents.

A *web site* consists of one or several interlinked HTML documents usually of a single domain e.g. a website for movies, medical knowledge, etc. In practice, however, a website can be more complex and consist of several types of documents and even different forms of output e.g. a table, picture, etc. from scripts or services that is dynamically (e.g. AJAX⁸) embedded within the HTML scaffold.

Depending on the structural elements and type of content used, a web document can be more or less structured. Natural language text passages can be mixed with structured parts such as tabular content. As a result, HTML is a *semi-structured* document type where the schema (column attribute information) of the structured parts is lost and a separation of content and structure is no longer easily possible.

Hyperlinks, i.e. URL's are the means to weave the Web and build the Web graph where each node represents a document. A hyperlink in HTML represented through the tag `<a>` utilizes URLs for referring to other documents, for example:

```
<a href="http://www.yahoo.de/index.html">Yahoo</a>.
```

Although generally speaking, we talk of documents, hyperlinks can also link to other resources such as images, audio/video, web executable scripts or services, etc. Hyperlinks that connect web pages carry important information that can be exploited to improve retrieval as well as learn about the actors that produce the interlinked web documents and their behavior. Essentially, the web can be regarded as a virtual social network where each page represents an actor and each link a relationship. Hence, methods from social networking theory can be applied in order to analyze its structural properties as well as the role, position, and prestige of each social actor. Groups of actors can build communities i.e. sub-graphs that can also be analyzed applying social network analysis.

Hyperlinks can be used either to organize information on the same site or point to pages on other sites (Henzinger, 2005). Outgoing links often indicate an implicit conveyance of authority to the web pages being pointed to, i.e. those pages which are likely to contain authoritative information.

⁷<http://www.w3.org/Addressing/>

⁸<http://www.w3.org/standards/webdesign/script.html>

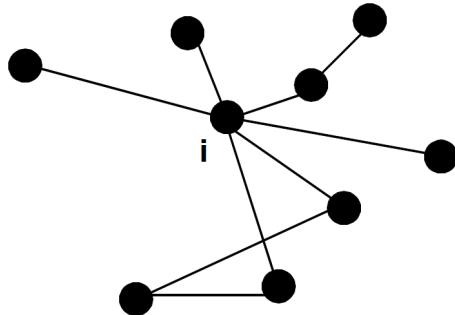


Figure 2.3: A social network. Actor **i** is the most central one.

In this sense, the Web can be regarded as a directed graph based on two assumptions:

- Assumption 1: hyperlink between pages denotes author perceived relevance (quality signal), and
- Assumption 2: the anchor of the hyperlink describes the target page (text).

Two types of social network analysis (Liu, 2007b, chap. 7), *centrality* and *prestige*, can be applied for analyzing hyperlink structures on the Web. Subject to analysis are “prominent” actors that are extensively linked with others. The assumption is, that an actor involved in many ties is considered to be more important (Figure 2.3) than actors with fewer contacts.

For analyzing centrality, properties such as *degree*, *closeness* and *betweenness* can be investigated. Prestige or authority is a more refined measure of prominence than centrality. Here, ties sent (out-links) and ties received (in-links) (Figure 2.4) are distinguished. The main difference is that prestige focuses more on in-links while centrality focuses on out-links. The properties of prestige are degree, proximity and rank.

Algorithms in search-engines such as Google exploit the web hyperlink structure, e.g. PageRank (Brin and Page, 1998) by applying social network analysis to rank documents according to their importance. A web page about “Obama” that is linked very often by other web pages, is hence regarded as more important than pages that are less referenced.

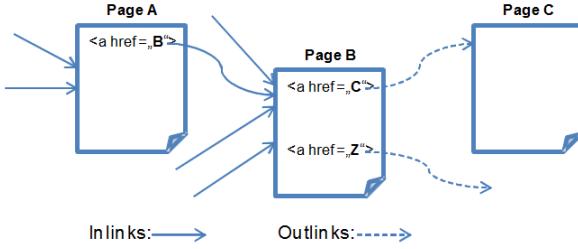


Figure 2.4: Web page hyperlink structure (in/out-links).

2.1.2 Data Representation Models

Depending on the purpose of processing, such as retrieval, pattern mining or information extraction (IE), various models to encode a web document exist. As a result, different granularities for encoding a document can lead to different results and allow for the discovery of different information patterns. In general, it is possible to build abstractions for the following levels; page, site, domain or web scale.

In the following sections, an overview of existing encoding schemes to translate a web page into an abstract representation is provided.

Linear Representation

HTML documents can be processed in many ways. The simplest way is to read and process the sequence of tag tokens and content elements of the entire document in linear form as defined below.

For a sequence T of tag tokens t_0, t_1, \dots, t_{N-1} the following functions can be defined:

- $index : t_i \rightarrow i$, where $i \in \{0, 1, \dots, N - 1\} = I$ and $N = \text{length of the HTML document}$, $t_i \neq t_j$ for $i \neq j$.

The $\text{index}()$ function assigns an index i to a token t_i in the linear token string. Consider, that tokens at different positions are regarded as distinct, despite using equivalent HTML tags as content, format and layout may differ for each piece of data. The following two functions $\text{htmlTag}()$ and $\text{data}()$ assign subsets of I (the set of token indices) to either HTML marked up positions or data fields.

- $htmlTag : t_j \rightarrow h$, where $h \in H = \{\text{HTML}, \text{A}, \text{I}, \text{DIV}, \text{P}, \text{TD}, \dots\}$ is the set of HTML tags of the form $\langle h \rangle$ or $\langle /h \rangle$ and $j \in J \subset I$.

This function assigns an HTML tag from the available vocabulary (e.g. HTML standard 1.x) to each token t_i in the linear token string. Hence, the value of t_i can be an HTML tag or data (content information) introduced below, while J represents the linear index positions that are filled with HTML tags.

- $data : t_k \rightarrow d$, where $d \in D = \{PCDATA \cup l \mid l \in L \subset H \text{ and } k \in K \subset I\}$ as the set of data items such as text (PCDATA) and other leaf elements 1 (*bachelor tags*) such as IMG, BR, HR, etc. that have no subsequent elements to be enclosed by opening and closing tags, $k \in K \subset I$.

The last function assigns a linear index token position k to leaf elements, which consist either of textual data or bachelor tags, such as images, etc.

In Figure 2.5 an example of the HTML from the previous example is shown as a sequence of tags.

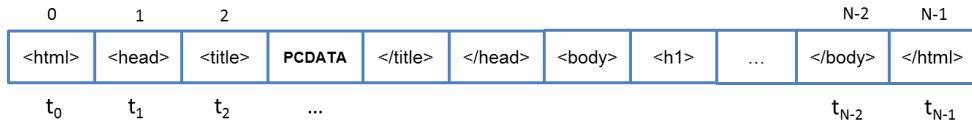


Figure 2.5: HTML tag sequence. Data elements are denoted by PCDATA.

Annotations in HTML

User-provided annotations could be introduced by either inserting pairs of opening-closing tags (as semantic labels) surrounding the data elements to be annotated and creating an index list utilizing the described functions or using attributes in the leaf elements that contain the data.

For the HTML document in Figure 2.1, a semantic indexing could look like:

- 11 → title, 16 → website, 18 → price, 16 → author

Besides such atomic entities, an entire semantic structure could be marked, e.g. *Book* by adding a semantic label “Book” to the containing `<tr>` tag, by simply using an additional attribute such as “class”: `<tr class="Book">`.

In a similar way, the atomic labels title, website, etc. listed before could be added to the respective containing HTML tag using a dedicated “property” attribute, e.g. at index 16: `<td property="title">`. In RDFa, which will be introduced later, the same principle based on properties is used. In general, annotations can be integrated with the related data records directly (inline annotation), or stored and referenced separately (offline annotation), e.g as an XML file.

Tree Representation

Abstracting from the content embedded in HTML and regarding each content element simply as a text or a bachelor tag, a tree abstraction for an HTML document can be obtained. The HTML tree consists of a root element, internal nodes (having tag names and tag attributes) that are HTML tags and leaf elements that can be text strings (PCDATA) or bachelor tags that have no closing tag in the HTML specification, which is in contrast to the XHTML specification where each opening tag must have a corresponding closing tag. The children of a node can be accessed using their labels or respective indices. The leaf elements can be reached by traversing along the path from the root to a leaf element. Paths that identify leaf elements can be represented as a list with the root as the first element.

XHTML trees⁹ as well as XML trees can be modeled as unranked trees (Neven, 2002, Libkin, 2005) over a finite alphabet¹⁰ Σ following the definition provided below:

Definiton 2.1. (*XML as unranked¹¹ ordered tree = Σ -tree*). *The set of Σ -trees, denoted by τ_Σ , is inductively defined as follows:*

- every $\sigma \in \Sigma$ is a Σ -tree; (every tag/node is a tree)
- if $\sigma \in \Sigma$ and $t_1, \dots, t_n \in \tau_\Sigma$, $n \geq 1$ then $\sigma(t_1, \dots, t_n)$ is a Σ -tree.

For every tree $t \in \tau_\Sigma$, the set of nodes of t , denoted by $\text{Dom}(t)$, is the subset of \mathbb{N}^ (the set of strings over the alphabet consisting of the natural numbers) defined as follows:*

- if $t = \sigma(t_1, \dots, t_n)$ with $\sigma \in \Sigma$, $n \geq 0$, and $t_1, \dots, t_n \in \tau_\Sigma$, then $\text{Dom}(t) = \{\varepsilon\} \cup \{u_i \mid i \in \{1, \dots, n\}, u \in \text{Dom}(t_i)\}$.

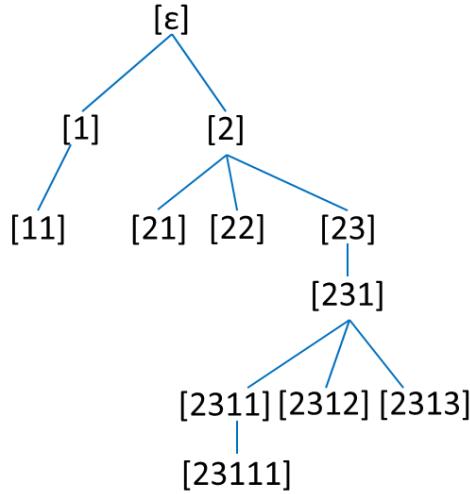
⁹XHTML is based on XML (a subset of SGML), while HTML was originally defined via SGML as its application.

¹⁰Consider, that the alphabet is only known if a schema is available, e.g. as DTD, XSD.

¹¹As there is no a priori bound on the number of children of a node in a Σ -tree; such trees are therefore unranked.

Thus, ε represents the root while u_i represents the i -th child of u . Finally, labels can be assigned via a respective mapping, $\text{lab}^t(u)$ as the label of u in t .

Example 2.1. The domain of the HTML core tree (no leaves) shown graphically in Figure 2.1 (right) can be represented following the definitions above as a Σ -tree, here in list-form: $\{\{\varepsilon\}, \{1\}, \{11\}, \{2\}, \{21\}, \{22\}, \{23\}, \{231\}, \dots\}$.



Although relying on a finite alphabet is a restriction regarding XML representations, for HTML, which is based on a finite set of tag elements, this representation fits well.

Definiton 2.2. (Binary relations in unranked trees.) Nodes in unranked trees are elements of \mathbb{N}^* , i.e. finite strings whose letters are natural numbers. A string $s = n_0n_1\dots$ defines a path from the root to a given node. The string s is built by applying string concatenation (dot operator) to the natural numbers $n_i \in \mathbb{N}^*$.

Then, basic binary relations on \mathbb{N}^* can be defined as:

- *child*: $s <_{\text{child}} s' \Leftrightarrow s' = s \cdot i$ for some $i \in \mathbb{N}$
- *next sibling*: $s <_{\text{next sibling}} s' \Leftrightarrow s' = s_0 \cdot i$ and $s' = s_0 \cdot (i+1)$ for some $s_0 \in \mathbb{N}^*$ and $i \in \mathbb{N}$
- *first child*: $s <_{\text{first child}} s \cdot 0$

From the algorithmic perspective such a tree can be created from the linear-scanned HTML source code by applying *depth-first search* and the

left-to-right principle. For subsequent processing each node can be marked by a unique ID during the traversal.

The Document Object Model (DOM) specification is based on such an abstraction for representing HTML documents in form of tag trees (Figure 2.1, right). DOM is an interface specification for accessing HTML or XML-documents in a well-defined form using HTML or XML parsers that process a document as a tree with nodes having attributes and child nodes. The tree has one dedicated root element as the initial point for reading the document.

In web page analysis, HTML code is pre-processed e.g. cleaning, filtering, etc. before the DOM tree – that is subject to further structural analysis – can be created. The flexibility of the HTML syntax leads to a lot of web pages that do not obey the W3C specification resulting in ill-formatted tags that can't be corrected even by using tools such as Tidy¹² for creating well-formatted and valid HTML. This is the reason why the DOM tree representation does not provide correct trees for all documents to be used for further processing.

From a tree representation of an HTML document an element, i.e. node or leaf can be uniquely identified and its content retrieved by defining an XPATH¹³ expression. For the DOM tree example shown in Figure 2.1 the link node (`<a>`) in the HTML table could be identified via the following XPATH expression:

```
\HTML\BODY\TABLE[0]\TR[0]\TD[0]\A[0]
```

(Semi-)structured Data Representation

Data embedded in HTML often has a (semi)-structured form that can be modeled as *nested relations*. Focusing on the structured data elements in HTML shows that two types of widespread data rich pages (Figure 2.6) exist, which could be annotated using fixed templates from a structured Web data model and its appropriate HTML mark-up encoding scheme:

- *List pages*, that contain a list of data objects. Here, visually, horizontal and vertical data regions can be identified. It is assumed, that within each region every data object is formatted using the same template.
- *Detail pages*, that focus on a simple object and have a designated HTML structure.

¹²<http://tidy.sourceforge.net/>

¹³<http://www.w3.org/TR/xpath/>

Besides list and detail pages, more complex variations may exist that comprise horizontally or vertically nested list structures. Examples are the nested HTML tables or various hierarchies that use `<div>` elements in combination with tables and other structural HTML elements, e.g. paragraphs `<p>`, lists ``, etc.

The screenshot shows a composite of two web pages. On the left is a photograph of Cristiano Ronaldo in a white Real Madrid jersey, looking towards the right. On the right is a detailed game report from the 2006 World Cup between Germany and Portugal.

FIFA Fussball-Weltmeisterschaft Deutschland 2006™

MATCH Report

Deutschland - Portugal

Spieldatum: 05.07.2006 | **Spielort / Stadion:** Stuttgart / Gottlieb-Daimler-Stadion | **Uhrzeit:** 21:00 | **Zuschauer:** 52000

Dritter Platz | **3:1 (0:0)**

Erzielte Tore: Bastian SCHWEINSTEIGER (GER) 56', PETIT (POR) 60' Eigentor, Bastian SCHWEINSTEIGER (GER) 78', NUNO GOMES (POR) 88'

| Deutschland | Portugal |
|-------------------------------------|--------------------------------|
| [1] Oliver KAHN (GK)(C) | [1] RICARDO (GK) |
| [2] Marcel JANSEN | [2] PAULO FERREIRA |
| [3] Michael BALLACK | [3] RICARDO COSTA |
| [4] Jens NOWOTNY | [4] FERNANDO MEIRA |
| [5] Bastian SCHWEINSTEIGER (-79') | [6] COSTINHA (-46') |
| [6] Philipp LAMMANN | [7] SIMAO (77') |
| [7] Miroslav KLOSE (-65') | [8] JOSÉ VALENTE (45') |
| [8] Bernd SCHNEIDER | [9] CRISTIANO RONALDO |
| [9] Lukas PODOLSKI (-71') | [10] HANICHE |
| [10] Michael BALLACK (C) | [11] SIMAO |
| [11] Jens LEHMANN | [12] QUIM (GK) |
| [12] Arne FRIEDRICH | [13] RICARDO SANTOS (GK) |
| [13] Miroslav KLOSE | [14] CANEIRA |
| [14] Gerald ASAMAH | [15] BOA MORTE |
| [15] Thorsten HITZINGER (+79') | [16] RICARDO CARVALHO |
| [16] Tim MEISTER | [17] TIAGO |
| [17] Tim BOROWSKI | [18] NUNO GOMES (+49') |
| [18] David COONKOR | [19] HELDER POSTIGA |
| [19] Jürgen KLINSMANN (GER) | [20] Luiz Felipe SCOLARI (BRA) |
| [20] Trainer | |
| [21] Celestin KTAGUNGOSIRA (RWANDA) | |

Verwarnungen: Torsten FRINGS (GER) 7', RICARDO COSTA (POR) 24', COSTINHA (POR) 33', PAULO FERREIRA (POR) 60', Bastian SCHWEINSTEIGER (GER) 78'

Platzverweise:

Schiedsrichter:
Toru KAMIKAWA (JPN)
Coffi CODIA (BEN)
Funfur Offizielle
Celestin KTAGUNGOSIRA (RWANDA)

Schiedsrichterassistent 1: Yoshikazu HIROSHIMA (JPN)
Schiedsrichterassistent 2: Kim Dae Young (KOR)

Torschütze gegen Almeria: Real-Superstar Ronaldo. (Foto: Reuters)

SPANIEN - PRIMERA DIVISION

- > [Ergebnisse und Tabelle](#)
- > [Die Top-Torjäger](#)
- > [Aktuelle Karten-Statistik](#)
- > [Aktuelle Elfmeter-Statistik](#)
- > [Aktuelle Torhüter-Statistik](#)
- > [Die ewige Tabelle](#)
- > [Alle spanischen Meister](#)
- > [Aktuelle News zum FC Barca](#)
- > [Aktuelle News zu Real Madrid](#)

Vor 20.900 Zuschauern im Stadion Juegos del Mediterraneo erzielten Cristiano Ronaldo (27.) und der ehemalige Hamburger Rafael van der Vaart (69.) die Tore für Real. Die Gastgeber waren durch Albert Crusat (14.) in Führung gegangen.

Figure 2.6: Typical (semi-)structured web pages containing mainly unstructured text (left) or detailed structured or tabular information (right).

Considering that web pages contain not only data records, but also other information such as spam, menu structures, etc., such parts can be seen as noise and make annotation and further processing, e.g. extraction, difficult, as they also can have a similar HTML structure as the relevant data records.

Definiton 2.3. (*Data model based on nested relations according to Liu (2007a, chap. 9.).*)

1) *Atomic types:*

The given is a set of basic types $B = \{b_1, \dots, b_k\}$. Each b_i is an atomic type¹⁴, its domain $D^b = \text{dom}(\{b_i\})$ a set of constants.

2) *Tuple types:*

If $\{B_1, \dots, B_n\}$ are basic or set types $\Rightarrow [B_1, \dots, B_n]$ is a tuple type with the domain $D^{\text{tuple}} = \text{dom}([B_1, \dots, B_n]) = \{[x_1, \dots, x_n] \mid x_i \in \text{dom}(B_i)\}$.

¹⁴Atomic types correspond to attribute types known from relational databases, e.g. integer, string, etc.

3) *Set types:*

If T is a tuple type $\Rightarrow \{T\}$ is a set type with the domain $\text{dom}(\{T\})$ being the power set of $\text{dom}(T)$.

Nested relations can be modeled based on basic or atomic types, tuples and set types. Following this scheme, classic flat relations that correspond to un-nested or flat set types and nested relations that are of arbitrary set types, can be represented.

Based on these definitions, a tree representation for types and their instances can be deduced:

Definiton 2.4. (*Type trees.*)

- a) Basic types can be represented as a leaf or node.
- b) A tuple type T is a tree rooted at tuple node with n sub-trees, one for each T_i .
- c) A set type is a tree rooted at a set node with one sub-tree.

Following these definitions, instances of basic types, tuples, sets etc. can be represented using trees as well. Tuple instances are called “data records”, instances of set types “lists” of ordered data records according to a particular web page structure. An example of a tree structure is shown in Figure 2.7. A detailed description for defining various typed data structures is provided by Liu (2007a, chap. 9).

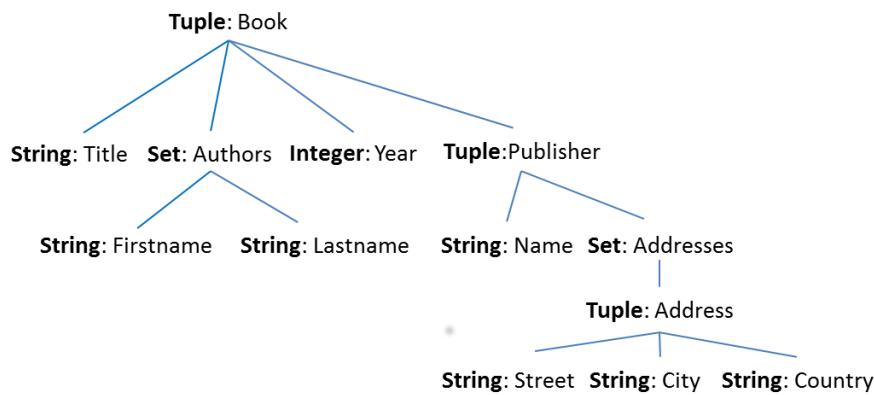


Figure 2.7: A sample type tree representing a book.

Visual Web Page Representation

As an alternative to a tree or token representation of the HTML structure visual characteristics of a web page – such as the location at which tags or

sub-trees of HTML are rendered – can be exploited for inferring structural relationships. They could also help for detecting semantic relationships.

In a web browser each HTML element that corresponds to a node in the DOM parse tree is rendered as a rectangle, i.e. every tag element is augmented with a bounding box by the upper-left corner's x,y screen coordinates along with width and height. The visual information can be obtained after the HTML code is rendered. On this basis a DOM tree can be constructed based on the nested rectangles.

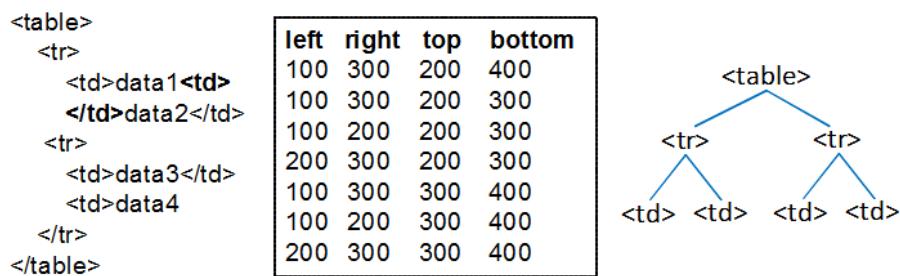


Figure 2.8: HTML fragment, boundary coordinates and its DOM tree.

While processing HTML as a tree or as tokens can lead to a more complicated processing caused by incorrect or invalid HTML code, analyzing visual information can help to re-build the DOM tag tree based on the nested rectangles. In general, the rendering engines in web browsers are highly error-tolerant and as long as the browser is able to display a page correctly the DOM tree can be built. The example in Figure 2.8 shows incorrect HTML code and its re-constructed DOM tree by making use of visual information (Liu, 2007d, pg. 356).

2.2 Knowledge Representation

The previously introduced URLs are a special form of a Unified Resource Identifier (URI), which is a more general concept for identifying any kind of entity in the real world (or its representation) using a particular access protocol on the Internet.

In the **Semantic Web** URIs are used to specify and look up entities and their relations using the `http://` scheme, e.g. to formulate assertions in the Resource Description Framework (RDF) – a meta data model specified

by the W3C. Looking up those specified resources can be done simply by dereferencing the URI over the HTTP protocol.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"
  "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  version="XHTML+RDFa 1.0" xml:lang="en">
<head>
  <title>Product Web Page</title>
  <base href="http://www.amazon.com/ont/" />
</head>
<body>
<h1>Available Books:</h1>
<table>
  ...
  <tr>
    <td><span property="aont:title">A Brief History of Time</span>
      <a href="http://... ">S. Hawking's Web Page</a>
    </td>
    <td><span property="aont:price">EUR 9.45</span></td>
    <td><span property="aont:author">S. Hawking</span></td>
  </tr>
  ...
</table>
</body>
</html>
```

Figure 2.9: Semantic Layering using RDFa.

Staying in the HTML world, semantic markup can be added to existing web pages by utilizing RDFa¹⁵ – a thin semantic layer upon HTML. Here, dedicated attributes, e.g. properties are utilized for adding semantic predicates to the HTML markup that contains information of interest. In the example shown in Figure 2.9 the predicates named *title*, *price* and *author* are taken from a hypothetical controlled vocabulary modeling books on Amazon.

2.2.1 RDF

Compared to HTML that is used to structure and link Web documents, RDF is a universal graph-based data model for structuring and linking data that describes real world entities.

The graph model comprises nodes that represent arbitrary resources and uni-directional edges that help to compose assertions about decentrally organized information entities i.e. nodes or resources. For uniquely identifying

¹⁵<http://www.w3.org/TR/xhtml-rdfa-primer/>

nodes and edges, the URI (Section 2.1.1) concept is adopted and applied to abstract resources that refer to real world entities, such as, a person and its attributes such as name, affiliation, address, etc.

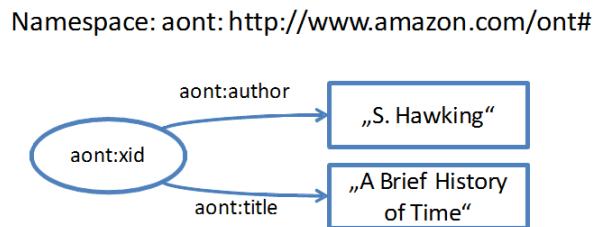


Figure 2.10: A simple RDF graph - nodes and directed edges.

RDF follows a triple structure [subject, predicate, object] for describing resources and making assertions about them, i.e. formulating statements about resources that describe relations between a subject and an object. The subject represents a resource for which a statement is formulated, predicates describe binary (uni-directional) relations between subject and object. An object can be a resource or a literal value, while subjects and predicates are always resources. Consequently, an RDF graph is formed by a set of statements that consist of named resources and relations, and their literal values. In the example in Figure 2.10, the following two statements are formulated using the N-triple notation¹⁶:

```

<aont:xid> <aont:author> "S. Hawking"
<aont:xid> <aont:title> "A Brief History of Time"
  
```

The name `xid` that follows the namespace abbreviation represents a unique id for this resource i.e. the subject for which the assertions were made. In this example, properties (`title`, `author`) from a virtual amazon controlled vocabulary modeling books defined under the namespace <http://www.amazon.com/ont#> are used to describe books that may be offered on the amazon shopping website.

Furthermore, RDF allows the description of other statements by providing a concept called *reification*, which is relevant for creating more complex meta-data describing the context of the formulated statements. As statements (triples) in RDF are binary relations, reification serves to make

¹⁶<http://www.w3.org/2001/sw/RDFCore/ntriples/>

additional statements about existing ones equivalent to many-to-many relations. Important additional information may comprise trust, provenance, uncertainty and other meta knowledge. The following additions serve to reify the aforementioned statements of the book example by using the RDF reification vocabulary comprising the type `rdf:Statement` and the properties `rdf:subject`, `rdf:predicate` and `rdf:object`.

```

<aont:xid> <rdf:type>      <rdf:Statement>
<aont:xid> <rdf:subject>    <aont:Writer>
<aont:xid> <aont:name>     "S. Hawking"
<aont:xid> <rdf:predicate> <aont:wrote>
<aont:xid> <rdf:object>     <aont:Book>
<aont:xid> <aont:title>    "A Brief History of Time"
<aont:xid> <aont:said>     <aont:Wikipedia>

```

It is important to note that the reified statement does not serve to imply¹⁷ the original statement, i.e. from “Wikipedia said that Hawking wrote ABHOT” it can’t be concluded that “Hawking wrote ABHOT”.

Thinking of a network of ontologies (Figure 2.11) that add machine-processable semantics to any kind of web content is exactly what is envisioned by the initiators of the Semantic Web.

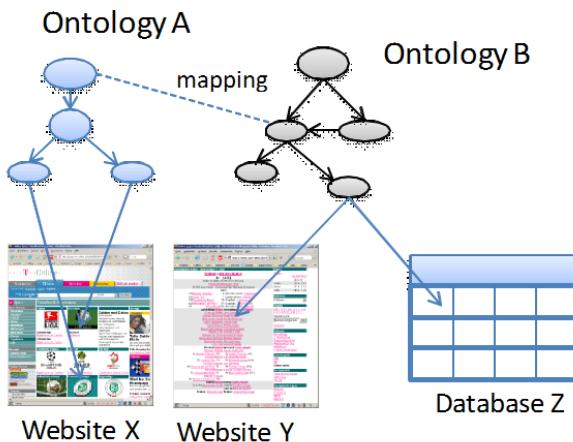


Figure 2.11: Adding machine-processable semantics to web data (Mika, 2007).

In general, ontologies can be organized according to their level of semantics, i.e. their complexity. As ontologies specified using RDF-S or OWL are

¹⁷<http://www.w3.org/TR/rdf-mt/#Reif>

the corner stones of the Semantic Web stack of the W3C (Figure 2.12), their basic concepts will be described next.

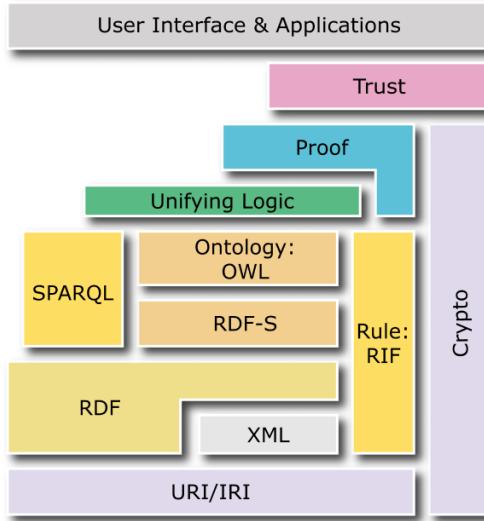


Figure 2.12: The Semantic Web Stack of the W3C (Dengel, 2011).

2.2.2 Ontologies

Philosophers refer to ontology as the theory of “the nature of being and existence” as well as “basic categories of being and their relations”. The Greek philosophers Socrates and Aristotle were the first to deal with the notion of abstract ideas, a hierarchy among them, and class-instance relations, resulting in a well-defined model capable of describing the real world. Currently, different definitions have been adopted by artificial intelligence (AI) researchers and knowledge engineers.

The ontology definition used here goes back to a definition by Gruber in 1993 (“An ontology is a specification of a conceptualization”, (Gruber, 1993)) and further extended and adapted by other researchers (“ontology is a formal and explicit specification of a conceptualization of a domain of discourse” (R. Studer, 1998)). This definition is widely used in the Semantic Web. Following these definitions, an ontology can be regarded as:

- a shared, formal conceptualization of a domain i.e. description of concepts and their relations within that domain,
- expressed in formal languages following well-defined semantics and

- built upon a shared understanding within a community, i.e. to agree on the concepts and their relations for a domain and how they are used.

RDF-S is a widely used extension of RDF allowing for the creation of shared vocabularies i.e. ontologies. While a shared vocabulary could be seen as a form of a “schema” of a domain – if taking analogies from the database community – instances of classes of an ontology would conform to data records or rows of a database table.

In practice, creating an ontology would entail the following steps: defining the classes, arranging the classes in a taxonomic hierarchy, defining properties (slots) that correspond to relations between the classes and describing allowed values for these and creating instances of an ontology by filling in the values for the slots.

tbox and abox

Core ontologies generally formalize intentional aspects of a domain, while extensional aspects are represented by class instances (Ehrig, 2006). In this sense, ontologies comprise the *tbox*, which contains general assertions and axioms over classes and relations, and the *abox*, which solely contains assertions over class instances. Both, tbox and a box form a *knowledge base*. In the example in Figure 2.13, the extensional aspects in form of instances of the ontology, are linked to concepts of the ontology using instance-of relations and other semantic relations. Basically, all lower levels inherit relations from upper levels. As ontologies represent a certain view or perspective of the reality (or a part of it) it may differ from real world understanding of entities and their relations depending on its purpose, e.g. represent and process web data of a particular domain such as football (soccer). Furthermore, ontologies can be described using different levels or layers of specificity or generality:

- a generic (top level)ontology that contains the ontological theory for the concepts used
- a domain-independent (core) ontology that contains general concepts and relations of the top-level ontology and
- domain ontology that hold concepts and relations that are relevant/for describing a certain domain

- a last layer can be attached to the domain level by linking its instances to domain concepts e.g. “Torsten Frings” instance-of FootballPlayer

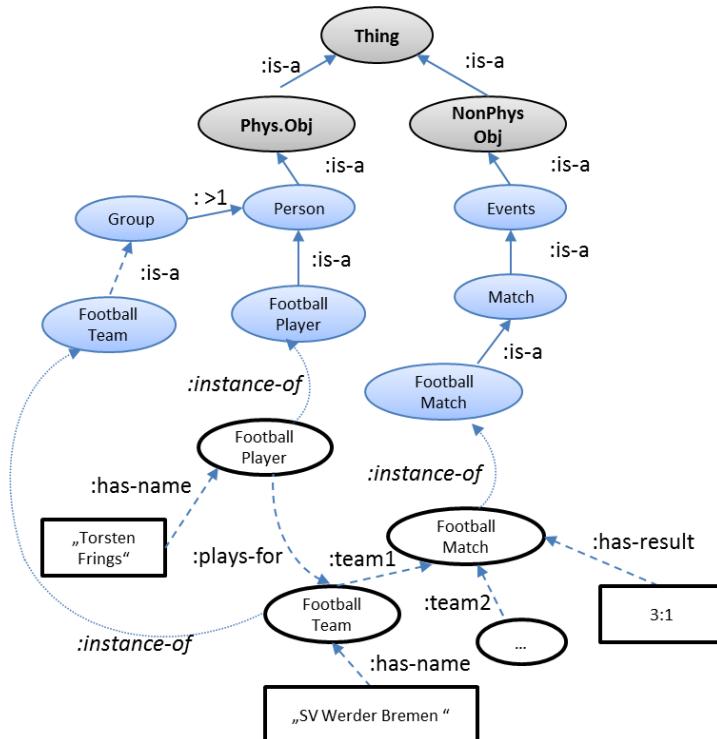


Figure 2.13: Example ontology describing (part of) a soccer domain.

RDF-S allows a definition of simple ontologies and infer implicit knowledge, but does not allow an expression of complex relations that are found in various forms in natural language.

Linguistic Grounding of Ontologies

Buitelaar et al. (2009) argue, that there is a need to separate linguistic and ontological levels. They present an expressive model beyond RDF-S, OWL etc. for associating linguistic information to ontology elements. The approach is based on previous models, such as LingInfo (Buitelaar et al., 2006) – an RDF-S based lexicon model, and LexOnto – a model for specifying the meaning of complex linguistic structures with respect to ontology elements. Linguistic information is associated with ontology elements by introducing so-called LingInfo objects that are attached to classes and properties with a property “linginfo”.

Linked Open Data (LOD)

The “Web of Data” or “Linked Data Web”, based on the linked data principles¹⁸ – a set of rules for linking data entities on the web based on RDF, which Tim Berners-Lee formulated in 2006 – gains growing impact, while its applications, e.g. sig.ma¹⁹ give a sneak preview of what a “semantic” Web could look like. The related developments are driven by the *Linked Open Data (LOD)*²⁰ movement and gain more and more attention from industrial and governmental sectors, which can be well observed by looking at the growth of the Linked Open Data cloud since 2007²¹ and the contributing and participating organizations, such as the governmental institutions from the UK, libraries, large medical knowledge centers, the New York Times, ACM, etc.

Projects such as DBPedia or the broader Linked Open Data (Bizer et al., 2009) initiative help to transform and interlink large existing data sets to RDF. As few mechanisms currently exist that enable non-expert users to semantically annotate web data and contribute to the Semantic Web, methods and practices that have been applied in Web 2.0 can also be adopted for the Semantic Web. A grassroots example how this can be done successfully is Revyu²² – a reviewing website that generates linked data as a “by-product” while allowing users to review anything in the real world.

The advantages of linked data relate to information richness and precision compared to methods that are only based on text processing and analysis. Furthermore, the web of data can be used by software agents as well as by humans affording appropriate visual interfaces or semantic data browsers. While semantic metadata enables semantic search and reasoning on a far higher level, the linked web of data has to deal with noise as well. Existing semantic data browsers or aggregators use many heuristics to deal with heterogeneous data. As it is still regarded as being at its early stage, pragmatic strategies and machine learning techniques may help to exploit the potential of the semantic web. Yet, it is encouraging to see big search engine companies such as Google and Yahoo realizing the value of semantic markup. Hence, it can be expected that the existing form of the Semantic Web (as linked data) will win more quality and data coverage in the near future.

¹⁸<http://www.w3.org/DesignIssues/LinkedData.html>

¹⁹<http://sig.ma/>

²⁰<http://linkeddata.org/>

²¹<http://richard.cyganiak.de/2007/10/lod/>

²²<http://www.revyu.com>

2.3 Emergent Semantics

In the following sections, a basic understanding of social networks, collaborative tagging and ontology emergence from folksonomies is given.

2.3.1 Social Networks and Folksonomies

According to Mitchell (1969) social networks are “social interactions of any kind between a larger group of people”. In the Internet and Web era social interactions of any kind take place through social software, which integrates the user in the process of content creation as well as sharing or organizing information. Popular applications are uploading and tagging photos on Flickr, videos in YouTube or indexing of bookmarks in del.icio.us. The most critical aspect concerning the impact and usefulness of such applications are their user acceptance in the respective target community, i.e. a subset of web users which manifest in network size and the critical mass of users that use the system. Besides application-dependent characteristics, size and structure of the network define the complexity of social software.

Collaborative Tagging

In social or collaborative tagging, users are manually indexing resources by using free keywords, i.e. by assigning one or more tags to existing or new resources which they upload. Because a user shared “controlled vocabulary” is missing, vocabularies that are created this way are referred to as “uncontrolled user vocabulary” - called a folksonomy, a term derived from “folk” and “taxonomy” by Thomas Vander Wal.

The contributing users form a community, i.e. a social network of users that use the system for sharing web resources such as images, web pages or any other data. The tags in such folksonomies can be used to share, look up and organize the shared resources.

Compared to the previously described formal ontologies, folksonomies are flat and have neither hierarchies in the sense of taxonomic *is-a* or *part-of* relation, nor any other kind of explicit semantics. The most common notion of social tagging is “*free tagging*” as a form of mapping of cognitive concepts or terms from users that are associated with resources, which is in contrast to classification of well-defined and formalized concepts. Furthermore, folksonomies can be distinguished as *broad* or *narrow* with respect to the underlying model. Broad folksonomies are based on a bag model (Marlow et al., 2006), where each user holds individual tag sets besides a

shared pool of tags between the users. Narrow folksonomies²³ are based on a set model, where tags are assigned by the users only once during the creation of user-generated content.

Furnas et al. (2006) showed that broad folksonomy systems approximate a power law distribution that can be seen as a specific property of complex systems.

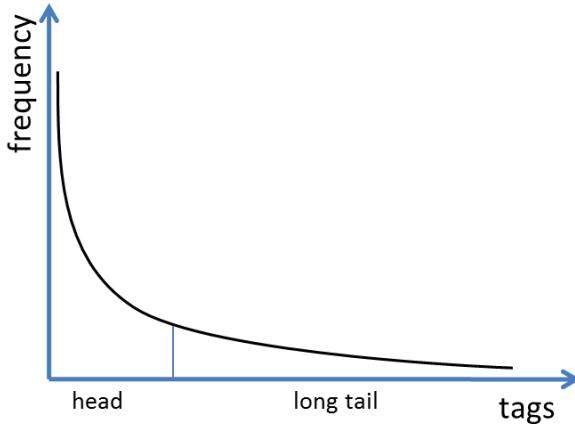


Figure 2.14: Power law distribution in folksonomy systems.

In Figure 2.14, the most frequent tags are represented in the head, while the long tail contains rare tags. Golder and Huberman (Golder and Huberman, 2005) have confirmed in a study based on delicious data, that, indeed, after passing the critical mass of contribution a consensus is formed by the most relevant index terms, i.e. a set of categories that are nearly as stable as those defined by experts, e.g. by catalogers. One important property of the power law distribution is, that it is scale invariant¹, which is regarded as a major property of complex systems.

Though the individual resources are most frequently accessed by using the top most index terms, also a fair percentage of users also use terms from the “long tail” for generally refining queries. Van der Wal²⁴ described the distinct stages (Figure 2.15) of a growing tagging system in 2007.

It is important to reinforce that control in folksonomy systems is held by the users as a result of social interactions and inputs. Moreover, the dynamics of interaction and participation are different in different tagging systems, e.g. free, suggestive, etc. It is clear that different usage models and

²³<http://www.personalinfocloud.com/blog/2005/2/21/explaining-and-showing-broad-and-narrow-folksonomies.html>

¹http://en.wikipedia.org/wiki/Scale_invariance

²⁴Folksonomy (<http://www.vanderwal.net/folksonomy.html>)

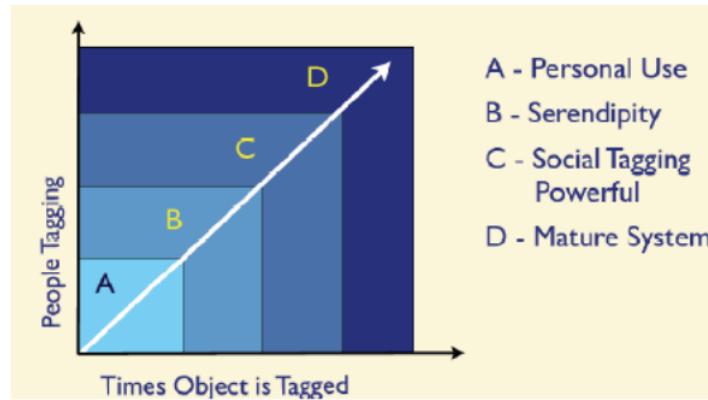


Figure 2.15: Scaling of social tagging systems.

forms of output are needed by different types of folksonomy systems that rely on an appropriate incentive model. For this reasons, studies should consider the motivations and incentives that drive user participation.

Marlow et al. (2006) describe a model and a taxonomy of tagging systems together with corresponding incentives and contribution models. In order to influence the social tagging behavior of users and to weaken user control some forms of tagging support have been proposed some time ago. One of the main advantages of folksonomy systems is the prospect of reasoning about user, tags and resources, which can be based on top of the following tri-partite model for deriving ontological relations from folksonomies.

Tripartite Model of Ontologies

Traditional ontologies follow a model consisting of concepts and their instances which can be represented by a bipartite graph. This model can be extended by introducing actors, hence, the social component forming a tripartite graph model (Figure 2.16) first formalized by Mika (2005).

Definiton 2.5. (*Tripartite model of ontologies*)

A set of concepts (tags/keywords) $C = \{c_1, \dots, c_l\}$, instances (items/resources) $I = \{i_1, \dots, I_m\}$ and actors (users) $A = \{a_1, \dots, a_k\}$ form a tripartite graph.

A folksonomy, can then be defined as a set of annotations: $T \subseteq A \times C \times I$, represented as a hypergraph with ternary edges.

A ternary association is represented by edges that connect a given actor with a certain instance using a certain concept. As a result, the (tripartite)

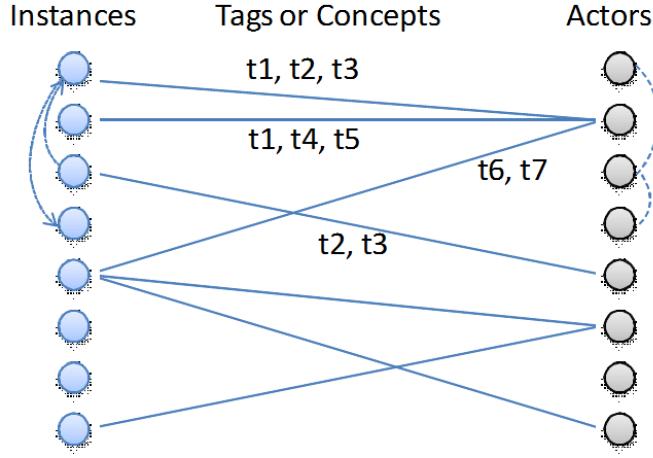


Figure 2.16: Resources, Tags and Users.

hypergraph of a folksonomy T , can be defined as $H(T) = \langle V, E \rangle$, where $V = A \cup C \cup I$, $E = \{\{a,c,i\} \mid (a,c,i) \in T\}$

For simpler understanding of such a model, the defined hypergraph can be reduced into 3 bipartite graphs AC (actors and concepts), CI (concepts and instances) and AI (actors and instances). Folding a bipartite graph into two graphs allows for the uncovering of semantics from the social network formed by the collaborative tagging process. For example, from the AC graph (the affiliation network of people and concepts) two distinct graphs can be extracted; a social network of users based on overlapping sets of objects and a lightweight ontology of concepts based on overlapping communities. Furthermore, lightweight ontologies can also be extracted from overlapping sets of instances.

Problems of Folksonomy Systems

Like all other existing text processing techniques, folksonomy systems also have to deal with the complexity and ambiguity of natural language stemming from different word forms, synonyms, polysemy, etc. Hence, pre-processing steps known from natural language processing such as stopword removal, stemming, etc. must first be applied.

Besides this, free tagging can lead to a large diversification of index terms. Although, on global level users might benefit from the “long tail”

of rare used tags for refining their search query, on personal level the recall can be lowered, if index terms are too diverse. A problem described as “future retrieval” (Marlow et al., 2006) can arise if users change their tagging behavior by e.g. introducing a new category. Earlier tagged resources fitting into the newly introduced category cannot be found. Furthermore, newly introduced tags are not attached to older resources (Golder and Huberman, 2006), i.e. how to react to changed mental model of users, re-integration of old resources, etc. poses a significant challenge. In order to combat the existing problems, several techniques have been developed. *Batch editing* allows the manual change of old tags according to a new tagging strategy, while *tagging support* allows to deal with misspelling, etc. Marlow et al. (2006) describe different strategies; blind, viable or suggestive tagging.

In a follow-up process, appropriate methods of ontology emergence (described in Section 2.3.1) have to be used to unveil associated implicit ontological or semantic relations for the tag sets at hand.

2.4 Metrics for Semantic Analysis

Similarity metrics serve as the basis for various forms of semantic analysis for comparing two or more entities, e.g. words, concepts, sentences, etc., and other structures in information processing systems, such as information retrieval or search systems. In the following an overview of the most important types of measures is given.

2.4.1 Basic Metrics

The cosine similarity measure (Grossman and Frieder, 2004, pg.2-18) is used to compare two vectors in a multi-dimensional vector space. Depending on different retrieval models, i.e. Boolean (Baeza Yates and Neto, 1999, pg.2-25), vector-space (Baeza Yates and Neto, 1999, pg. 27), etc., documents as well as queries can be represented as vectors in a “bag-of-words” model. The cosine similarity measure calculates the cosine of the angle between the query and document vector. As the vectors can be of different length, the cosine similarity measure “normalizes” the results, i.e. the assumption is, that document length has no impact on relevance.

Let D be a document vector (d_{i1}, \dots, d_{it}) of size t and Q a query vector $(w_{q1}, w_{q2}, \dots, w_{qt})$ filled with their corresponding term weights. The cosine similarity coefficient is defined as:

$$S_C(Q, D_i) = \frac{\sum_{j=1}^t w_{qj} \cdot d_{ij}}{\sqrt{\sum_{j=1}^t (d_{ij})^2 \cdot \sum_{j=1}^t (w_{qj})^2}}$$

For (asymmetric) Boolean attributes, similarity metrics such as, the Jaccard (Markov and Larose, 2007, pg. 38) and Dice (Grossman and Frieder, 2004, pg. 19) metric, that are based on set intersection and overlap have been proposed.

2.4.2 Graph-based Metrics

In semantic models such as a thesaurus or word nets, similarity measures are based on path comparison within the used graph model. Wu and Palmer (Wu and Palmer, 1994) propose a metric that works with path lengths and the most specific common subsumer (LCS) (in ontologies: most specific, i.e. lowest upper class that contains both concepts). Let c_1, c_2 denote the entities to be compared, c_s , the most specific upper class node. Furthermore, n_1, n_2 are the number of “is-a” links from c_1 and c_2 to c_s . n_3 is the number of “is-a” links from c_s to the root. The similarity is then defined as:

$$sim_{wup} = \frac{2n_3}{n_1 + n_2 + 2n_3}$$

Lin’s hybrid similarity measure (Lin, 1998) developed from the Resnik metric (Resnik, 1995) takes into account in addition to structural characteristics of the regarded graph, the information that is shared by two concepts as well as the differences between them.

Another similar hybrid approach was described by Jiang and Conrath (Jiang and Conrath, 1997). Evaluations using the wordnet taxonomy²⁵ showed results with a high accuracy. Hammouda and Kamel (Hammouda and Kamel, 2004) describe a model that incrementally creates a directed graph having words as nodes and relations as edges. The similarity of two graphs is computed on the basis of the overlap, the length and frequency of contained paths. In their methods, pre-processing using classic NLP methods such as stopword removal, stemming, etc. for cleaning up the used HTML documents is essential. Pairs of words are compared using the cosine similarity metric on the basis of TF-iDF term weights. Another aspect is the usage of different levels of abstraction for the HTML tags, such as title, meta information, and headers.

²⁵<http://wordnet.princeton.edu/>

2.4.3 Semantic Analysis of Folksonomies

The user-generated tag structures in folksonomies are rather flat and do not contain any explicit semantics. Despite this, semantic relations, e.g. semantic relatedness of tags can be derived by analyzing tags and their usage. According to the definition given by Gurevych and Niederlich (2005) semantic relatedness denotes the “degree of relatedness of a pair of terms”. Semantic similarity can be seen as a special case of a semantic relation in addition to others. Example relations are:

- polysemy (words with several meanings) and homonymy (homonymous word, e.g. kiwi - fruit, bird), which deal with ambiguities.
- synonymous, identity relations - synonyms e.g. student - pupil
- super-class/sub-class or hypernym-hyponym relations, e.g. fruit - apple, banana
- part-of relations (holonym-meronym): hand (holo) - finger (mero)
- contrary relations, antonyms e.g. cold and hot

Methods for analyzing tags based on similarities are either based on statistical analysis or lexico-semantic analysis utilizing external sources such as a thesaurus, WikiPedia or word nets.

In order to remedy some of the previously described problems of folksonomies typical methods for tag normalization, e.g. stemming (Frakes, 1992), the Levenshtein distance (Damerau, 1964) or Google and WikiPedia (Cilibrasi and Vitanyi, 2007, Gabrilovich and Markovitch, 2007) are utilized.

Co-occurrence Analysis

The statistical co-occurrence analysis represents a method that analyzes if and how often tags occur together (co-occur), and in which contexts they co-occur, e.g. in sentences, documents, etc. within a set of tags that have been assigned to a resource by multiple users. The method relies upon the “distributional hypothesis” (Lux et al., 2007).

Furthermore, several similarity metrics exist in order to calculate similarities of tags and resources or between query and resources, e.g. documents. In principle, characteristics (features) of the compared items are converted into a binary or numerical vector.

A simple co-occurrence calculation (matching coefficient) is performed by counting the co-occurrence of pairs of tags in a corpus. Absolute co-occurrence is calculated as the number of resources on which a pair of tags (t_i, t_i) co-occur, i.e. have been assigned to. Table 2.1 shows an example of a 4 x 4 tag co-occurrence matrix.

| | t_1 | t_2 | t_3 | t_4 |
|-------|-------|-------|-------|-------|
| t_1 | 5 | 0 | 2 | 2 |
| t_2 | 0 | 3 | 0 | 1 |
| t_3 | 4 | 1 | 3 | 3 |
| t_4 | 2 | 0 | 4 | 0 |

Table 2.1: Tag co-occurrence matrix with absolute counts.

Other typical used metrics are the Jaccard, dice, overlap or the cosine coefficient (Manning and Schütze, 1999). The latter is used widely in tag analysis and co-occurrence calculation. The Jaccard coefficient for tag analysis is defined as follows:

Definiton 2.6. (*Jaccard similarity metric*)

$$Jaccard(t_i, t_j) = \frac{|R(t_i) \cap R(t_j)|}{|R(t_i) \cup R(t_j)|}$$

, where $R(t_i)$ is the set of resources tagged with t_i and $R(t_j)$ the set tagged with t_j .

The normalized co-occurrence for each pair of tags applying the cosine similarity metric is defined as follows:

Let t_i be an n-dimensional tag vector with n being the number of total distinct tags in the data set and t_{ik} being the absolute co-occurrence count of t_i and t_k , i.e. the number of times the two regarded tags co-occur in the entire data set. In case of $t_i = t_j$, the frequency of a single tag in the whole data set can be determined. The relatedness of two tags can then be computed applying the cosine similarity metric:

Definiton 2.7. (*Cosine Similarity on tag vectors featuring co-occurrence counts*)

$$sim(t_i, t_j) = \frac{t_i \cdot t_j}{|t_i| \cdot |t_j|} = \frac{\sum_{k=1}^m t_{ik} \cdot t_{jk}}{\sqrt{\sum_{k=1}^m (t_{ik})^2 \cdot \sum_{k=1}^m (t_{jk})^2}}$$

In the approach of Specia and Motta (2007), a tag t_i is regarded similar or related to a tag t_j if it not only co-occurs with t_j , but also with all the other tags co-occurring with t_j . Consequently, similar or related tags should also have similar patterns of co-occurrence, i.e. also consider the co-occurrence context of tags.

Tag relatedness should not be influenced by popularity as it was pointed out by (Begelman, 2006).

2.4.4 Other Approaches

Document similarity was also computed applying an approach based on self-organizing maps (SOM) (Honkela et al., 1997), making use of Part-Of-Speech tagging and word sense disambiguation (Sinha and Mihalcea, 2007) in order to resolve problems with synonyms and polysemy. Other methods utilize search engines such as Google for calculating semantic relatedness of terms (Bollegala et al., 2007) or use categories in WikiPedia as it is the case in the WikiRelate approach described by Strube and Ponzetto Strube and Ponzetto (2006).

2.5 Machine Learning

Machine Learning (ML) is referred to as the ability of computing systems to “learn” without being “programmed” explicitly. Algorithms of that kind are able to learn how to solve a particular task from experience, while improving their efficiency with more experience. According to Mitchell (1997), a machine learning algorithm is said to “learn from experience E with respect to some Task T and some performance measure P, if its performance on T, as measured by P, improves with respect to E.”

Generally, ML algorithms are based on the idea of generalization and inference from provided sample data, i.e. its experience that is used to “train” the algorithm. Sample data for training generally originates from some unknown probability distribution. A learner is said to be good, i.e. produces useful results for new cases, if it has learned something more general about that regarded distribution. Hence, additional assumptions about the nature of the prediction or target function are necessary in order to solve a learning task for unseen situations, i.e. beyond the data from the training set. In literature, this is referred to as “inductive bias”. Mitchell (1980) states, that “the inductive bias of a learning algorithm is the set of assumptions that the learner uses to predict outputs given inputs that it has not

encountered”. In the case of “Occam’s Razor”²⁶, it is assumed that the simplest consistent hypothesis about the target function is the best. Here, consistency is related to the ability of the hypothesis to produce correct output for all given samples.

Depending on the kind of the available sample data and the learning task, supervised and unsupervised learning is distinguished.

Supervised machine learning algorithms approximate a function $y = f(x)$ with the help of labeled training data (labeled samples). While x is referred to as the input data, the output or prediction is called y given that training samples are represented as a tuple $\langle x, y \rangle$. In the training phase, the learning algorithm learns a hypothesis f^h (prediction function) from the sample data in order to be able to predict or classify unseen input data of the form $\langle x, y \rangle$, i.e. the output y is predicted by the learning algorithm. Besides the binary classification that just uses two different categories e.g. 0 (male) and 1 (female), in regression continuous values, e.g. from an interval $[0,1]$ are predicted. The most common supervised machine learning methods are concept learning, decision trees, artificial neural networks, naive bayes and support vector machines (see (Mitchell, 1997)).

Unsupervised learning algorithms have no pre-labeled samples for predicting the output of unseen data. The algorithms try to discover inherent common characteristics or frequent patterns for data classification. The input data has the form $\langle x \rangle$, hence, as no manual annotation is needed the cost of the data is low. Prominent unsupervised learning methods are clustering - a method to detect data characteristics to group objects according to some similarity measure and self-organizing maps (SOM), a form of an artificial neural network which uses a map like representation of unlabeled input data.

In the following, some of the most common supervised and unsupervised methods are briefly described.

2.5.1 Supervised Learning

In the simplest form of supervised learning, a Boolean valued function is approximated from a set of examples. Such methods try to find the best-fitting hypothesis in a predefined space of possible hypotheses by applying the *general-to-specific ordering rule* for hypotheses. Mitchell (1997) describes the FIND-S algorithm for initializing the most specific hypothesis and generalizing in the case if a positive example is not covered. In order

²⁶<https://www.britannica.com/topic/Occams-razor>

to find the “best fitting” hypothesis, the methods apply the CANDIDATE-ELIMINATION or the LIST-THEN-ELIMINATE algorithm.

In decision tree learning, the learning process works with a tree structure based on two types of nodes, decision nodes (internal nodes) and leaf nodes. A decision node is used to specify some test on a single attribute or feature, while a leaf node represents a class. During classification, the tree is traversed in a top-down manner by evaluating the given test instance (example) according to its attribute values until a leaf node is reached. Prominent implementations of decision tree algorithms are ID3 (Quinlan, 1986) and C4.5(Quinlan, 1993).

Naive Bayes is a probabilistic classifier (Markov and Larose, 2007) based on the Bayes’ Theorem. It is called “naive” because of the independence assumptions for the used features. For a predefined set of m features f_1, f_2, \dots, f_m and n_i the number of times f_i occurs in a document d , each document d can be represented by the document vector $d = (n_1(d), n_2(d), \dots, n_m(d))$. A bayesian classifier creates a probabilistic model by estimating the probabilities $P(A)$, $P(B)$ and the conditional probability $P(B | A)$ from the training data.

Support Vector Machines (SVM), first described by Cortes and Vapnik (1995), represent another type of supervised machine learning method widely used in the field of image recognition, text classification and bioinformatics. The idea behind SVM is to separate one class label from another by constructing a hyperplane in high dimensional space. The best result is obtained when the margin between the hyperplane to the nearest data points (or support vectors) of any class is maximized.

The main advantage of this method is its high accuracy, while having issues with non-linear sample data. This restriction was overcome by using kernel-functions to use hyperplanes, i.e. linear classifiers in a higher dimensional feature space.

2.5.2 Unsupervised Learning

Unsupervised learning is applied widely to open domain IE tasks where manual labeling of sufficient training documents is not possible or very expensive in order to build an appropriate classifier. On the other hand, collecting a huge set of unlabeled data is cheap.

Clustering methods try to exploit the inherent structure of data in a document or a data set for detecting the organization of similar patterns into sensible groups or clusters, while discovering similarities and difference

among the regarded patterns. Basically, a cluster represents a collection of data instances which are similar to each other and “dissimilar” to instances in other clusters. Clustering algorithms use a similarity or distance function to measure how similar data instances are. Furthermore, two types of clustering, *partitional* and *hierarchical* are distinguished. Partitional clustering methods divide the data sets into a few similar subsets, while in hierarchical clustering the data sets or instances are organized in a hierarchical tree structure. Besides this, clustering can be exclusive (*hard clustering*) where members of the cluster are only allowed to belong to one cluster, in contrast to *soft clustering* methods that allows a member to belong to one or more groups using degrees of membership.

The best known partitional clustering algorithm is k-Means. It iteratively clusters the data into k clusters using a distance function. Each cluster has a cluster center (the centroid) which is used to represent the cluster. Usually, the mean of all data points in the cluster is used. The algorithm starts by selecting k data points randomly as start centroids and calculates the distance between each centroid and every data point. Each data point is assigned to the closest centroid. After that, the centroid for each cluster is re-calculated on the basis of the data points in the current cluster. The process is repeated until a convergence criterion is met. The algorithm is described in more detail in (Liu, 2007d, pg. 120).

In hierarchical clustering methods clusters are generated by producing a nested sequence of clusters in form of a tree. Individual data points are at the leaves of the tree, while all data points are covered by the root. Internal nodes contain child cluster nodes, while sibling clusters partition the data points covered by their common parent. The tree structure can be built bottom up (agglomerative clustering) or top down (divisive clustering). Agglomerative methods build the dendogramm (see Manning et al. (2008, ch. 17)) iteratively from the leaves to the root while merging the most similar or near pair of clusters at each level. The process ends if all clusters are merged this way into a single cluster.

In the “Star” clustering approach each element builds a cluster with the closest (or most similar) elements. Consequently, each element is allowed to belong to multiple clusters. The advantage of this method is that it produces very few clusters that each contain more data instances. Other known clustering algorithms such as Clique, String and Single Link methods have been described by Kowalski (1997).

Another interesting learning technique was described by Teuvo Kohonen. Kohonen maps or self-organizing maps (Ritter and Kohonen, 1989) are

based on artificial neural networks and are used to visualize high dimensional data into a low dimensional (generally two dimensions) map. The semantic relationships in the data are reflected by their relative distances in the map, since similar input vectors will be mapped closely forming a cluster. While some distortion is unavoidable, the mapping preserves the most important neighborhood relationships between the input data. The more frequent input patterns are also mapped to larger domains at the expense of the less frequent ones, making it easier to focus on the significant pattern (Lin and Soergel, 1991).

2.5.3 Instance-based Learning

Instance based learning (IBL), e.g. k-Nearest Neighbor (k-NN) (Liu, 2007d, pg. 112) is a supervised learning and classification technique that can be briefly described in the following way. In the training phase, sample objects or instances e.g. from a person database are stored as data points of an n-dimensional space. The data points have so called “class labels” associated that designate their belonging class. An example of a person database (table) could consist of several instances of persons (rows) that have attributes such as weight, height, age and test score etc. In order to generalize a new unseen (target) object, e.g. a person, the stored samples are consulted for finding the appropriate class of the new instance. Therefore, training samples are processed when a new instance arrives by analyzing its relationship to the previously stored samples. A class is represented by a value of a target function that is assigned to the new instance. IBL methods are sometimes called Lazy Learning because they delay the processing, i.e. classification step until a new instance arrives. In the testing or classification phase, unlabeled new data instances are classified by calculating the k nearest data points, i.e. instances that are already classified. Usually, the class of the majority of the regarded data points is then returned.

2.6 Evaluation Methods for IR

The goal of information retrieval evaluation is to test the performance of IR systems with respect to the user’s information need. Although relevance measurement is still the major and widely established concept, user utility or satisfaction and system aspects gain more and more attention.

In general, testing is based on retrieval tasks as practiced by several Text REtrieval (TREC) or Message Understanding Conferences (MUC), which

aim at facilitating and supporting standardized evaluation in text or information retrieval by sharing evaluation datasets, tasks and data collections in different languages and domains.

Basically, information retrieval or extraction systems are tested with different configurations and users. The basis of each system evaluation is represented by a reference corpus containing human judgments for relevance. The difficulty here is, firstly, that obtaining relevance judgments from users is an expensive and time-consuming process. Secondly, most of the time relevance is an inherently subjective and elastic measure, which might be hard to capture for all cases. As information retrieval systems can be composed of several components and use a variety of techniques and metrics that influence relevance, how to assess each individual contribution can also be tricky.

Depending on the target of the evaluation, it can be reasonable to perform either *intrinsic* or *extrinsic* evaluation (Moens, 2006). In information extraction, for example, intrinsic evaluation can be applied to measure the performance of a certain extraction task, while extrinsic evaluation would imply testing information extraction in a broader context, e.g. testing a search engine where extraction plays a role. Intrinsic evaluation refers primarily to using a gold standard, which is usually created by using one or more human experts. For evaluating a system using a gold standard, a sufficiently high inter-annotator agreement (> 0.8) is required. The difficulty here is to obtain an annotated test set, that is large enough to assess the performance of a retrieval system. The results of the algorithm, i.e. the automatically created results, can then be compared to the gold standard or one or more baselines. In the following, a definition of the most important performance measures for information retrieval or information extraction *Precision* and *Recall* (Rijsbergen, 1979) is given.

| | Positive | Negative |
|----------|----------|----------|
| Positive | TP | FN |
| Negative | FP | TN |

Table 2.2: Classification vs. Observation.

Precision (P) is the proportion of correctly classified positive examples divided by the total number of examples that are classified as positive (*exactness*). Recall (R) is the number of correctly classified positive examples in the test set (*completeness*).

Definiton 2.8. (*Precision and Recall*)

$$P = \frac{TP(\text{relevant_retrieved})}{TP + FP(\text{all_retrieved})} \quad R = \frac{TP(\text{relevant_retrieved})}{TP + FN(\text{all_relevant})}$$

Table 2.2 illustrates the relation between the classification (prediction) vs. an external judgement (observation). In the confusion matrix shown, the columns represent the classification, while the rows represent the observation (or reality). In information retrieval, TP correspond to retrieved documents that are relevant, while FP are those that are retrieved but not relevant, etc.

In the ideal case, recall and precision values are close to 1, but in reality a trade-off between precision and recall was observed in the so-called Cranfield experiments (Spaerck-Jones, 1981).

The F-measure represents a combined measure that is derived from precision and recall, trying to balance this trade-off. If precision and recall are of equal importance, the F_1 measure (harmonic mean) is used:

Definiton 2.9. (*Harmonic Mean*)

$$F_1 = \frac{2 \cdot P \cdot R}{P + R}$$

Further important metrics comprise the error rate (E) and the accuracy (A), which are defined as follows:

Definiton 2.10. (*Error Rate and Accuracy*)

$$E = \frac{FP + FN}{N} \quad A = \frac{TP + TN}{N}$$

The accuracy is calculated as the proportion of correct classifications in all test cases (N), while E takes into account both types of errors; False Positives (FP) and False Negatives (FN). Accuracy is an efficient measure particularly for cases where class assignment is exclusive. When comparing accuracy and precision, accuracy can be described as the ratio between an observed or experimental value and a target value.

2.7 Summary

This chapter has introduced the basic concepts and data models of the Web and models for introducing semantic knowledge to semi-structured data. Data representation models are primarily based on token-based processing as known from text processing or document tree models (DOM) and

graphs, which are used to implement semantic web data representations via the RDF model. For example, the use of ontologies as the target knowledge representation has an important impact on the expressiveness of the regarded information system and its retrieval capabilities, allowing the inference of implicit knowledge from the processed information when applied to several distributed data sources. In addition to this, semantic search and information integration for establishing service and data interoperability are regarded as the major benefits of using ontologies while folksonomies represent a convenient method for interacting with systems that allow for processing and interacting with user-generated contents. Depending on the available semantic knowledge – be it implicit or formal and explicit – different methods for analysis and inference are applicable.

The next chapters will try to shed light on the fundamental paradigm shift from classic information retrieval to semantic retrieval and interaction from an HCI perspective, focusing on introducing semantic information to semi-structured data, the role of ontologies and folksonomies in retrieval and information extraction as well as exploiting semantic information in visual retrieval interfaces for improving user experience and exploration.

Chapter 3

Semantic Retrieval and Visual Exploration

Information sources on the classic semi-structured web bear no semantic information, as they generally serve the purpose of human perception, not automatic or machine-based interpretation and understanding. This changed with semantic web initiatives, such as the Linked Open Data (LOD) Web, where in parallel to human-readable web pages, machine-understandable semantics for describing the data is included.

Furthermore, a bottom-up approach for introducing semantics to web data gained momentum through social tagging, a collaborative or social process for generating meta-data provided by a community of users. The main benefit of this approach is the promise of emergent semantics (see Section 2.3) from user-generated vocabularies (folksonomies), which can be exploited for sharing, organizing, and retrieving web resources efficiently. Analyzing user-provided tags allows the further creation and gathering of different views on the data in focus from numerous perspectives and contexts. Extracting these insights from the respective sources directly with automatic methods is often impossible as their semantics can hardly be inferred from e.g. (partly) incomplete textual contents or structured elements alone.

In the following, concepts and models of information seeking are first introduced and discussed in addition to covering the general information browsing approach. Thereafter, Section 3.2 introduces the basic ideas and methods behind information extraction from semi-structured web pages, followed by a concise overview of supervised and unsupervised methods in

Sections 3.2.1 and 3.2.2. In subsequent sections, approaches for labeling structured web data applying social and semantic tagging (Section 3.3) are described, followed by sections on visual retrieval interfaces, methods of semantic analysis and emergent semantics and their exploitation for interaction and exploration of information in Section 3.4.

3.1 Information Retrieval Models

Search or information retrieval (IR) systems allow a user query to be composed, processed and return a set of results. In general, a user query aims to represent the user's information need, for example as a sequence of entered keywords. The query is processed on the basis of an underlying retrieval model such as the widespread Boolean or vector space model. Therefore, the user query as well as the documents to be sought, need to be represented in the respective model as e.g. a bag of words or vectors.

The underlying retrieval model determines the performance, the expressiveness of the query language, and search operations, etc. of an information retrieval system to a great extent. A more detailed introduction of the basic IR models and their enhanced variants is described by Baeza Yates and Neto (1999, ch. 2).

Once a set of results is discovered, a search system allows an interaction with the results, which may be returned as a list of ranked links to the original web documents.

Existing state of the art interfaces of IR systems generally utilize the aforementioned classic IR models and different weighting schemes, such as the term histogram analysis and the TF-iDF scheme (Baeza Yates and Neto, 1999, pg. 29ff), which influence the ranking of the presented results. Web search systems further utilize link analysis (see Section 2.3.1), structural and visual characteristics of web documents to calculate appropriate rank scores. Enhanced approaches allow documents to be queried beyond term matching utilizing, for example, latent semantic analysis (Deerwester et al., 1990) and other techniques that are able to unveil implicit semantics analyzing global statistics of word occurrences in the query contexts. Furthermore, explicit semantics can be introduced utilizing knowledge models, as discussed in Chapter 2.

In addition, different types of arrangements utilizing clustering methods (see Section 2.5.2), grouping of terms, concepts and their relationships via aggregations at different levels of description can be employed. For example, the hyperlink structures and term relations are widely exploited in various

visualizations. The most common metaphor for interacting with a search list is through scrolling and clicking. Folksonomy-based systems have adopted similar interaction structures in their interfaces, while additional features, e.g. tag clouds, related tags, query assistance, etc. have been integrated as well.

3.1.1 Information Seeking

Marchionini (1989) gives the following definition of information seeking, which outlines the basic tasks involved: “information-seeking is a special case of problem solving. It includes recognizing and interpreting the information problem, establishing a plan of search, conducting the search, evaluating the results, and if necessary, iterating through the process again.”

Different theoretical models of how humans tend to seek information have been discussed in the literature, amongst them, the standard (Broder, 2002), the cognitive (Norman, 1988) and the dynamic (Bates, 1989) model. The widely-used standard model consists of a four phased cycle (as described by Salton (1989) and Shneiderman et al. (1998)), which is repeated until a satisfactory result is found:

1. Identifying an information need
2. Specifying a query
3. Examining the results
4. Reformulating the query

Enhanced variants of this basic model have been introduced in order to allow for representing and recognizing the user’s present information need (Marchionini and White, 2007). According to Hearst (2009), today’s web search primarily focuses on query specification and results interaction, while the other tasks are rarely supported. In his influential model of a general task performance Norman (1988) introduces the notion of a *mental model*, stating that “users must first have a basic idea of their goal to be achieved and then use their mental model of the situation to perform some kind of actions in the world”. In contrast to the notion of a “mental model” in the HCI community, where it is described as a mechanism for explaining one’s understanding of a system or interface, here, it is referred to the definition given by Marchionini (1989): “A person’s mental model is a dynamic, internal representation of a problem situation or a system, which can take

inputs from the external world and return predictions of effects for those inputs.”

3.1.2 Browsing vs. Search

A fundamental principle derived from psychological results in cognitive science is called “recognition over recall” - referring to the fact that recognizing things by looking at them is far easier than thinking up something (Lidwell et al., 2003). The analogy in search would be to rather click on a displayed term among a few than thinking about a search term and typing it to the interface. This principle is closely coupled to the theories that compare querying/searching vs. browsing/navigation in data collections or information spaces (Belkin et al., 1993). Aula (2005) corroborate this view in their work by stating that “searching is a more analytical and demanding method ... whereas browsing requires the user to recognize promising looking links.” As a result, it is less mental work to scan a list of items and choose the interesting ones, than thinking of an appropriate search term.

In addition, browsing information structures systematically and following well-determined paths requires a well-suited system to organize categories, classes of information, etc. dedicated to a regarded domain. The main challenge is how to obtain such meta-data which is suited “to describe and organize the information in a way the user is familiar with” Hearst (2009, ch. 3). Such general category systems are described as “a set of meaningful labels organized in such a way as to reflect the concepts relevant to a domain.” Hearst (2009, ch. 8). He further distinguishes three types of such systems: flat, hierarchical and faceted.

Talking of “browsing a collection of items”, an important distinction is made between an *information structure* and a *navigation structure*. Hearst (2009) quotes that the former refers to “the organization, the labels and terms used for the content items”, whereas the latter denotes ”the path that can be taken through the information structure”.

The following section deals with methods for gathering specific data elements from web documents for building structures that follow a pre-defined template or semantic structure. In contrast to text, which can be processed via techniques of natural language analysis, web documents follow a semi-structure, where structured data records are mixed with unstructured parts with natural language texts. Trying to extract the “wanted information” and filtering the “unwanted” is therefore the major goal of structured data extraction or information wrapping.

3.2 Structured Data Extraction

Information Extraction (IE) is a sub-field of Information Retrieval (IR) dealing with techniques to detect and extract data entities embedded in documents. Relevant and useful parts of data, i.e. specific information structures are analyzed and extracted more or less automatically utilizing methods of pattern mining, grammar-based extraction or machine learning. While information retrieval tasks deal with retrieving relevant documents from one or more document collections, information extraction gathers relevant information from documents.

In the case of semi-structured web pages, structured as well as unstructured parts co-exist with noise, which is the non-content bearing part of a document such as navigation menus, advertisement, etc.

The structured parts embedded in web pages originate widely from relational databases or other structured data sources. Usually, such data is published to the web losing its original schema information, such as, type of data structure, attribute names, etc. Grumbach and Mecca (1999) were the first to identify this obstacle as the “lost schema” problem, hindering the use of automatic methods for information processing on semi-structured web data. Since then, the variety and types of data sources have changed beyond the relational model and therefore, more complex data models behind data structures propagated to the web are possible. In summary, information extraction allows structured data from unstructured or semi-structured data sources to be populated.

In the following, the basic concepts and methods of information extraction from semi-structured documents are introduced. Extracting information entities via methods of natural language analysis and processing are, therefore, not the focus.

Extraction Wrappers

Information extraction from semi-structured web documents can be distinguished in data-driven and structure-driven information extraction. In the latter approach, specialized extraction programs – called wrappers – for web data can either be written manually using extraction grammars, e.g. using regular expressions or by applying supervised and unsupervised methods. Writing wrappers using extraction grammars manually presumes expert knowledge and is therefore time-consuming and error-prone. On the other hand, experts are able to program complex extraction grammar rules

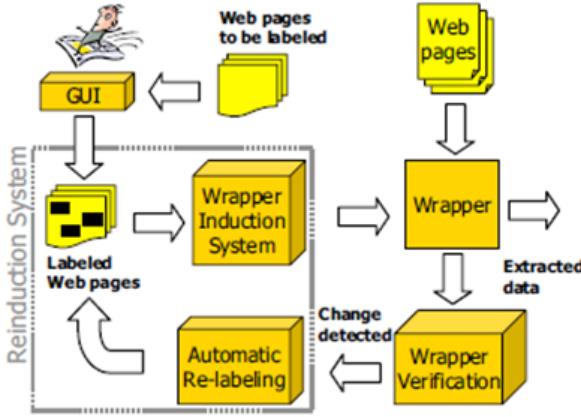


Figure 3.1: Wrapper life cycle (Lerman et al., 2003).

for gathering highly complex data structures, which is not covered by state of the art methods of AI and ML yet.

Research has tried to deal with some of the described problems by developing new methods and systems for supervised and unsupervised wrapper generation. Laender et al. (2002) identify the degree of automation, robustness and the quality of the extracted data (Laender et al., 2002) as the most significant criteria for information extraction systems.

Figure 3.1 illustrates the *wrapper life cycle* running through the labeling, induction/generation, extraction and verification steps and in case of change detection, to the re-labeling and re-induction steps. In case of structural changes, wrappers may fail partly or completely and not be able to deliver the target data entities. In practice, structural changes occur more seldom than changing contents. Therefore, wrappers are only affected if the website structure changes rapidly, making it necessary to re-generate the wrappers.

Specially designed extraction languages and grammars can be used to assist the users in constructing wrappers for semi-structured web documents. The Minerva system (Crescenzi and Mecca, 1998) allows to define grammars in EBNF (Extended BackusNaur Form) style. For each document, a set of production rules are defined. Huck et al. (1998) developed a similar approach (Jedi) allowing to write wrappers using attributed grammars. The resulting rules are evaluated utilizing a fault-tolerant parser to cope with ambiguous grammars and irregular sources, which is well-suited for semi-structured data sources such as HTML web pages.

As a result, languages and grammars are flexible and allow data records to be extracted with high precision. However, they bring the burden of “programming” the extraction rules that need to be rewritten in case of struc-

tural changes in the source documents. Fortunately, such extraction grammars can be generated by applying semi-supervised methods or learned, using machine learning techniques. Exploiting the capabilities of flexible grammar-based extraction in a semi-supervised system can be beneficial as will be elucidated in Section 6.3.

Wrapper Induction

Wrapper Induction, a generic supervised machine learning technique for inducing wrappers from given sample data such as data items or web pages was first described by Kushmerick et al. (1997). The learned wrappers are able to extract data from a set of “similar” web pages that contain data records from the same (structural and semantic) class. A formal definition of the wrapper induction problem is given below:

Definiton 3.1. (*The Wrapper Induction Problem*)

For a given set of sample web pages p_1, p_2, \dots, p_n learn a wrapper w for the information source that generated the pages.

Induction:

“Task of generalizing from labeled examples to a hypothesis (a function) for labeling instances of attributes to be extracted.” (Kushmerick, 1997)

$$\text{Wrapper } w: p \rightarrow (a_1, a_2, \dots, a_k)_t,$$

where $k = 1, \dots, K$ and $t = 1, \dots, T$ with K : total number of attributes, T : total number of tuples.

The following definitions give a precise definition of the more general wrapper generation problem and the use of semantics in the creation process.

Definitions

The problem of generating a wrapper for web data extraction can be stated as follows:

Generate wrappers that are highly accurate and robust, while demanding as little effort, e.g. user intervention as possible to develop them.

Starting from this definition the terms syntactic wrapper, semantic tuple/structure and semantic wrapper can be defined as follows according to the definitions given by Arjona et al. (2002):

Definiton 3.2. (*Syntactic Wrapper*)

A syntactic wrapper is a function $W : P \rightarrow T$. Given a web page P , it gives back a tuple with the information of interest.

Definiton 3.3. (*Semantic Tuple/Structure*)

Let L be an ontology, a semantic tuple T_s is the result of properly associating the information in the tuple with concepts defined using L .

Definiton 3.4. (*Semantic Wrapper*)

A semantic wrapper is a function $W_s : P \rightarrow T_s$. Given a web Page P , it returns a semantic tuple with the information of interest.

In automatic (unsupervised) data extraction, a wrapper is generated from the data without human intervention. The assumption is that websites are usually encoded using a few templates that can be identified from a few reference documents using “repetitive pattern” mining. Such regular structures can be detected by using suitable similarity measures (Gusfield, 1997).

String and tree alignment techniques (Simon and Lausen, 2005, Zhai and Liu, 2005) are common methods to generalize the extraction patterns found in order to be able to scale to other unseen similar web pages to extract data from. In practice, an important trade-off between the degree of automation of a tool and the flexibility of the wrappers generated by it can be observed.

3.2.1 Supervised Wrapping

Supervised wrapper creation approaches can be divided into supervised wrapper generation systems and supervised wrapper learning. The former allows wrappers to be generated by providing assistance in the wrapper creation process, while in the latter approach, e.g. wrapper induction, users only provide samples of the data to be extracted. The systems then learn wrappers for extracting data from unseen or new web pages or sites. In supervised wrapping the extraction target is specified explicitly.

A pioneer in applying machine learning methods (see Section 2.5) for learning wrappers was Nicolas Kushmerick. He defined various classes of wrappers that could be induced from labeled examples. In his approach (WIEN), delimiter-based extraction rules (Kushmerick, 2000) are derived from a given set of training examples. The learned extraction knowledge structures are formally equivalent to (possibly stochastic) regular grammars or finite state automata or transducers (Muslea et al., 1999). In general, the

introduced techniques do not rely on linguistic constraints, but rather on formatting features that implicitly delineate the structure of pieces of data found. The pages are assumed to have a predefined structure, and specific induction heuristics are used to generate specific wrappers. For instance, if the pages have an HLRT structure (i.e. have a head, a body containing flat tuples of data delineated by a left and a right component to be extracted, and then a tail), an HLRT wrapper is generated. Such wrappers, like the one used in WIEN – do not deal with nested structures or with variations typical of semi-structured data. This approach was extended further by the Stalker (Muslea et al., 1999) system, which expresses hierarchical extraction wrappers as trees in which internal nodes represent lists of records and leaves represent single fields. The system extracts information by descending the tree to successively refine the document segment to be extracted. At each node, boundaries are defined by disjunctions of so-called linear landmark automata, finite state machines that recognize sequences of tokens, token classes and wildcards. These automata are intended to consume prefixes and suffixes of the desired segment and are learned using an incremental covering algorithm (Liu, 2007b). In SoftMealy (C.-N. Hsu, 1998), wrappers are specified by so-called contextual rules associated with transitions in a finite-state transducer. A contextual rule is a disjunction of token sequences that marks the inside or outside of a field boundary. The states correspond to fields.

Visual assistance in wrapper generation is provided by systems such as W4F (Sahuguet and Azavant, 1999), Lixto (Baumgartner et al., 2001) and Thresher (Hogue and Karger, 2005). The W4F toolkit allows for semi-automatic building of wrappers in 3 phases: retrieval, extraction and mapping, for which rules can be defined. It uses its own HTML Extraction Language (HEL) and provides graphical wizards to assist the extraction process, i.e. writing of extraction rules in HEL. In XWRAP (Liu et al., 1999), the user highlights important regions and semantic tokens on a page and the system creates corresponding extraction rules.

OLERa and Thresher accept a rough example from the users to generate extraction rules, in contrast to working with complete and exact samples. OLERa (Chang and Kuo, 2004) is able to learn data from web pages with single data records and Thresher uses tree alignment between the subtrees in DOM to create a generalized wrapper for the selected samples or fragments.

Finally, an approach of instance-based learning for extracting web data was described by Zhai and Liu (2007). Their method is based on matching instances of structured objects via prefix-/suffix similarity of data items

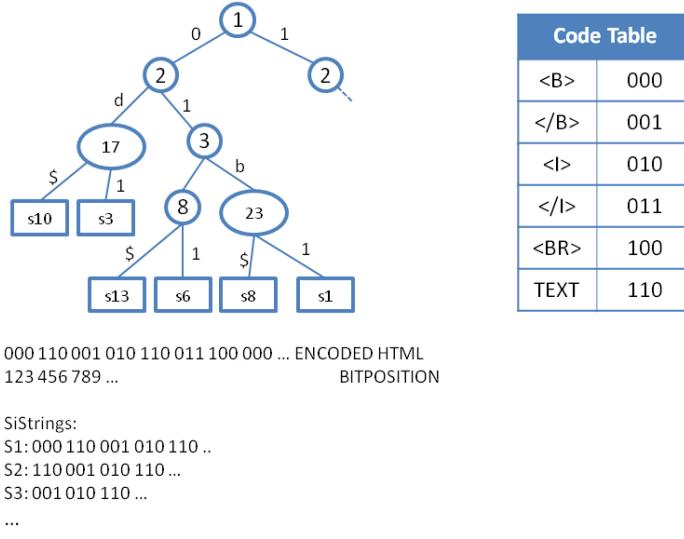


Figure 3.2: PAT tree representation of HTML code.

embedded in the HTML code. In order to ensure flexible wrapper generation and minimizing labeling effort, active learning (Moens, 2006) is applied to create the templates to be matched. Although this approach achieves high accuracy in extracting data items in product pages, list pages or nesting is not supported by this approach.

3.2.2 Unsupervised Wrapping

Automatic or unsupervised approaches do not rely on labeled training data and user interaction for creating wrappers. They rather exploit the intrinsic structure of the data by detecting data-rich regions from where data is extracted. If the extraction target is not specified by the users a priori, problems of ambiguity can occur if several schemas comply with training pages. As a result, a lot of unwanted data can be extracted making it necessary to apply a few post-processing steps to restrict the obtained data sets. The IEPAD approach developed by Chang and Lui (2001) was one of the first systems where learning wrappers is solved using repetitive pattern mining by utilizing a PATRICIA²⁷ tree - a binary suffix tree. Since a PAT tree (Figure 3.2) only stores exact matches, the classic center star algorithm (Gusfield, 1997) is applied to align multiple strings which start from each occurrence of a repeat and end before the start of the next occurrence. A generalized pattern using signatures is used to denote the template that is able to

²⁷Practical Algorithm To Retrieve Information Coded In Alphanumeric (Morrison, 1968)

process all (relevant) candidate data records. In a post-processing step, the user has to mark which data items are relevant and eventually add semantic markup. In the following example, the encoding of the popular “congo code” (**Congo<I>242</I>
 Egypt<I>20</I>
**) is given.

```

<BODY>
  Books of:<B>#PCDATA</B>
  ( <IMG src=.../> )?
  <UL>
    ( <LI><I>Title:</I>#PCDATA</LI> )+
  </UL>
</BODY>

```

Figure 3.3: Wrapper based on Union Free Regular Expressions.

The Roadrunner (Crescenzi et al., 2001) system solves a page-level extraction task and works with multiple input pages, while other approaches mainly perform record level extraction and process single web pages. In DeLa (Wang and Lochovsky, 2003), no user interaction is required. It allows the extraction of nested objects and operates in two steps; by firstly detecting data-rich regions, and secondly extracting pattern for the wrapper generation. DEPTA/MDR, like IEPAD and DeLa are only applicable to pages with two or more data records and single web pages. RoadRunner works by comparing the HTML structure of two or more given sample pages belonging to the same “page class”, generating as a result a schema for the data contained in the pages. The wrapper is progressively refined trying to find a common regular expression for the two pages. This is done by solving mismatches between wrapper and the sample. Generated wrappers are represented in the form shown in Figure 3.3. In the example, the ? operator indicates optionality, (...)+ repetition of an HTML pattern and #PCDATA string pattern generalization.

An important issue in automatic extraction is how to distinguish data tokens and tokens that belong to the (formatting) template structure. The general assumption made is, that HTML tags belong to templates, while the others are regarded as data tokens. The selection is left to the users. In addition, Roadrunner assumes that other matched tokens are also considered as part of the template structure. Other unsupervised approaches, such as, VIPER (Simon and Lausen, 2005), exploit the visual information between data regions to detect separators between data regions and apply repetitive pattern mining and generalization, such as, (multiple) string alignment the same way as described previously.

3.2.3 Discussion

While writing extraction rules manually applying wrapper programming languages (Huck et al., 1998) can be very effective for covering complex data structures inside HTML, they require expert knowledge and writing grammars manually can be a complicated and error-prone task. Machine learning on the other hand, needs appropriate training data for learning wrappers, which may also be ineffective, while scalability beyond a list of web pages can't be ensured either. Therefore, interactive approaches with more or less human interaction (Chang and Lui, 2001, Crescenzi et al., 2001) are seen as a good compromise to using fully automatic wrapper tools, which entail further post-processing steps for filtering out irrelevant data.

Consequently, the previously discussed main methods for structured data extraction subsumed as wrapper induction (Section 3.2.1) and automatic structured data extraction (Section 3.2.2) have to deal with considerable challenges due to training data for learning wrappers, which has to be created at great cost, while at the same time generalization and scalability of the described tools can't be guaranteed, making it necessary to provide additional amounts of training data. Automatic structured data extraction on the other hand produces too much unwanted data, while filtering for separating the “wheat from the chaff” brings additional costs.

Re-inducing non-working or invalid wrappers has been dealt with in the related work part (see 3.1). Liu (2007c) propose using active learning to minimize labeling effort which is a major drawback in wrapper induction.

3.3 Semantic Layering

Models and methods for semantically layering (semi-)structured web data have to be grounded into models which exploit important characteristics of hypertext, i.e. levels of structural markup (see Section 2.1.1), the underlying data representation models and visual characteristics, as introduced in Section 2.1.2.

In contrast to automatic annotation approaches applied to unstructured sources that rely largely on text analysis and natural language processing techniques, semantic tagging of rather structured sources using ontologies for example, is accomplished by overlaying semantic information to structured data entities/records by exploiting structural characteristics of the HTML DOM tree (Hogue, 2004), e.g. via facilities to select and mark examples of a particular class (see examples in Chapter 2). In practice,

overlaying is integrated in dedicated user interfaces that provide graphical assistance.

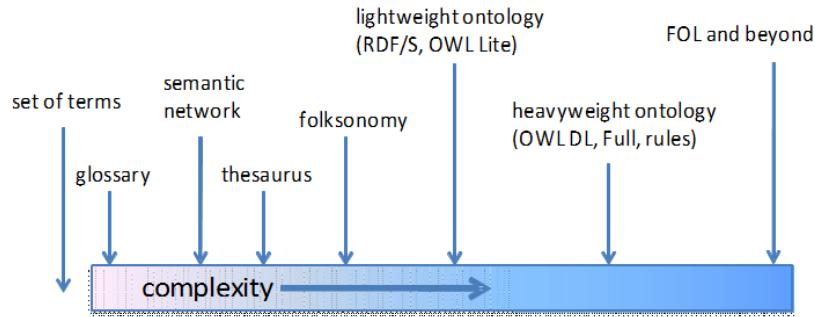


Figure 3.4: Ontologies organized according to level of semantics (Mika, 2007)

Figure 3.4 illustrates different levels of complexity for modeling semantics. Simple forms based on terms from an uncontrolled or controlled vocabulary represent convenient ways to label structured web data, while modeling approaches based on explicit and formal semantics are powerful but costly. To give an example, simple data fragments on a web page, such as bibliographic information, address, etc. can be annotated using classes from established controlled vocabularies such as DC, FOAF, SKOS, etc. by assigning semantic labels (type of object) and properties, e.g. `dc:title`, for matched data records using context-menus in a browser. Hereby, simple domains or social networking interrelations can be modeled by using appropriate linking schemes and relation properties, such as `person_a foaf:knows person_b` in triple notation.

In the following, a short description of ontology-based approaches for semantic annotation, methods for exploiting user tags and inherent semantics in folksonomies for creating visual information retrieval structures and interfaces is given.

3.3.1 Ontology-based Annotation

Ontology-based annotation is a process of adding semantic markup to data by using concepts and relations from a (domain) ontology. The main advantage of this approach is that meaning is grounded in a model-theoretic definition with well-defined semantics (see Section 2.2.2), which can be exploited for semantic search, information integration, etc. As manual anno-

tation using ontologies based on e.g. RDF-S or OWL is a non-trivial task and requires expert-knowledge, systems that support experts, knowledge engineers and other users have been developed.

Existing ontology-based approaches differ with respect to the degree of automation, the complexity of the used knowledge representation, e.g. free text + RDF-S or relational meta-data (Price and Sherman, 2001), XML, RDF-S, OWL, etc., the support for unstructured and structured knowledge, e.g. text, HTML, XML, JPEG-2000, MPEG-2, etc. and the kind of graphical support/assistance and interaction metaphor of the user interface (Kahan and Koivunen, 2001, Handschuh and Staab, 2003, Quint and Vatton, 1997, Heflin and Hendler, 2001).

In general, limitations exist with respect to the degree of user intervention at certain steps and the need of experts, such as, domain specialists or knowledge/ontology engineers.

3.3.2 Social Semantic Tagging

Early collaborative forms of semantic tagging aimed at sharing annotations largely between a community of researchers, which have worked on similar projects and fields. In contrast, semantic tagging approaches motivated by the social web movement aim to reduce the need for experts, while implementing easy-to-use annotation processes based on a social network of users in a collaborative tagging process. Hence, low cognitive overhead for assigning meta-data to web resources is regarded as being crucial reducing the burden of manual creation and management of semantic information.

Enhanced models of collaborative tagging allow data in existing or newly created web resources to be associated with meaning, e.g. concepts from an ontology, in order to enrich user-provided flat tag structures (which have problems related to synonyms, polysemy, etc. as described in Section 2.3.1). While on the one hand, social semantic tagging user interfaces have to adopt support/assistance in order to overcome limitations of free tagging, the underlying models have to deal with the semantic context of tagging.

In order to ensure that tagging data created this way is grounded into Semantic Web standards, semantics are included in the tagging process by exploiting external sources, e.g. WikiPedia or lexical-semantic word nets and standardized ontologies. Basically, the idea is to allow to state semantic assertions about resources, tags and their relations. One possibility is to use semantic keywords or terms that refer to ontology concept for describing a specific property of a given resource (Marchetti et al., 2007) via RDF

statements (Figure 3.5), or make use of the linked data principles (Passant and Laublet, 2008, Heath and Motta, 2008), i.e. using URIs to existing resources to define machine-understandable semantics.

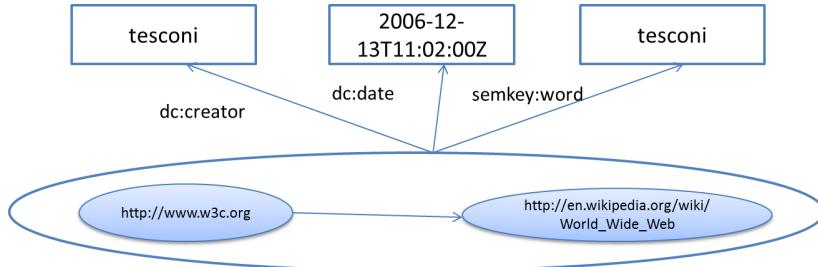


Figure 3.5: RDF graph of an example semantic assertion.

Furthermore, the tripartite model that was described in Section 2.3.1 can be extended by introducing different types of meaning. Passant and Laublet (2008) introduce local meanings for each tagging action, while the global meaning for a tag is defined through social aspects related to the users that used a respective tag. In addition to this, users are able to tag the tags as well as their relations in order to create semantic associations between tags as described in an approach by Tanasescu and Streibel (2007). The following example shows the *knowledge path* between “wheel” and “vehicle” expressed as a list of triples in order to express semantic associations between tags:

$\langle \text{"wheel"}, \text{"vehicle"} \rangle = [(\text{"wheel"}, \{\text{"singular - of"}\}, \text{"wheels"}), (\text{"wheels"}, \{\text{__}\}, \text{"car"}), (\text{"car"}, \{\text{"is - a"}\}, \text{"vehicle"})]$

Knowledge paths created this way can be compared with respect to similarity in order to unveil new and implicit associations (serendipitous discoveries) between tags and resources.

3.3.3 Exploiting Tag Semantics for Retrieval

In contrast to introducing formal semantics to web data, folksonomy systems have adopted a bottom up approach on the social web, allowing data and (implicit) semantics provided by the underlying social network of users to be held. As a consequence, semantics do not have to be imposed in a post-processing step, but, can be exploited immediately for exploring stored (and linked) resources.

Semantic Analysis of Tagged Data

Semantic analysis of folksonomy data (see Section 2.4.3) allows implicit semantics from the tripartite graph of users, tags and resources for structuring and organizing the underlying information spaces to be extracted. Besides that, enhanced forms of social semantic tagging via dedicated annotation interfaces allow richer semantic tag structures, e.g. taxonomic relations, explicitly – beyond flat tag lists as shown in the previous section to be obtained.

Although the uncontrolled nature of folksonomies inevitably leads to the described problems with respect to inaccurate and incomplete results, broad folksonomy systems are characterized by high recall rates for relevant documents, as resources can be annotated by multiple users. This is due to the variety of assigned tags to particular resources, thereby increasing the probability that a common understanding and a shared vocabulary to describe the individual resources emerges. A high recall rate is particularly important for explorative search and serendipitous browsing (Marchionini, 2006), where the goal is to maximize the number of potentially relevant items retrieved rather than precision, which is targeting at reducing the number of non-relevant items retrieved. Bear in mind that web search engines, e.g. Google are highly tuned towards precision of the first n (< 100) query results. Trying to explore results after the e.g. hundredth page makes little sense, as semantic cues (like tags in folksonomy systems) that could help to relate the results to each other are hardly available.

3.4 Visual Information Exploration

The simplistic interface approach employed in web search engines or (visual) retrieval systems in general is well-established and efficient at first sight. Nevertheless, research projects of big web search companies, e.g. the former Google's Wonder Wheel²⁸, reveal that existing forms of visualization and interaction with ranked search result lists are not sufficient looking at the complexity and increasing amounts and types of information on the Web. In fact, a pressing need for enhanced visual interfaces to support novel forms of (semantic) interaction can be observed, manifesting in a trend from traditional *navigational* or *keyword-based search* towards *explorative* and *semantic search*.

²⁸<http://www.hmtweb.com/marketing-blog/google-wonder-wheel-contextual-targeting-tool/>

A crucial parameter of basic keyword search systems is the relevance of each result, which is generally mapped to rank order in the presented result list. Additional information such as links to other (probably) similar or related resources are placed in (spatial) proximity of every single result entry.

Interface structures that go beyond this basic approach allow the exploration of the interlinked information structures via suitable visual metaphors in 2D and 3D and additional navigation capabilities in order to enhance user experience in information discovery and allow for efficient interaction with search results.

The requirements for dealing with large amounts of data (“big data”) increasingly dictate a process where the parts of interest first need to be focused on before exploring the restricted information space following semantic paths, i.e. semantic relatedness complementing and enhancing statistical properties such as the TF-iDF metric.

It is clear that exploiting distinct levels of aggregation through a process like faceting will enable more degrees of interaction beyond statistical relevance. Hence, recall gains a higher impact in semantics-driven search and discovery of information. In this sense, exploring information via ontological semantic concepts and relations expressed as semantic paths as e.g. RDF triples, allows the information space to be focused and restricted and allows distinct levels of aggregation to be exploited, hence, adding semantic dimensions to the search capabilities of information retrieval systems.

On the social web, a semantic dimension to search and interaction comes into play by means of user-created folksonomies and visual metaphors to access relevant web information sources. In folksonomy systems, enhanced search capabilities are provided through tag-based search by exploiting implicit semantics from folksonomies and through new forms of visual interaction, for instance the tag-cloud used as a retrieval interface. Not to forget the social dimension of search, which is determined by the structure of the hypergraph introduced in Section 2.3.1 directly affecting the social relevance of information. As a result, the notion of relevance itself is subject to change.

Heath and Motta (2008) state, that “relevance was for a long time constrained to a global, topical relationship between a query and a set of items”. In contrast, recent studies (Teevan et al., 2010) show that users increasingly prefer to gather information of rather personal value and relevance, found via unexpected or serendipitous discoveries (Ottoson, 2008). Furthermore, Oblinger and Oblinger (2005) provide evidence that the information behav-

iors and expectations of the “net generation” are different, as “they multi-task and want to interact with dynamic information resources via *fluid* user interfaces that allow lookup, learning and investigation to be conducted at the same time”.

In the following sections, a review of existing 2D and 3D user interface approaches and techniques for navigating large information spaces, is provided.

3.4.1 2D Semantic Retrieval Interfaces

In the last two decades, several ways to visualize large scale web data in 2D have been explored. Besides hyperbolic geometries that have been used to display large hierarchies (Lamping et al., 1995), animated exploration of dynamic graphs with radial layout (Yee et al., 2001) or exploration in form of large node link trees (Plaisant et al., 2002) have been investigated. Furthermore, different forms of tree maps (Bruls et al., 2000) and graph-based layout techniques based on force directed visualizations have been developed. For example, Hoare and Sorensen (2005) describe an information foraging tool that uses a 2-dimensional proximity-based visualization.

Faceting vs. Graphs

Semantic data visualization approaches mostly adopt graph-based approaches, providing benefits for aggregation and filtering of related information for given resources. The widely applied faceting approach for search interfaces allows data at different perspectives or views to be accessed, while missing support for graph-based navigation. The existing solutions show that it can be a powerful interface for focused domains. Nevertheless, Mirizzi et al. (2010) argue that though, “faceted browsing for RDF datasets improves usability compared to keyword searches by providing a better information lookup”, they become difficult to use with growing number of presented results. In addition to this, it is not possible to exploit implicit relations, i.e. relations which are not explicitly provided by the used data sets.

Recent retrieval interfaces utilize novel forms of visualization and interaction metaphors enhancing the basic visual models in order to offer an intuitive and efficient visual access to various types of web content, e.g. user-generated content, linked data, etc.

Exploiting Explicit and Implicit Semantics

The Semantic Wonder Cloud (SWOC) (Mirizzi et al., 2010) interface supports explorative search for DBpedia by utilizing new associations between data instances exploiting additional knowledge sources, e.g. web search engines or folksonomy systems. Although their hybrid approach primarily relies on semantic similarity for exploring the knowledge space, they integrate a text-based IR approach as well. Visual support is provided by a “point and click” interaction metaphor in order to provide a more convenient user experience. A similar user interface approach is eyePlorer²⁹ which exploits information from WikiPedia to build a knowledge graph that can be explored.

*Alexandria*³⁰, a sub-project of the *theseus* research program, is able to uncover and visualize important semantic relations and properties of concepts related to search terms. The interface provides domain specific visualizations of interrelations, such as, connections between entities, people, countries, etc. obtained from the underlying knowledge network.

The sig.ma system - a semantic search and browsing mashup - associates and finds search terms with instances on the linked web of data or other semantic knowledge sources by employing a multi-faceted approach. Its user interface allows simple interactions in order to constrain results by adjusting specific properties and their values (show/hide/remove property/value) for distinct sources (approve/reject). Browsing related or linked resources is achieved by following links to other related instances stored in the object of the returned [subject, predicate, object] triples. The obtained triples for a given search query are based on the *sindice* (Tummarello et al., 2007) semantic web index, which collects and compiles semantic meta-data distributed across the web ranging from search engines, social web sites to governmental sites, etc. The gathered information is of the type contacts, events, reviews, etc. based on RDF, RDFa and Microformats.

3.4.2 Retrieval Interfaces for Folksonomy Systems

For folksonomy-based retrieval systems, in general, two basic types of existing interface approaches – depending on which step(s) of the information seeking process at hand are addressed – can be distinguished. In the first type, an initial search tag (e.g. manual input or clicking on a tag in the cloud) is required, which is visualized together with the retrieved results

²⁹<http://de.vionto.com/show/>

³⁰<http://alexandria.wefind.de/>

after the query execution. In the interface, the relationships between the search tag and other tags that are related to the retrieved resources are illustrated, helping to detect and resolve ambiguities, refine queries, etc. The “related tags” list is such a widespread interface element based on up to twenty tags that appear most frequently together with the search tag(s). The second type is inspired by tag clouds, which allow a broad overview of the tag space in order to serve as a starting point for exploration, etc. Here, semantics helps to enhance clarity of such overviews by displaying which tags are similar or related. Hence, the associated resources might be worth of being explored.

Most of the described interface approaches that make use of user tags have focus on the social aspect of the data, which is reflected through the connections of the social actors over similar data, e.g. in PearlTrees³¹. Other approaches, such as eyePlorer³² provide a partitioning scheme for basic categories, e.g. place, person, organization, time, etc. using radial layout and a “cake” metaphor. The layout structure exploits structural information from underlying web data sources, e.g. WikiPedia for linking related items.

Clustered and Topical Tag Layouts

Extracting and re-using semantic relations from collection of tags for given resources allows (semantic) aggregations to be formed by utilizing e.g. clustering, grouping, segmentation, etc. according to a similarity metric (see Section 2.4.3). In general, clustering can be based on one single shared feature, e.g. term similarity, shared significant phrases (Käki, 2005), etc. or multiple shared features. The most simple form is clustering based on inter-document similarity obtained from the weighted document vectors in a bag-of-words model (Baeza Yates and Neto, 1999).

Although Hearst and Pedersen (1996) suggest the application of clustering methods to search query results in order to be effective, one important issue resulting from the unsupervised nature of clustering on term-vectors was identified; topics may be placed at varying levels of description, e.g. categories that are more general could appear together with more specific ones.

Such “inconsistent levels of description” in the organization of information entities – be it tree-like or a graph structure –, are rejected by users, as it was shown by usability studies conducted by Chen et al. (1997).

³¹<http://www.pearltrees.com/>

³²<http://eyeplorer.com/>

Regarding tag cloud layouts, topically arranged tag cloud layouts were first described by Hassan-Montero and Herrero-Solana (2006), which used a layout similar to classic representations where in each row of the tag cloud different tags from different main topics are placed. Furthermore, Fujimura et al. (2008) presented an overview-like representation of large scale tag sets, where the tags are mapped on a scrollable topographic image with central tags located in the highest regions and related or more specific tags placed around in lower regions.

Other approaches build clusters automatically by producing faceted hierarchies (Stoica et al., 2007) or assign documents to predefined categories (Sebastiani, 2002).

Semantic Arrangement of Tags

Schrammel et al. (2009) conducted a series of experiments with topical/semantic arrangements of tags compared to other layouts, e.g. alphabetic, random, etc. Their results showed that semantic layouts can improve search performance for general search tasks compared to other layout forms, while the results are not as good for specific search tasks. Besides the general conclusion that many aspects of such approaches are not understood, the clustering algorithm that was used seems to have a major impact on the results interaction. Further experiments dealt with task-related measuring of tag cloud performance in visual exploration and perception (Lohmann et al., 2009), which showed significant differences for different layouts. Hence, dedicated tag cloud layouts must be designed depending on the specific task and user objective .

Enhanced semantic representation of folksonomy data was achieved by determining the relatedness of tags statistically by analyzing their tagging context considering the entire set of annotations. In his work, Mika (2005) describes a method for transforming the original graph representation of the complete annotation structures for calculating tag co-occurrence and creating a tag co-occurrence graph, which contains the co-occurrence counts for each pair of tags. In order to obtain more balanced results based on tag similarity, calculating the relative co-occurrence applying different similarity metrics is reasonable.

Hierarchical structures from folksonomy data can be extracted by applying algorithms described by Grahl et al. (2007) and Gemmell et al. (2008), which provide a basis for more structured browsing or personalized navigation. In Specia and Motta (2007), a non-exclusive agglomerative clustering

technique is described in order to map groups of tags to ontological concepts. Further work makes use of a divisive k-means algorithm (Hassan-Montero and Herrero-Solana, 2006) in order to provide a semantically ordered tag cloud or suggest clustering the tag space using graph-based clustering, splitting the co-occurrence graph where the edges are weakest (Begelman, 2006).

3.4.3 Visual Exploration in 3D

Interaction in 3D environments at first glance is considered to be a natural environment for humans. The additional degrees of freedom allow the visualization of more information on a restricted space and interaction with more complex structures and their relations is possible. Although 3D applications have increasing hard- and software requirements, the available computing power and memory capacity have currently reached a level where more demanding applications can be realized. The main difficulties are concerned with the lack of experience of (most) users with forms 3D interaction. In addition to this, no 3D design standards exist, which makes it difficult for system designers to develop systems beyond “island” solutions.

For many years several research studies for comparing 2D and 3D visualizations have been conducted. Sebrechts et al. (1999) examined different visualizations of search results. They compared text, 2D and 3D visualizations. Their results reveal the fact that 3D navigation overload for users is not easy to neglect, while experience plays a central role. It is also stated that when going through a certain learning or training process it is possible to achieve faster and more efficient results than with 2D visualization. Furthermore, in 3D visualizations dealing with business information, Schönhage et al. (2000) show that despite the fact that more effort is needed to train users of the comparative 3D tool than with a known 2D variant, complex 3D visualization provide high benefit for trained users. Robertson et al. (1991) show that exploring large information spaces can be effectively fulfilled in a 3D “cone tree”. For web content visualization, Risdan et al. (2000) which compared 2D vs 3D and an additional list representation suggest the use of combinations of 2D and 3D. Cockburn and McKenzie (2001) investigated 3D for document management and could not observe significant differences, while users find a 3D user interfaces more attractive. Other studies by Cockburn and McKenzie (2002), Perez and Antonio (2004) showed that 3D design must be well tailored to the specific tasks in order to be effective. In general, recent developments acknowledge 3D interfaces as giving more “joy” in combination with new forms of input methods such as touch, body

movement tracking, etc.

3D Applications for Information Browsing

Applications for 3D information browsing allow interaction with collections of data such as web pages, search results, pictures, videos, etc. by utilizing different forms of 3D visualizations and navigation. Compared to interactions in the two-dimensional space, interaction is overall accomplished based on 6 degrees of freedom (DOF) – using translations along the x, y and z axis and rotation around x, y, z. As a result, more interaction capabilities can be mapped and more data visualized in a compact representation. Today, 3D visualizations on 2D screens are common, i.e. in reality this is often 2.5D. One of the properties of 3D is that users can interact with visualized objects in a more immersed way.

Several research and commercial applications use 3D information spaces to explore collections of 2D data, such as web pages, search results, pictures, or videos. Popular applications, such as, Cooliris³³ – a plugin for the Firefox browser – allow to visualize different forms of search results (primarily pictures and videos) horizontally along a 2D plane. Users are able to interact with the horizontally ordered items on the wall by flying along the wall. Furthermore, automatic zooming and camera panning is supported by restricting the movement along the axes. This prevents the wall from gliding out of the view port. Similarly, SpaceTime 3D³⁴ allows the visualization of search results in form of a stack in 3D. Users can interact and browse web pages in 3D, zoom to particular areas of a page, follow links, etc. The Sphereexplorer³⁵ provides a stack and wall visualization of web pages. It allows the accessing of web pages using translation along one of the axes from the center point and rotation along the axes x, y and z, while constraining translations and rotations.

Visualization of Semantic Information

One representative of an application for visualizing semantic information in 3D is OntoSphere (Bosca et al., 2007), a 3D visualization tool for ontologies. Users are able to interact with ontological concepts and instances via rotation, zooming and panning, in order to select and refine individual or compound entities. TagGalaxy³⁶ is a tool for visualizing tagged Flickr

³³<http://www.cooliris.com>

³⁴<http://www.spacetime.com>

³⁵<http://www.spheresite.com/>

³⁶<http://taggalaxy.de/>

images. It represents tags as planets, where similar tags are spatially arranged around the planet as satellites. Similarity is reflected in distance to a search tag, i.e. the home planet. The Sphereexplorer provides a stack and wall visualization of web pages. It allows interaction with web pages using translation along one of the axes from the center point and rotation along the axes x, y and z, while constraining translations and rotations as well. Other similar applications are the Giraffe Semantic Web Browser (Horner, 2008) and knowscape (Babski et al., 2002).

3.4.4 Search UI Concepts and Information Bias

Looking at existing search user interface concepts, the most common approach is to enter query terms and interact with a list of results, which provide a short-hand description of the retrieved contents through, for example, a title and additional information, which is extracted from the original document or data source. Utilities to help users in information discovery from textual sources are given by e.g. KWIC (keyword in context) or by creating simple abstracts by taking the few first lines automatically. Highlighting of query terms was shown to be an important aspect for being “eye-catching” (Landauer et al., 1993). Moreover, the series of experiments by Joachims et al. (2005) showed that in web search, users are strongly biased towards highly ranked results returned by widely used search engines such as Google. As a consequence of this information bias, the majority of (semantically) highly relevant results are ignored.

Semantic Relevance

In the list presentation, search results are widely ordered according to statistical relevance, while sometimes a graphical representation of the relevance score helps to hint to the relevance of the shown entities for the entered user query. Moving from statistical relevance of results to *semantic relevance* is one major benefit of knowledge-driven search approaches (Tran et al., 2008), which can be seen as complementary technologies for combating information overload. Use of implicit semantics from web documents was applied in search systems, e.g. via latent semantic indexing (LSI). Google for instance, has implemented a solution named Wonder Wheel based on LSI into their search interface. This search capability which allowed related terms (to the user query) extracted from the search results to be explored, has since been removed from the search engine and integrated into Google Ads³⁷.

³⁷<http://www.google.com/ads/innovations/ctt.html>

Towards Semantic Interaction in Retrieval Interfaces

In the previous sections it was discussed that with the immense growth of data on the Web and the resulting problems with information overload, the need for more interactive visual interfaces supporting semantic search and exploration (“from finding to understanding” according to Marchionini (2006)) becomes more urgent than ever.

This is true, particularly in the context of the social web, which proved to be beneficial for combating many existing problems of the web such by capturing contextual as well as relevance information from the social network. Hence, the social dimension of search³⁸ is of equal importance with the semantic context as many of the successful social networking applications show.

In order to gain a more informed view of the content of the result sets, Woodruff et al. (2001) experimented with the thumbnails of search documents. Efforts like this serve the purpose of understanding the content of the retrieved resources and the numerous aspects and topics of the result documents. In addition to this, forms of visualizations based on highlighting with different views and perspectives and a camera-style interaction via the zoom and pan metaphors, can help to focus the details of the results and retain context (Robertson et al., 1993).

For example, one of the early search engines which provided visualization of search terms in a map- or cloud-like form is the Quintura³⁹ search interface. The interface shows related terms close each other. As the search context is of great relevance, semantic information can be regarded as the key to understanding the content. In classic web search interfaces, semantic attributes of data are rarely exploited beyond query term or keywords visualization (Carpinetto and Romano, 1996). Furthermore, a visual access to retrieved content can be achieved using thumbnails of sufficient size together with important aspects of the underlying information source.

IR Evaluation vs. User Experience

The most common measure for judging the effectiveness of a search or retrieval system is *relevance*. In general, a binary classification scheme is used to classify retrieved entities in relevant or non-relevant. The objective quality and the effectiveness of the retrieval system is measured via the metrics that were introduced in Section 2.6. While in document retrieval, the clas-

³⁸<http://www.eurekster.com>

³⁹<http://http://www.quintura.com/>

sified entities are for example text, semi-structured or structured hypertext documents, etc., the information extraction community speaks of structured data objects, semantic structures, data tuples of a certain class, etc.

Evaluation measures that go beyond an absolute relevance measurement relate to user satisfaction or user happiness, which is a different perspective compared to the more global view depending on solely measuring retrieval effectiveness. Manning et al. (2008, pg. 151) states that user happiness not only depends on factors such as speed of response, relevance of the returned results, but is also affected significantly by qualitative factors, such as layout, clarity and responsiveness of the user interface.

Focusing on web browsing or navigation scenarios, what is assessed is the “browsing experience” using a mix of quantitative and qualitative metrics. Explorative search, as a consequence, can be regarded as an enhanced form of browsing. It is clear that the general methodology of retrieval evaluation cannot be applied to explorative or semantic search as an important assumption of classic IR, where the user’s information needs do not change during the interaction with search results, is no longer valid, as stated by Bates (1989), who developed a model of information seeking called “berry-picking”.

For that reason, Manning et al. (2008) applying an evaluation method for quantifying aggregated user experience based on a mix of qualitative, e.g. user interface design, interaction, etc., and quantitative measures, e.g. relevance, speed, etc. Therefore, user studies have been widely consulted for evaluating user satisfaction and search interaction based on prior selected tasks, appropriate metrics and methods of observation and interviewing.

3.5 Conclusion

With growing amounts of web data worldwide, search applications gained high impact. In fact, the “Google Generation” rarely knows any other method to directly access web documents. This results in a restricted view of information on the web which affects not only private usage, but also medium- and small-sized companies, research labs, governmental management and other organizations.

Trying to return useful and relevant links to related documents to the users, existing search engines exploit statistical properties of data by applying methods, such as, the TF-iDF scheme (Manning et al., 2008, pp.118), link analysis or data mining. Despite all enhancements, the inverted index (Konchady, 2008, pp.82) is still the basic data structure to be queried

by users of a search system. As there are still too many equally relevant documents for a given search query which can be returned, web search is confronted with the famous “finding a needle in a haystack” problem. This is why ranking search results is generally regarded as “magic” and the applied “fine-tuning” a well-hidden secret of the web search companies. For general purpose keyword-based search queries on the web, an approach that is highly tuned towards high precision of the first ten or twenty documents, is well suited, while for more complex and professional retrieval tasks where both high recall and precision are equally important, search on a conceptual or semantic level is needed. As an example, a keyword-based search-engine is unable to give exact answers to natural language queries, such as, “Who won the soccer world cup in 2002?”, “Which book was written by Remarque in 1928?” or “Show me the sales volume of Siemens in 2005!”. Systems capable of handling such query types are typically knowledge-based systems that rely on offline well-prepared data and knowledge models, e.g. geography, mathematics, etc. - as in the case of the computational knowledge engine Wolfram Alpha⁴⁰.

Besides the necessity of highly accurate results beyond document-level search, another important aspect is related to the topicality of information. In principle, web search-engines index and thus merely allow past versions of a web page to be queried, which is determined by a preceding crawling process. As a consequence, the results are influenced by the type, strategy and the capacity of the working crawlers. In his investigations, Lewandowski et al. (2006) identifies and analyzes this issue as the “freshness problem” of web search engines, resulting from frequently changing web pages, which affect not only retrieval, but also information extraction as will be discussed later in Section 3.2.

Semantic Retrieval and Extraction

A prerequisite for semantic search on the web is to extract and retrieve accurate and relevant pieces of information from semi-structured sources. However, the diversity and complexity of the structure and content of semi-structured documents show that hand-crafted rules for extracting data are less feasible for many web documents and different structural types encoded in the HTML sources. Nevertheless, for specific applications, such as, extracting data from medical sources or accessing published structured contents, where high extraction accuracy is crucial and errors can't be tol-

⁴⁰<http://www.wolframalpha.com>

erated, this approach is still widely used in industrial and professional applications in combination with assisting natural language processing tools in order to cope with large volumes of data. Further barriers for increasingly automated information extraction exist through structural as well as semantic changes in web pages that could occur at any time. As a result, depending on the application and use case at hand, the extraction method and additional costs resulting from labeling data must be evaluated a priori.

Semantic Annotation and Tagging

Another requirement for semantic processing is given by semantic meta-data encoding meaning of the respective pieces of information, which must be added automatically or applying a semi-automatic approach by exploiting the ability of humans to associate meaning with information. Besides allowing specific types of information to be queried, thus enhancing information retrieval and extraction beyond keyword matching, a major advantage of semantic meta-data is that related conceptual links can be followed for ensuring semantic interoperability of data distributed across heterogeneous sources and different domains. In this way, a city in Europe can be disambiguated from a city with the same name but in another region by analyzing the semantic context of information in the semantic neighborhood (proximity) graph by consulting the underlying ontology or other type of semantic network model used.

In the last two decades, researchers have tried to develop increasingly automated approaches in order to reduce authoring effort for creating semantic annotations and meta-data, which can be immense with a growing amount of data sources and raw data on the web. As previously shown, semi-automatic information extraction techniques with e.g. graphical assistance for creating ontology-based annotations can be utilized for this task.

Furthermore, automatic annotation approaches for textual sources can benefit from techniques of content analysis, e.g. named entity recognition (Popov et al., 2003), paragraph analysis, co-reference resolution, etc. in order to develop strategies to learn annotations without user intervention. In addition, learning from user-provided samples (Ciravegna et al., 2002, Etzioni et al., 2005) by analyzing the context of marked samples using e.g. similarity functions (Dill et al., 2003) can be applied to both - unstructured or structured sources.

Given the characteristics for structured data encoded in semi-structured web pages (in contrast to unstructured text), it becomes apparent that re-

lations – be it taxonomic or semantic relations – between structured entities must be widely provided by human users and cannot be derived from the intrinsic structure of the data itself, as in most cases the information is unavailable. Although most of the IE systems regarded are able to discover particular objects and their values, e.g. addresses, they fail to establish relations to other objects on the same web page, e.g. to a phone number correctly. In the literature this circumstance is called the "lost schema" problem.

Semantic Interaction

A major challenge for existing web retrieval systems besides automatic semantic enrichment, is interaction with the discovered and extracted information structures from a variety of web sources.

In addition to this, further challenging use cases of semantic retrieval are question answering and explorative search, and the understanding of the notion of relevance, which might be more personal in nature than assumed in the classic retrieval approach.

Furthermore, new interaction devices and modalities, e.g. touch input, together with increasing processing power for graphical applications, visual retrieval interfaces enjoy a revival as can be seen from the previously described semantic user interface approaches for 2D and 3D.

The next chapter gives an overview of the concept of this thesis, starting with a definition of semantic interaction and interaction metaphors in the context of web-based retrieval systems and the tasks involved. In subsequent sections, the introduced tasks are applied in two use cases of the web for enabling semantic interaction in web-based retrieval systems.

Chapter 4

Semantic Interaction in Web Retrieval

Focusing on use case scenarios in web retrieval, this chapter aims to propose and outline new forms of interaction with web data exploiting semantic information. In particular, this chapter gives a brief explanation of “semantic” interaction metaphors and their role in information seeking followed by a definition of semantic interaction in web retrieval. Essential tasks such as semantic layering, semantic mediation and semantic human-computer interaction are outlined.

The described tasks are employed and investigated later in two separate use case scenarios; web-based question answering in a knowledge-based dialogue system and semantic exploration of web resources in folksonomy systems. While in the first use case a semantic interaction approach based on an expert-created ontology is investigated, in the second use case, emergent semantics from folksonomies are employed in 2D/3D visualization and navigation for exploring (large) information spaces.

The semantic interaction framework outlined in Section 4.3 aims at describing the basic building blocks, a workflow, and a conceptual design for implementing exemplary use cases, which are elaborated in the case studies in Part II.

4.1 Interaction Metaphors

Human computer interaction (HCI) aims at “investigating the interaction and communication between humans and computers by designing, evalu-

ating and implementing interactive systems to be used by humans, and researching major related phenomena around new forms of interaction between human users and machines” (Shneiderman, 1987).

The use of metaphors in HCI has a long tradition and metaphors like “talking to computers” has been a vision since the invention of computing systems. What HCI metaphors generally aim at, is to imitate human-like interaction and understanding how to accomplish a range of tasks. In the early years of computing systems, first simple dialogue systems, e.g. ELIZA (Weizenbaum, 1966) focused on natural language processing as a means to achieve human like communication with machines based on the *conversation metaphor*. Although implemented via primitive pattern matching, ELIZA was taken seriously by its users as a form of artificial intelligence system. Since then, efficient solutions have been realized for lower levels of natural language processing, e.g. analyzing words, sentences, named entities. Due to the complexity and ambiguity across tasks, contexts and cultures, understanding of human language by means of higher level processing and reasoning remains a big challenge. Nevertheless, first research prototypes of complex dialogue systems begin to make use of text understanding, world knowledge, semantic processing, etc. in order to resolve some of the limitations reported so far.

However, as the invented systems and applications from the last two decades show, graphical systems based on the *direct manipulation metaphor* represent the state of the art in human computer interaction. Simple metaphors like “point and click”, “drag and drop” or “pan and zoom” have been adopted successfully into visual interfaces for enabling intuitive and efficient interaction with a computing system (Marcus, 1994).

Looking at HCI metaphors more systematically, three basic types; *functionality*, *interface*, and *interaction* can be distinguished according to a classification scheme by Fineman (2004). He describes functionality metaphors as artifacts which encompass the user’s expectation of an application of a regarded system, interface metaphors as a means for allowing users to perform the tasks within a functionality metaphor and interaction metaphors – which are focused herein in the context of information retrieval (IR) – as the underlying general concepts that determine the user-performed actions and transport a generalized relationship that is valid in numerous contexts. In Figure 4.1, the three types of metaphors are illustrated via the email example.

Interaction metaphors have a crucial impact on information search and discovery in IR systems, which are based on one of the major information

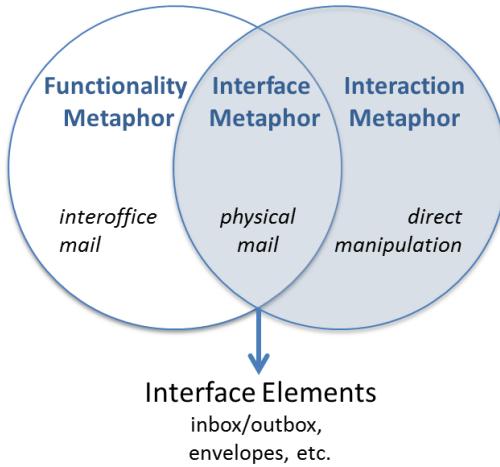


Figure 4.1: Metaphors in HCI (Fineman, 2004, pg.10).

seeking models (Belkin, 1980). More precisely, metaphors play a central role for expressing the user's contemporary information need, query processing and results interaction at the same time. Hence, a successful adoption of interaction metaphors has influence on ease of use, understandability and complexity of such systems.

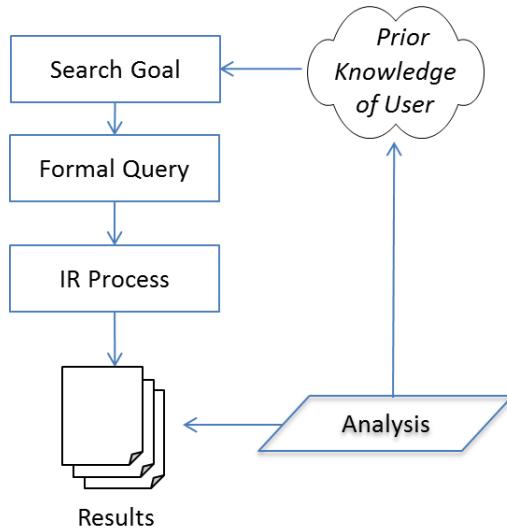


Figure 4.2: Enhanced IR model.

Moreover, Carroll and Thomas (1980) suggest also considering psychological aspects when designing user interface metaphors and implementing visual interfaces for humans. Understandability, being one major issue, is directly connected to meaning and semantics, imposing a semantic dimension onto the information seeking and discovery process. It might also

enable a meaningful, understandable and transparent user-machine interaction, which is outlined in the definition given below:

“Semantic Interaction (SI) in the context of web retrieval can be described as a form of Human Computer Interaction (HCI) with information entities, e.g. search results or information fragments from web pages, etc. based on semantic information. SI is enabled by means of Semantic Interaction Metaphors (SIM) by introducing a semantic dimension to the process of search and exploration.”

Figure 4.2 illustrates the general information seeking process, starting with the formation of a search goal, its query formulation in a representation understandable for the IR system and the results generation. As the interaction with the results may change, the user’s mental model and prior domain knowledge – the search goal – may adapt as well (Bates, 1989).

This model agrees well with the definition of semantic interaction, where the user’s actions and background knowledge may change while exploring and interacting with web resources based on meaning. In other words, the concepts, goals and relevance of results for users evolve during the information seeking process as the classic understanding that the initially formulated information need is valid until the end, cannot be assumed for most cases.

4.2 Semantic Interaction Tasks in IR

In Dengel (2012, ch. 1), knowledge is described as “the ability to interpret and process data in context”, but in most cases, it is difficult to infer the semantics of structured or semi-structured data directly from its context, i.e. the web page or site. Hence, other possibilities for adding semantics to web data must be sought. In general, two types of approaches could be applied, either gathering knowledge from external knowledge, e.g. from an expert-created ontology, or from the users by applying collaborative methods such as social tagging.

The process for realizing semantic interaction for web-based retrieval systems entails the following three stages:

1. Adding semantics to data in order to turn it to information or knowledge (Semantic Layering).

2. Establishing semantic access to web sources by extracting, retrieving and aggregation relevant information depending on the target application and retrieval use case (Semantic Mediation).
3. Enabling semantic interaction between the users and the system employing visual or other semantic user interfaces (Semantic HCI).

The following sections outline the aforementioned tasks, focusing on semantic retrieval and interaction with semi-structured web pages for two distinct use case scenarios: question answering in a knowledge-based dialogue system and semantic exploration of information spaces employed in 2D/3D visual retrieval interfaces for folksonomy systems. Therefore, the focused tasks are structured according to the three regarded layers of processing: semantic layering, mediation and user interaction. From the perspective of an information seeker, the tasks related to the data and semantic mediation layers – which aim to establish semantic access to the respective data sources – remain more or less transparent, whereas the third task has a major impact for the user-machine interaction, e.g. by employing distinct semantic metaphors for interacting with visual or other semantic user interface artifacts.

4.2.1 Task 1: Semantic Layering

The following sections describe the *expert-driven* semantic layering approach via e.g. an ontology, and the *bottom-up* social or collaborative tagging process based on a social network of users.

I. Expert-based Approach

A major requirement for automatic semantic processing is to augment web-based information with explicit and formal semantics which can be obtained from the knowledge representation models described in Section 2.2. The result of such a semantic annotation process is given by semantically enriched data structures, i.e. semantic instances describing entities from a regarded domain, e.g. books from a book store, news articles or sports reports, which have been populated to the web.

The main benefit of this approach is that it allows formal processing and reasoning over concepts and relations, which describe related domains via exact and well-defined semantics. However, understanding and describing a specific domain through a controlled vocabulary or a complex ontology

entails intensive research of relevant literature and the analysis of the involved needs, actors, tasks and activities (Mai, 2006), which is a costly and time consuming procedure.

II. Collaborative Tagging

As elucidated before in Section 2.3, social semantic tagging can be utilized as a means to bridge the semantic gap in web data in form of user-provided vocabularies, enabling a folksonomy-based access to semi-structured web sources. As implicit semantics from tags represent associative semantics, they can be exploited for retrieving, sharing and exploring data and contents. Users also add contextual information when tagging preferred resources (Gupta et al., 2010). Hence, contextual information can be extracted from the social networking graph, reducing the costs for analysis in external resources or applying extensive knowledge processing methods.

In practice, a *non-expert-driven* semantic layering approach could be realized by utilizing a collaborative tagging environment for annotating structured data or informative fragments in web pages, allowing users to select and tag relevant fragments by applying a visual selection metaphor. Herewith, annotations for single data entities as well as for compound structures could be obtained by employing dedicated tagging metaphors in visual annotation user interfaces.

In collaborative or crowd-based approaches, the incentive model has high impact for guaranteeing a successful adoption of the regarded semantic layering task. It has been shown by von Ahn and Dabbish (2004), that (online) games provide a good platform for embedding certain computationally complex tasks to be resolved by humans easily while playing a human computation game.

4.2.2 Task 2: Semantic Mediation

Depending on the type and structure of information to be extracted and the target knowledge source, different models and techniques for accessing semi-structured web sources can be employed.

I. Information Extraction

The process of extracting a collection of information structures from one or more target web sources starts with the analysis of the semantically layered (sample) target structure(s).

In the case of an expert-based approach, semantic wrapping or extraction techniques as introduced in Section 3.2, can be applied to retrieve and collect the target information structures.

In the collaborative tagging approach, a dedicated access to the folksonomy hyper-graph must be provided. In general, social networking applications offer an API for accessing the data entities and associated tag structures.

For tagged web-page structures, the tagging information requires encoding either directly into the HTML document by using tag attributes, suitable marking or by referencing the tagged entities through their XPATH expressions.

As structured web page fragments can be tagged in a variety of ways, appropriate solutions for semantic tagging and retrieval of structured data have to be explored. Here, the aforementioned models and paradigms of social networking and human computation will be investigated later in Chapter 5.

Legal Issues

Although not focused on in this work, legal issues with wrapping and extracting data from web sources need to be considered when developing such systems. In general, wrapper or extraction algorithms go beyond keyword indexing, where an index generally consisting of a list of keywords associated with a target URL is created. Storing semantic structures extracted from web pages, may thus store contents from external sources invalidating intellectual property or copyright. Therefore, an automated extraction system should guarantee that only free or open data sources are accessed. In the case of social web applications, open APIs to the folksonomy and related resources must be used. When parsing web sites, the meta-tags that store the permission status for automatic processing of the individual web pages must be verified a priori.

II. Semantic Aggregation

Building different aggregation levels for retrieved data entities from an entity collection, such as a set of web documents or large sets of search results entails the creation of different abstraction levels and views on the regarded entities according to given criteria, e.g. topical grouping or hierarchical organization. Hereby, the inspection of every single entity, e.g. a document, in a returned result set can be avoided, helping to save time and effort. For

example, a user could be interested in finding out which topics are covered by a web page, or focus on detailed information entities like a person's CV, address or affiliation.

The classic approaches for aggregation are based on syntactic properties, such as keywords in web search engines or distinct aggregation operators derived from SQL, e.g. AVG, GROUP BY, etc. In contrast to grouping data entities according to syntactic criteria based on the content features, e.g. appearance in page title, frequency of occurrence, etc., *semantic aggregation* is based on semantic information resulting either from a background ontology or other forms of a knowledge network, e.g. word nets. In general, semantic aggregations can be built on top of semantic categories or grouping, clustering of classes of objects at distinct conceptual levels. Semantic structures can also be categorized into corresponding predefined groups that share certain common characteristics resulting from inheritance. In retrieval systems, the adoption of criteria for grouping, clustering and classification of search results is fundamental for combating problems with information overload. In particular, in case of broad initial queries, which tend to return tremendous amounts of data, the user is not able to screen the results quickly.

Ontological Semantic Aggregation

In expert-created ontologies, aggregation is built primarily on the conceptual basis applying *grouping* relations based on the **is-a**, **part-of** or any other semantic grouping relation. For example, semantic categories can be built by defining a generalized semantic pattern, e.g. by grouping instances that are of concept c and have properties p1 of type t1 and p2 of type t2.

The underlying metrics for grouping or categorizing information entities can be based on semantic similarity or relatedness measures (Section 2.4) for the regarded semantic structures.

Folksonomic Semantic Aggregation

In folksonomy-based systems, semantic aggregation levels of the described type can be determined, for instance, by conducting a semantic analysis of tag structures based on co-occurrence analysis of tags and applying clustering on related tags for building topic regions. Besides semantic similarities or relatedness of the tag structures, methods such as tag-frequencies, structural and taxonomic hierarchies, and other semantic relations that can be extracted from the tagged data instances, can be considered.

Furthermore, explicit semantic relations can be gathered from external knowledge sources, such as lexical-semantic word nets⁴¹, and exploited for aggregation, if it is not possible to extract them from the folksonomy itself.

4.2.3 Task 3: Semantic HCI

The conversation metaphor represents a widespread way of interacting with an information system, e.g. for the purpose of question-answering. The user simply enters one or more keywords or a natural language question and obtains relevant and preferably exact results, being able to state further related questions until a satisfactory result is achieved.

Typically, a system and dedicated components for dialogue interaction, contextual reasoning and natural language understanding are required, besides the underlying subsystems for retrieving, analyzing and indexing data from different web sources. Looking from the user's point of view re-entering keywords or questions again and again, particularly when the results are unsatisfactory, is very ineffective and thus must be avoided. Furthermore, returned answers lack transparency, if their context is not shown or explained to the users as well.

In contrast, explorative search and navigational metaphors based on semantic information, e.g. concepts or tags, can be employed as a means to increase transparency of where to search next and help to gain insight into the semantic patterns, e.g. by means of associations. As a result, a key necessity of applications for explorative search and browsing for discovering relevant information is to enable semantic interaction and navigation employing easy to use and efficient visual interaction structures (Neale and Carroll, 1997) by exploiting the semantics of the target information structures as well as adopting interaction metaphors that are understandable (Carroll, 1988) to the users.

How relevance is judged is determined by the application, either based on semantic analysis or statistics. In the case of semantic retrieval, a given natural language user query is mapped to its semantic representation in the ontology and compared with the semantic representations of the retrieved or extracted semantic information structures as possible answer candidates.

Figure 4.3 shows the relation between semantic layering and the interaction with the semantically enriched information structures for both, the ontology and the folksonomy approach.

⁴¹<http://wordnet.princeton.edu/>

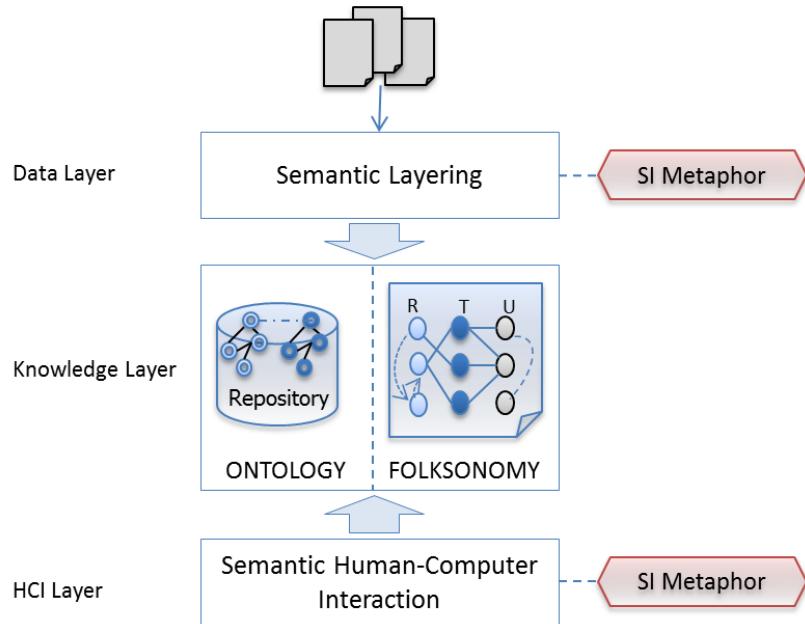


Figure 4.3: Semantic Layering and Semantic Interaction.

At the data layer (In Figure: top down), data is enriched either by users of a social network or from expert annotators for creating a semantic representation of the target information sources. While the ontology approach builds up a knowledge graph with the instantiation of conceptual instances with semantic relations, the folksonomy is represented as a tripartite structure where the major difference is the social impact of users and their relations to data, captured as a <resource, tag, user> relations.

On the HCI side, enhanced semantic interaction metaphors are required in order to exploit the semantic information from the underlying semantic data processing layers employed in dedicated (visual) retrieval interfaces.

For example, the wide-spread tag cloud interface as a reference visual retrieval interface for folksonomy systems is less interactive and has restricted semantic interaction capabilities, mostly focusing on the top most frequently used popular tags. Hence, semantic exploration of user-generated content is rarely supported.

4.3 SI Framework for Web IR

Enabling semantic interaction in a web-based retrieval system, first entails decisions about the target information structures and the method for layering and enrichment by either using an expert-created procedure or following

a bottom-up annotation method for describing the regarded domains.

In either case, the semantic structures must be retrieved/extracted from web sources, stored and prepared in a knowledge base in order to be queried or processed by the system. A semantic retrieval and interaction approach further aims at improving relevance by introducing a semantic dimension at each stage of the retrieval process.

This implicates a semantic processing pipeline of the form: semantic access (e.g. social web API) -> semantic analysis (e.g. tag co-occurrence analysis) -> semantic aggregation (e.g. similarity-based clustering) -> visualization & interaction based on semantic information (e.g. 2D semantic tag cloud based on related tags).

Before introducing a general conceptual design, the following important terms need to be explicitly defined:

- *Semantic Structures*: information structures describing facts or other types of information via a dedicated knowledge model, such as ontological concepts which express the meaning of the respective information entities.
- *Semantic Interaction metaphors*: encode a set of actions that can be performed to satisfy an information need, e.g. dive in, explore paths/branches, etc. by exploiting implicit or explicit semantics from the information space.
- *Semantic user interface element*: visual or non-visual semantic representation for the users that communicates information fast and efficiently (e.g. hierarchical semantic tag cloud, 3D visualizations, etc.).

Conceptual Design

The basic building blocks of the semantic interaction framework consist of two complementary layers, which are shown in Figure 4.4. On the right side, in the “semantic access” layer, semantic information structures, which can be obtained by resolving the tasks described under Sections 4.2.1 and 4.2.2, are stored in a semantic repository or knowledge base. It is important to hint at the two distinct functions <<`addSemantics`>> and <<`storeInstances`>> of the “semantization” process, which encapsulate the semantic layering and retrieval/extraction tasks.

The “semantic HCI” layer on the left side is responsible for the interaction part with the users. Here, a semantic interaction metaphor – imple-

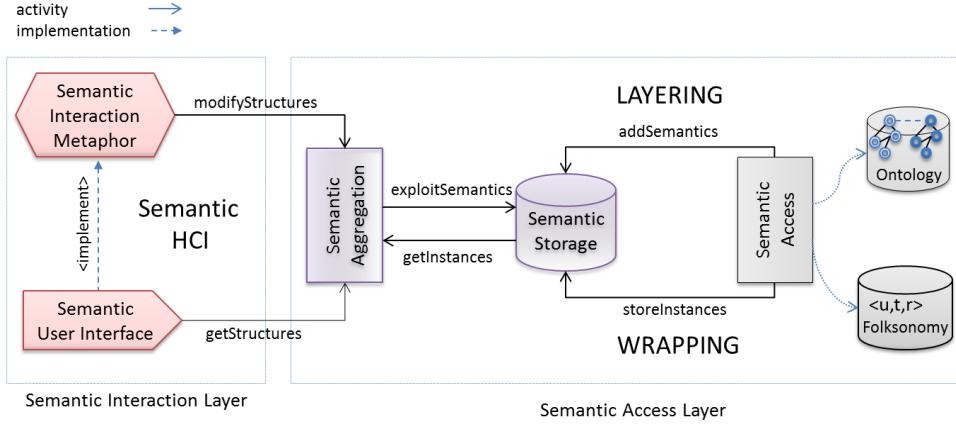


Figure 4.4: Semantic Interaction design pattern.

mented in a visual user interface – consults a semantic aggregation component, which is responsible for the mediation between the stored semantic instances and the user interface. On the one hand, it is responsible for providing a representation of the underlying semantic information structures for the semantic user interface through a `<<getStructures>>` interface, while on the other hand, it serves the interaction and navigation by allowing the modification of the respective information structures at hand through a `<<modifyStructures>>` interface.

| Case Study | Type of Access | Semantic Analysis | SI Metaphors |
|---|----------------|--|--|
| Collaborative Semantic Layering (Chapter 5) | IR/IE | Social Tagging, tag co-occurrence, word nets, HC, etc. | Selection and semantic labeling, bounding box |
| SI with Wrappers for QA (Chapter 6) | Wrapping | Semantic paths, hierarchies | Semantic queries and conversation |
| Semantic Tag Cloud (Section 7.1) | Folksonomy API | Co-occurrence relation, hierarchical clustering | 2D tag cloud, hierarchical semantic exploration, magnifier |
| Semantic 3D-Browsing (Section 7.2) and Navigation (Section 7.3) | Folksonomy API | Co-occurrence relation, star clustering | 3D semantic space, exploration in first-person shooter perspective, vehicle metaphor |

Table 4.1: Case Studies: systems, access methods and interaction metaphors.

The described framework can be set on top of an existing information retrieval model, a suitable knowledge representation, metaphors for visual-

ization, interaction and navigation, and measures for evaluating relevance depending on the use case and target application. An example implementation could therefore rely on folksonomy tags, the vector space model, 3D interaction and the vehicle metaphor to explore the information space and a semantic similarity metric as the measure of relevance. In the same way, semantic interaction could also be based on the conversation metaphor, natural language questions and information extraction.

Table 4.1 illustrates the researched case studies according to the type of semantic access, the used methods for semantic analysis and the employed semantic interaction metaphors.

4.4 SI with Wrappers in a Dialogue System

An important prerequisite for applying the conversation metaphor to web-based question answering in a knowledge-based system is to express natural language questions of users in an ontology.

Therefore, the user's question must be translated in order to map to the query language of the underlying retrieval engine. Not only must the user questions be translated to a semantic representation, but also the target information structures. This mediation task for extracting and generating ontological semantic structures from web pages can be realized by applying methods of information extraction.

Tasks:

Ontology-based SI with semi-structured web sources in a question-answering scenario can be realized through the following tasks for querying semantically enriched information structures:

1. Labeling sample instances of the target information structures employing a graphical interface and a suitable semantic tagging metaphor (Semantic Layering).
2. Generation of wrappers from the samples that are executed for extracting semantic data instances (Semantic Wrapping) and creating individual semantic instances on-the-fly or storing in a knowledge-base (Semantic Transformation).

3. Evaluating a user query utilizing a method for computing the semantic relevance of question-answer pairs of ontological instances (Semantic QA).

Figure 4.5 shows the four levels of processing for enabling semantic interaction based on ontological semantic layering and wrapping. A web page fragment showing information about a match from a soccer World Cup tournament was annotated by concepts from a football (soccer) domain ontology, created in the SmartWeb project⁴². More detailed explanations are given in the following sections under I-III.

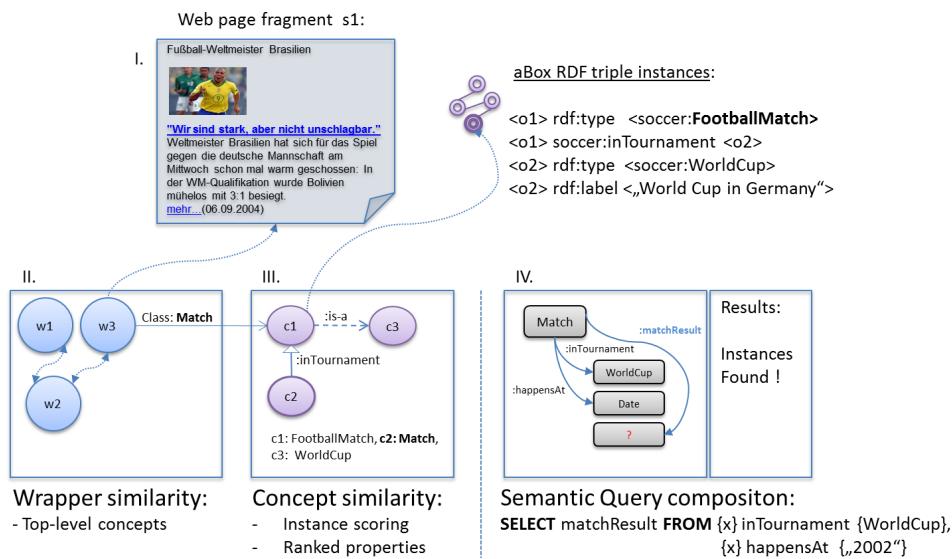


Figure 4.5: Levels of semantic interaction via ontologies.

I. Semantic Layering

In an expert-based approach, semantic information can be added to structured web data by using taxonomic structures or lightweight ontologies describing specific domains. A dedicated graphical interface can be consulted in order to assign conceptual labels and properties to individual data structures. Properties generally correspond to semantic relations between semantic instances, e.g. a street name that belongs to an address, or conceptual structures like as a personal information structure that has an address, which are often found on web pages with the CV of a particular person or

⁴²<http://www.smartweb-project.de>

a member of an institution. This annotation process results in semantically enriched data structures in form of additional meta-data.

Furthermore, as will be shown later in Chapter 5, the annotation metaphor used in such an interface plays a central role for motivating non-expert users to contribute to such annotation tasks or allow for creating complex semantic structures by selecting single entities on a web page and assigning semantic labels to them.

II. Semantic Wrapping and Transformation

Semi-structured web pages encoded in HTML are composed from diverse raw data sources using different data schemes, languages and formats. Despite this diversity data of the same type or class generally follows a predefined layout schema when generated by the same source or service, which can be exploited for extracting specific structural entities (Sections 3.2.1 and 3.2.2). Furthermore, depending on the retrieval or extraction task, how to access information - online or offline – is a crucial decision in supporting specific use cases such as question-answering.

For the expert-approach, ontology-based semantic layering can be utilized in order to add meaning to specific data structures in web documents. The main challenge here is to identify the parts that are relevant by means of discriminating patterns due to structural similarities to non-relevant parts, e.g. menus, advertising, etc. This task is not trivial, while human experts tend to identify significant parts looking at a web page easily.

The identified sample structures or HTML fragments (Figure 4.5, I.) can then be used to generate wrappers (Figure 4.5, II.) for extracting dedicated semantic structures (Figure 4.5, III.) for certain top level concepts (class Match) and populating entity collections of RDF triples to a knowledge repository. The semantic level is provided by the ontology used, here, the SmartWeb sports-event ontology (Oberle et al., 2007) providing a detailed description of the football (soccer) domain, as will be shown in the use case study in Chapter 6. Once a rich semantic knowledge base is created by considering online sources, the semantic data structures and associated semantic information obtained can be analyzed by employing methods for semantic analysis, aggregation and classification as presented in Chapter 3. For example, similarities between ontological instances can be analyzed in order to extract candidates that comply with a given semantic query (Figure 4.5, IV.). In the example that is shown, a semantic query is represented in an RDF query language for obtaining the result for a particular football

match at the World Cup in 2002.

Semantic Access at Query Time:

Furthermore, the architectural foundations of the Web, based on the stateless HTTP protocol for accessing resources and the client-server architecture allow two fundamental access patterns for realizing data access: *offline crawling* (Brin and Page, 1998) and *on-the-fly analysis*, besides hybrid forms such as federated querying (Kossmann, 2000).

Both of these fundamental architectural patterns impose a trade-off between data currency and degree of data completeness vs. the speed of query execution (Heath and Bizer, 2011). Bizer et al. (2009) list the number of data sources covered, the degree of freshness⁴³, the response time for queries and runtime discovery of new data sources as the basic factors that influence online access to the web. Therefore, establishing semantics access to web sources at query time entails specific requirements for obeying the aforementioned restrictions, which need to be considered when implementing such types of retrieval systems.

Applying the first (crawling) variant allows semi-structured data to be gathered from diverse web sources in order to populate large collection of semantically indexed web data - similar to the web search indexing approach. In general, via this pattern no permanent online access to the original web sources is intended. Recalling the freshness problem previously discussed, this solution is therefore not feasible in use cases, where information currency is crucial, e.g. in price comparison of products, stock rates, emergency management, etc.

As similar results and additional aspects can be extracted from several web sources, selecting the most fitting answer necessitates to calculate the relatedness and similarity between the semantic user query and several answer candidates more deeply, considering their semantic content and required semantic relations.

III. Semantic Querying

Scoring and ranking ontological instances according to a given query is a major requirement of semantic query processing and answer selection. The aim is to choose the most likely answer for the given semantic user query within a set of extracted candidate instances.

⁴³This term was used by Lewandowski et al. (2006) and could also be described as the up-to-dateness of the search index.

Approaches of instance scoring are generally based on calculating the semantic coherence (Gurevych et al., 2003), relatedness or similarity between the query and answer candidate representations.

4.5 SI in 2D/3D Visual Retrieval Interfaces

Emergent semantics that can be extracted from user-generated vocabularies (folksonomies) have several advantages compared to expert-created ontologies. Firstly, they contain user injected semantic associations, which can be exploited by retrieval or recommendation systems. Secondly, they allow for leveraging collective intelligence uncovering missing or inherent semantic relations that are implicitly defined by users. Thirdly, they allow for reducing cognitive workload for the users in semantic interaction, compared to interactions based on expert-created knowledge models.

Tasks:

On the social web, tagging of web resources in folksonomy systems, such as social bookmarks, images or blogs, etc. is well established, whereas social tagging of specific structured data inside web documents was rarely explored.

I. Social Semantic Tagging of Structured Data

The main constituents of a social tagging system for labeling structured data entities in a web page comprise:

- a collaborative tagging interface for marking structured data, which employs the social networking idea based on the previously described tripartite model of collaborative tagging.
- a visual annotation metaphor for promoting conceptual tagging, i.e. conceptualization of semantic structures.
- post-processing and consolidation of the obtained tag structures.

Figure 4.6 illustrates the exemplarily collaborative tagging process for structured data fragments in web documents, where a_i , b_i denote entities describing atomic facts, while c_i represent tags of conceptual nature.

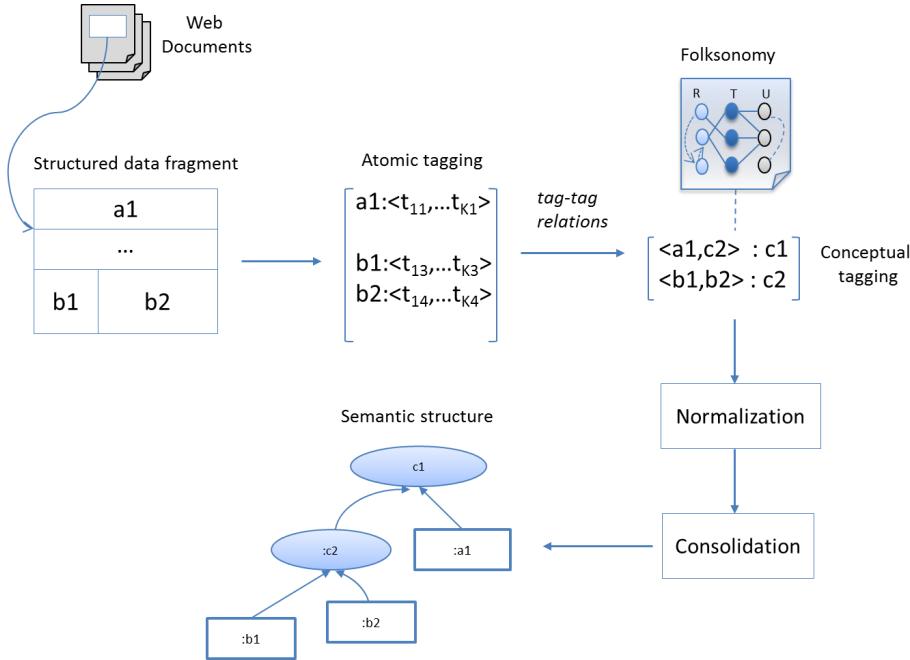


Figure 4.6: Social semantic tagging of structured data.

Structured data can be tagged applying appropriate selection metaphors for either marking single (atomic) data entities or building complex annotations applying dedicated selection metaphors for grouping or linking (e.g. line-drawing, using a bounding box or circle) of the individual information structures.

In the case of assuming the application of the widely used free tagging, flat folksonomies arise as a result of the agglomeration of user provided tags. Tag structures created this way can be analyzed via the tripartite model of collaborative tagging that was introduced in Section 2.3.1 or by consulting external knowledge sources, such as Wikipedia or WordNet in order to extract relevant semantics for the tagged resources.

Finally, the goal of the post-processing and consolidating step is to rank and add the correct meanings to tags and calculate their significance for the associated web fragments.

II. Folksonomy-based Semantic Access to Web Data

In existing folksonomy systems, a dedicated API for accessing the folksonomy hyper-graph is provided. For example, the delicious social bookmarking service allows access to the stored user-generated bookmarks and associated user tags from the underlying hyper-graph via a social application

interface⁴⁴ (API).

For structured data tagged via the social tagging approach described above, a similar access to the tag structures and associated structured data is required. In principle, semantic tagging of (structured) data fragments on a web page can be integrated with data extraction or wrapping techniques (Section 3.2), allowing for extracting instances of the labeled relevant or interesting data structures automatically. Therefore, a set of extraction or wrapping rules on textual or semi-structured data sources need to be induced.

For example, a web information source of a specific newspaper encodes structured data following the same or similar schema, independent of the domain to be described. Hence, articles describing political news, sports or finance can be extracted via the same extraction rules or wrappers if the schema fits all three sub-domains. What would be different is the mapping to associated semantic concepts or tags within the semantic transformation process.

Compared to the ontology-based approach, the resulting semantic structures can't be referred to as formal and explicit in the first sense. Therefore, different application scenarios for exploiting associative semantics in social networking applications must be investigated. It is clear, that the most important challenge is user motivation for tagging. Hence, inviting incentives for the semantic tagging task at hand is crucial. An automatically generated mashup⁴⁵ could be one possible incentive, allowing access to personalized contents from diverse web sources. The semantic tagging task could also be addressed via human computation online games, as will be discussed in Section 5.2.

III. Visual Exploration based on Associative Semantics

For web users, semantic access based on folksonomies has been shown to provide additional benefits for supporting the exploration of collections of knowledge entities on the web by promoting serendipitous discoveries (Mathes, 2004). Furthermore, Fu et al. (2010) showed that exploratory search performance depends critically on the match between internal knowledge (domain expertise) and external knowledge structures, which is inherently supported by the bottom-up semantic layering approach in social or collaborative tagging. Hence, it can be assumed that the similarity between

⁴⁴<https://delicious.com/developers>

⁴⁵[http://en.wikipedia.org/wiki/Mashup_\(web_application_hybrid\)](http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid))

the internal and external knowledge representation, which result from the bottom up creation of semantic meta-data by users in a social tagging process, has positive impact for improving knowledge processing and retrieval. Figure 4.7 illustrates the idea of associations in a network model of our memory.

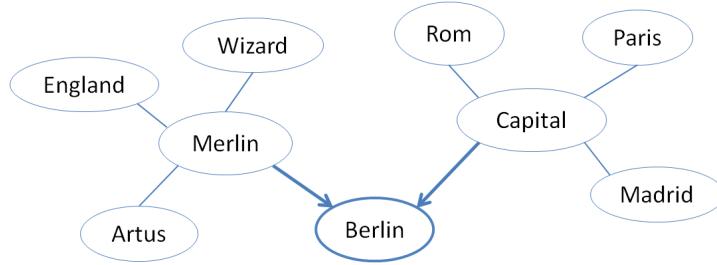


Figure 4.7: Activated nodes of associations in a network (Reinsberg, 1997).

In the example above, the activation of the target node (“Berlin”) is precisely triggered by consulting two associated clues “rhymes with Merlin” and “capital in Europe”. Consequently, additional associations serve the purpose of disambiguation.

This evidence is further supported by cognitive and psychological analysis of social tagging, showing that related tags reflect immediate associations between concepts in our collective knowledge model; a form of a collective group memory (Held and Cress, 2008).

In folksonomies, the notion of related tags represents semantic relatedness of individual terms, which can be determined through semantic analysis (see Section 3.3.3) applying e.g. the statistical co-occurrence method. In this way, a more informative way of representing the semantic associations of the underlying information space is possible. For example, the meaning of unknown tags or keywords can be deduced through tags that are located in the “spatial” or “semantic” neighborhood. Here, spatial proximity is mapped to semantic relatedness. Spatial proximity can be exploited applying different spatial metaphors and interaction modalities, as will be investigated in the case studies presented in Chapter 7. As will be shown later, integrating folksonomy-driven exploration into visual retrieval interfaces employing dedicated interaction and navigation metaphors has the potential to enhance user experience in exploring collections of web resources.

4.6 Summary and Roadmap

The status quo in web search is represented by relevancy ranked results, where semantic information is rarely, or not considered at all. With the increasing availability of semantic technologies and standards new ways to explore, organize and present web information become feasible. Apart from the relevance of results based on statistical term occurrences, semantic relatedness of the discovered information and efficient and intuitive user interaction are major challenges for next generation semantic retrieval interfaces. Hence, in order to improve user experience in searching and exploring web information, introducing a semantic dimension to information seeking and human computer interaction in visual user interfaces of retrieval is essential.

Besides that, a shift from representing documents solely by their textual representation towards a semantic representation takes place, where semantic objects representing the contents of web documents in form of ontological instances are provided as graph-based semantic structures. Furthermore, tag-based representations that originate from collective user contributions can be increasingly exploited, analyzing the semantics of the user-provided flat tag lists.

Studies on “search engine user behavior” conducted by iProspect⁴⁶ show, that more than 80% of users will try new forms of search if they are not satisfied with the results they find within the first 3 pages of results returned by the system. Another consequence of the study is that users increasingly appreciate the quality of data aggregation and (visual) interaction.

As a result, semantic information can help to explain the relation and relevance of documents and queries based on the meaning of their contents. This type of retrieval is able to support high precision and recall types of search by allowing the exploration of certain types of documents having particular information structures related to a given query or information need. In contrast, common IR systems focus statistical relevance of documents and queries, which ignited controversial discussions about the notion of relevance in information systems (Heath, 2008, Marchionini and White, 2007, Toms et al., 2005).

Roadmap:

The following Chapter 5 investigates alternatives for semantic annotation of structured web data and informative web page fragments based on the

⁴⁶<http://www.iprospect.de>

Wisdom of the Crowds principle.

The case studies described in the last part in Chapters 6 and 7 aim to investigate and implement various aspects of the semantic interaction approach and dedicated tasks above introduced for both use cases presented in the Sections 4.4 and 4.5.

Part II

Case Studies and Applications

Chapter 5

Collaborative Semantic Layering

Although researchers proposed different ontology-based forms of annotation for web content in the last few years they point out that the process of creating annotations is expensive, difficult to maintain and error-prone, while having still the barrier of sparse and low quality meta-data for the semantic web beyond single datasets created by small communities of researchers or semantic web enthusiasts. Nevertheless, recent analyses of the usage of micro-formats and RDFa (Figure 5.1) show that RDFa usage has increased by 510% between March 2009 and October 2010, from 0.6% to 3.6% of web pages of the 12 billion that were analyzed worldwide. For the structured part of the web in particular, these developments are very encouraging.

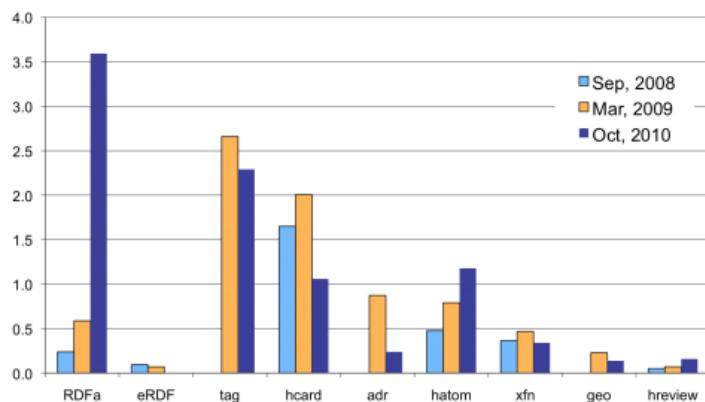


Figure 5.1: Microformats and RDFa deployment across the Web⁴⁷

While micro-formats or RDFa help to reduce the encoding effort for

⁴⁷<http://tripletalk.wordpress.com/2011/01/25/rdfa-deployment-across-the-web/>

users, providing a simplified knowledge representation (Allsop, 2007, McCool, 2006), manually created meta-data still suffers from a lack of quality as most users tend to make mistakes when additionally confronted with the ambiguity and complexity of natural language.

The trend to exploit collaborative forms of tagging together with formal and explicit semantics encoded in standardized ontologies can be seen as a compromise to balance the aforementioned trade-off between semantic meta-data quality and the annotation costs and efforts.

In social web applications, simple entities such as links (bookmarks), images, videos, blogs, etc. are mainly tagged by users. Tagging is generally performed on document level or at the level of single objects or articles. The question investigated in the following sections is how concepts such as social tagging or the human computation paradigm can be applied for the task of semantic annotation of information structures or informative sections in web pages.

The focus of the research is on associative semantics in the form of user tags that are assigned to the target information structures in a collaborative tagging process. Such user-contributed semantics in the form of tags can be collected, analyzed and consolidated in order to assign the most representative semantic descriptions for a resource, e.g. to be exploited by information retrieval systems. Besides descriptive tags, users may also assign contextual information or related questions to information structures on a web page in order to be exploited, for example, for information extraction or question answering.

In the following, firstly, a study related to collaborative semantic tagging dedicated to the annotation of information structures in semi-structured web pages is presented in Section 5.1. Then, alternative approaches for tagging web contents utilizing the Human Computation (HC) paradigm employed in online games are investigated in Section 5.2.

5.1 Semantic Layering of Structured Data

Existing approaches for collaborative annotation on a large (web) scale apply social tagging for annotating either textual data or single elements of a web page, such as images, videos, links and other types of web page elements. Hence, social tagging on a document level is widely supported, while annotating structured information in web pages via social web applications has not yet been addressed.

In order to be able to apply semantic tagging to structured web data,

an appropriate incentive model must first be provided to the users. A social web application must then employ a suitable selection and tagging metaphor in the visual user interface. The main challenge here is how to design engaging user interfaces with a clear incentive model and employ the right interaction metaphors for motivating users to participate in providing rich and high quality meta-data seamlessly.

Therefore, the aim of the studies in this chapter is to investigate means for semantic annotation of information structures inside semi-structured web pages in order to be exploited, for example, in retrieval or question answering systems. As a first step towards this, a collaborative tagging environment and appropriate visual annotation metaphors for creating annotations dedicated to information structures in web pages are explored. Thereafter, the user-created tag sets are analyzed in order to extract implicit semantics, e.g. conceptual tags, predicates, etc. which could be used to describe important aspects of a regarded resource.

Before describing related work and the workflow for collaborative tagging of structured data in web pages, some important definitions are first given:

- *Semantic Structure* describes an information entity with mark-up that accurately reflects and allows the expression and extension of the meaning of the content.
- *Semantic Layering* refers to assigning implicit or explicit semantics to regarded content, not only tags.
- *Conceptualization* refers to the means to form a concept of an entity or thing of the real world. In this definition, conceptualization refers to concept building in the senses discussed in Section 2.2.2, i.e. an ontology as shared conceptualization of a domain.
- *Consolidation* refers to consolidation of data from multiple sources or users into one central representation. For tags, this means that the meaning of a consolidated tag should be as close as possible to the corresponding concept from an expert ontology or expert tag, e.g. from a gold standard or a synonymous entity describing the same fact, event or structure.

5.1.1 State of the Art

The *sticky note* metaphor described by Pascoe (1997) was used in many location-based applications to attach meta information to things or objects

in the real world such as buildings, locations, streets, etc. that were visualized on a map. The Diigo⁴⁸ social bookmarking application uses such a metaphor for creating and sharing of social annotations. Registered users can highlight and comment or attach sticky notes to arbitrary parts of a web page, as shown in Figure 5.2.

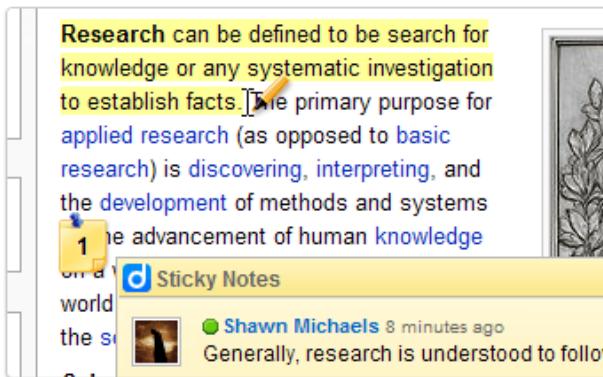


Figure 5.2: Highlighting and sticky notes using the Diigo Tool.

Compared to the supervised annotation systems described in Section 3.3, that utilize a graphical user interface for marking web data using concepts and relations from a controlled vocabulary, e.g. a lightweight (domain) ontology, social semantic tagging approaches (Section 3.3.2) rely on semantics from the users of a community, i.e. a social network.

Most of the annotation systems known from NLP applications are based on marking parts of natural language text, which is exploited in information extraction systems for textual sources. Semi-supervised Wrapper generation systems for web-data (Section 3.2) mostly make use of a graphical user interfaces for marking and annotating “sample” web data for the purpose of structured information extraction, i.e. they exploit structural characteristics of the data.

Figure 5.3 shows a simple semantic structure expressed in the N-triple notation. In the example, a well-defined concept from an ontology (Book) and appropriate attributes (title, price, author) have been assigned to describe the respective structured data object. Thinking of users who contribute to the semantic annotation of such structures, the respective general concepts and their attributes could be found by consulting a controlled vocabulary such as WordNet or specific domain ontologies.

⁴⁸<http://www.diigo.com>

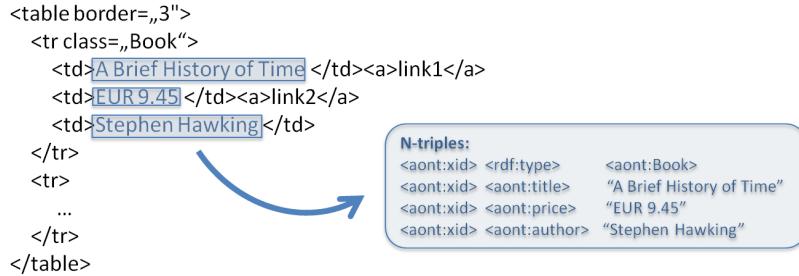


Figure 5.3: Annotation of a semantic structure on a web page as N-triples.

Background: Tag Analysis

As discussed in Section 2.3.1, tag structures tend to stabilize over time with increasing user participation allowing the identification of representative related tags and conceptual semantic structures more precisely, which is a prerequisite for extracting emergent semantics from folksonomies. Consequently, speaking of “promoting conceptualization” of semantic structures is a twofold process: identifying representative tags for the individual data items (properties) and detecting conceptual tags for the compound structure.

Croft and Cruse (2004) distinguish between three cognitive levels of tags: superordinate, basic and subordinate tags.

In the work of Golder and Huberman (2005), the focus lies on basic level tags which are shown to have a greater probability for agreement of terms than superordinate and subordinate level tags. Besides this, basic level tags have the least cognitive cost for the user, i.e. they are thought of more quickly (Croft and Cruse, 2004), thus, are more likely to have a high frequency resulting from high agreement among the users that tagged the respective resource.

5.1.2 tag2Wrap - Workflow

In the following, a collaborative semantic tagging environment (Aras et al., 2009) for annotating semantic structures in web pages is described. The main objective of the workflow shown in Figure 5.4 is to support the creation of emergent semantics from user-provided tags which could be exploited by web information extraction or retrieval systems. In contrast to the Diigo approach, which is a general purpose social annotation environment, rather semi-structured web page fragments are tagged by applying appropriate selection metaphors for promoting the conceptualization of the user-provided

tag structures.

1. Collaborative Tagging of Semantic Structures

Having in mind the uncontrolled nature of vocabularies that are formed through collaborative tagging, the following two tasks need to be resolved in order to form “reusable” semantic structures from the individual flat tag list that have been attached to web page elements:

1. Tagging of semi-structured data records in a web page applying a selection and annotation metaphor for building semantic structures.
2. Extracting semantic structures from the user-provided tag sets applying appropriate methods of analysis (statistic and lexical-semantic analysis).

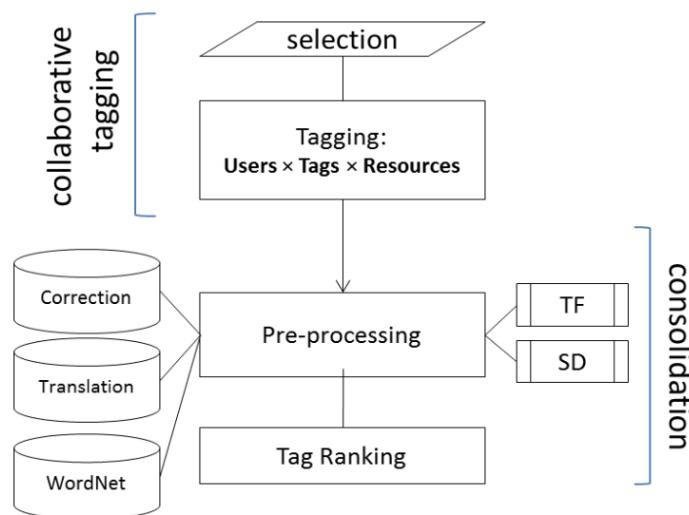


Figure 5.4: Workflow for creating emergent semantic structures from tags, depending on the Term Frequency (TF) and Semantic Depth (SD) metric, see Def. in Section 6.2.3

While technically, a web page can be (pre-)processed on the basis of its DOM tree, visual characteristics (Cai et al., 2003) have a direct impact on the ability to recognize and select a coherent semantic structure (Xiang and Shi, 2006).

Hence, hierarchical selection and tagging requires the use of two basic selection principles, one for tagging atomic/low-level elements of a structured data object, such as the author information of a book (of the basic

data type string) and another, for grouping selected elements. In practice, this corresponds with finding appropriate sub-trees in DOM using a visual selection metaphor, e.g. bounding box.

| data | interaction | structural information |
|------------|-------------|--|
| structural | single | URL, XPATH, node position |
| | multiple | URL, XPATH of the parent node, visual cues |
| textual | single | URL, XPATH to leaf, text offset |
| | multiple | stores multiple single textual selections |

Table 5.1: Selection types, interactions and structural information.

The goal of subsequent atomic selection and grouping steps is to create, for example, a taxonomy hierarchy, which is modeled via *is-a* relations in lightweight ontologies. Semantic relations of this form can again be mapped easily to *subject-property-object* triples. Table 5.1 shows the distinct interactions and associated structural information which is stored for the tagged data to be used in pre- and post-processing steps.

Applying the tripartite model that was described in Section 2.3.1, a resulting annotation instance consists of a triple data structure formed by the annotations gathered from N users that used K tags for annotating the regarded data items and the entire conceptual data structure. A list of free tags is attached to each item of the example structure from Figure 5.3, resulting in 3 or 4 (if the entire concept is tagged as well) tag lists. Formalizing this annotations scheme, the following representation can be used:

- For each atomic entity a_i annotations of the form $a_i = \{t_1, \dots, t_K\}$ are stored, with t_i being tags from an uncontrolled vocabulary, and $t_i \neq t_j$ for $i, j \in \{1, \dots, K\}$ and $i \neq j$.
- for each conceptual entity c_j a *property-of* relation of the form $\textit{property-of} : c_j \rightarrow \{a_i\}$ is stored.

2. Consolidation of Tag Lists

As folksonomies have semantic problems that stem from the uncontrolled nature of vocabularies in free tagging, several methods for pre-processing, e.g. usage of WikiPedia or WordNet for resolving issues with synonyms, polysemy, etc. have to be applied. Consequently, tag consolidation (see Section 5.1.4) in the proposed approach entails pre-processing steps such

as tag correction, translation, calculating the semantic depth of the corresponding synsets in WordNet and the tag frequency. As several synonyms may exist for a given tag, a ranking method is applied to select the most promising tags. In summary, the building blocks of the semantic tagging framework comprise a user interface for enabling users to select and tag semi-structured data elements on a web page, a selection metaphor for promoting/assisting conceptualization of tag structures, and a consolidation step (post-processing) for extracting a representative semantic structure from the user assigned annotations by analyzing semantic-lexical relations and tag frequencies.

The following sections describe the data model and used pre-processing steps, the user interface and the details of the consolidation method for the user-generated tag structures.

5.1.3 Design and Tagging User Interface

In the following, the data model and the user interface of the collaborative tagging environment is presented.

Data Model

The applied data model is based on the tripartite model of collaborative tagging, comprising users, tags and resources for building social annotations for arbitrary web pages. The basic characteristics of such annotations for individual semi-structured data objects can be listed as follows:

- annotations are stored using the XPATH to each atomic element
- grouping atomic entities serves the purpose of conceptualization. Here, the XPATH of the parent node (common ancestor) of the selected atomic entities that are leaves or sub-trees in DOM are stored.
- a unique identifier is used for each annotation.

The resulting annotations are stored in the Annotea-RDF format utilizing the Jena Semantic Web Framework⁴⁹. For identifying the particular annotations, the XPointer framework of the W3C is used. Here, the XPATH to the particular data item in a web page is utilized in the context of the annotation. In the example shown in Figure 5.5, an annotation of a “book” data item on an example book web page is stored via its URI, the :annotates

⁴⁹<http://jena.sourceforge.net/>

and the :context RDF properties from the basic annotation namespace⁵⁰ for identifying the particular tagged entity.

```

<RDF:Description RDF:about="urn:annot10210932">
  <NS1:body RDF:resource="urn:body10210932"/>
  <NS1:annotates RDF:resource="http://www.tzi.de/tag2wrap/book/" />
  <NS3:language>de</NS3:language>
  <NS1:context>http://www.informatik.uni-bremen.de/tag2wrap/book/# 
    xpointer(string-range(/html/body/table/tbody/tr[2]/td/div/div[1]/table/
      tbody/tr[3]/td[3]/div[2]/table/tbody/tr/td[2], "", 1, 2))
  </NS1:context>
  <NS3:date>2008-09-5T15:42:29+0200</NS3:date>
  <NS1:created>2008-09-5T15:42:29+0200</NS1:created>
  <NS3:creator>consolidate</NS3:creator>
  <RDF:type RDF:resource="http://www.w3.org/2000/10/annotationType#Comment"/>
</RDF:Description>
```

Figure 5.5: tag2wrap Annotations in RDF via XPATH.

User Interface and Selection Metaphor

The user interface metaphor is based on the idea of reading and commenting on a paper, document, etc. and attaching sticky-notes, writing comments, etc. The user is able to mark interesting parts and add keywords to describe the selected information. While the paper version allows for adding notes at any position, tagging web page elements works on the underlying tree structure and embedded contents.

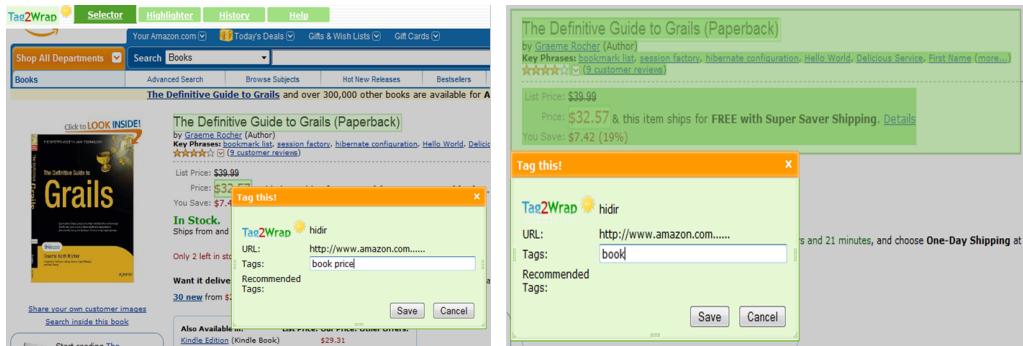


Figure 5.6: tag2wrap UI - single (left) and box selection (right).

Two simple interactions are sufficient for selecting single individual data items and grouping single data items to form more complex structures.

Figure (5.6, left) shows a website that contains a list of books with

⁵⁰<http://www.w3.org/2000/10/annotation-ns#>

detailed information such as title, author, etc. Using structural selection, users are able to select, e.g. the title of a book for tagging. When releasing the mouse button after the selection is made, a dialog box pops up where the user can enter one or several tags for this data item. The user can proceed with the other data items in order to tag parts or the entire semantic structure, i.e. a book. Single selected entities can be grouped as a structured entity using the bounding box selection (Figure 5.6, right). Again, one or several tags can be added in order to describe the selected block that contains the individual associated data items. The latter calculates the parent node for all contained single selections in order to determine a unique XPATH for the grouped elements.

5.1.4 Consolidation of Tag Structures

Figure 5.7 illustrates the idea of tag consolidation for a resource. In the shown example, a user has tagged a football match “Italy against Germany” by first marking the two data items “Italy” and “Germany”, followed by grouping the data items using the bounding box selection. As a result, 3 tag list (left) are assigned to the regarded data items. The number in front of the tag lists corresponds to the node index in DOM for identifying and collecting the obtained annotations from different users. For simplified processing, the annotations are stored separately utilizing the unique XPATH to each tagged entity, i.e. a content element or an inner node.

The goal of the consolidation process is to add the correct meaning to tags and calculate their significance in order to unveil a semantic structure from the user-provided tag lists for the regarded structure of semi-structured information fragments. Hence, characteristics of tag lists, such as misspelling, different word forms, repeated tags, synonyms, etc. have to be resolved. A lexical-semantic word net, such as the English WordNet, can be utilized to analyze semantic relations between words, such as hypernym-hyponym relations, synonyms, meronyms, etc. In this work, the following procedure is applied to obtain a semantic structure:

1. Normalization of the given tag structures, e.g. spelling correction, stemming, etc.
2. Calculating tag frequencies where repetitions are removed at the same time
3. Using lexical-semantic analysis for determining the appropriate meanings

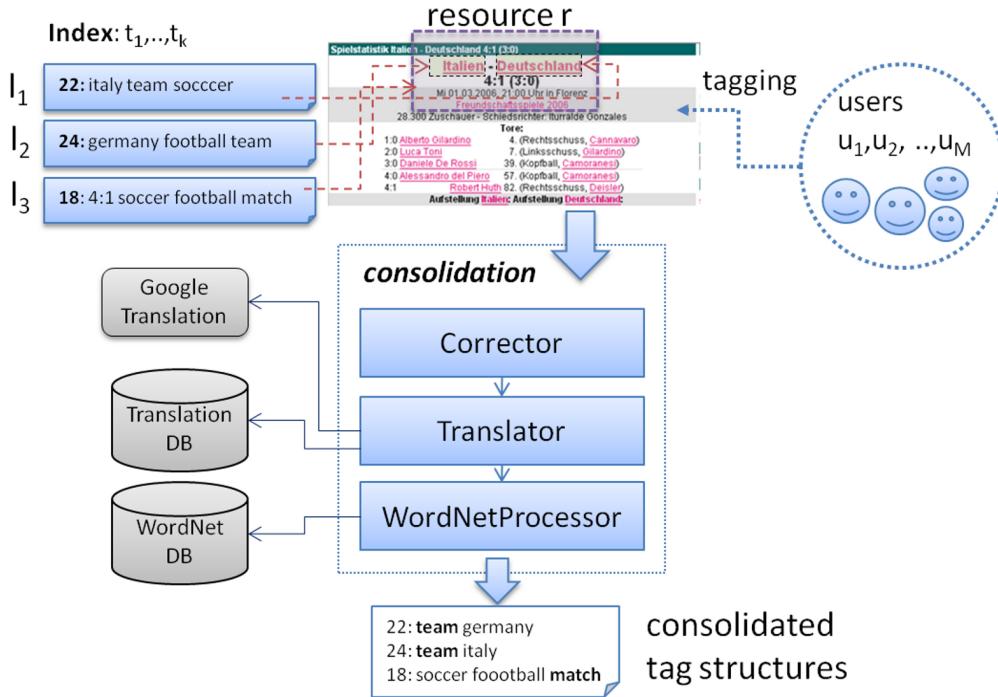


Figure 5.7: Consolidation of tag structures.

Appropriate sense keys from the WordNet used are assigned by consulting the WordNet API and identifying synonyms for tags. In case of multiple possible sense keys, the one with the highest probability of occurrence is chosen applying an enhanced method. Synonyms, then, can be identified by following up the semantic relations of the regarded concepts.

Calculating Tag Significance

Tag frequencies tend to stabilize over time for the most popular tags for a resource, which has been shown by Halpin et al. (2007).

For that reason, tag significance is first calculated by using the following term frequency (TF) measure over similar or synonym tags t in a tag list l , where $h_l(t)$ is the frequency of the tag t in the list l and a_l the total number of tags in the list:

$$\text{TF } (t,l) = \frac{h_l(t)}{a_l}$$

From the hypernym relation of the assigned words, it is possible to detect word hierarchies accordingly. Nouns in particular tend to lead to long chains of word hierarchies. The semantic depth (SD) is evaluated using

the following recursive formula for counting the number of hypernyms for a tag up to its root concept in WordNet:

$$SD(t) = \begin{cases} SD(hypernym(t)) + 1, & \text{if hypernym of } t \text{ exists} \\ 0, & \text{else} \end{cases}$$

Finally, the rank of the previously described scores is computed by multiplying term frequency (TF) and semantic depth (SD) for a tag t in a list l accordingly. In the case of several existing hypernyms for a term, the one with the highest occurrence probability (available in WordNet) is chosen.

$$\text{Rank}(t,l) = \text{TF}(t,l) \cdot SD(t)$$

The rank of a tag in its tag-list is used to consolidate and select the tag with the highest rank. The described procedure is applied to all marked structural content elements on a web page. It should be considered that synsets with high occurrence probability can again comprise several words that eventually need to be disambiguated somehow. Another possibility would be to store all the most likely words for ranking and describing a resource. For the purpose of this work, however, this fact is not taken into account.

| | sports | weather | cv | book | movie | Σ |
|----------|--------|---------|--------|--------|--------|----------|
| Novice | 166 | 30 | 89 | 33 | 62 | 380 |
| Expert | 7 | 2 | 12 | 8 | 11 | 40 |
| Σ | 4.22% | 6.57% | 13.48% | 24.24% | 17.74% | 10.53% |

Table 5.2: Number of marked resources in the test web pages.

5.1.5 Evaluation and Results

The described consolidation method was implemented as a prototype and evaluated in a web experiment by letting users create a collection of tags for given web pages from several domains. Students, research associates from the computer science faculty, and external test persons were invited to participate. Tagging was introduced using a website with a step by step explanation of the tagging tool, a screencast and an online questionnaire form that users had to fill out after tagging was completed. The users were divided into two groups: experts and novices depending on their background knowledge about social networking and tagging, etc.

Users had to tag several predefined web pages from the domains sports, weather, cvs, books and movies in German. 20 persons ranging in age from 20 to 38 participated in the tagging experiment and created 1165 tags that were distributed over the web pages from the five predefined domains. After pre-processing, 1088 usable tags remained for analysis.

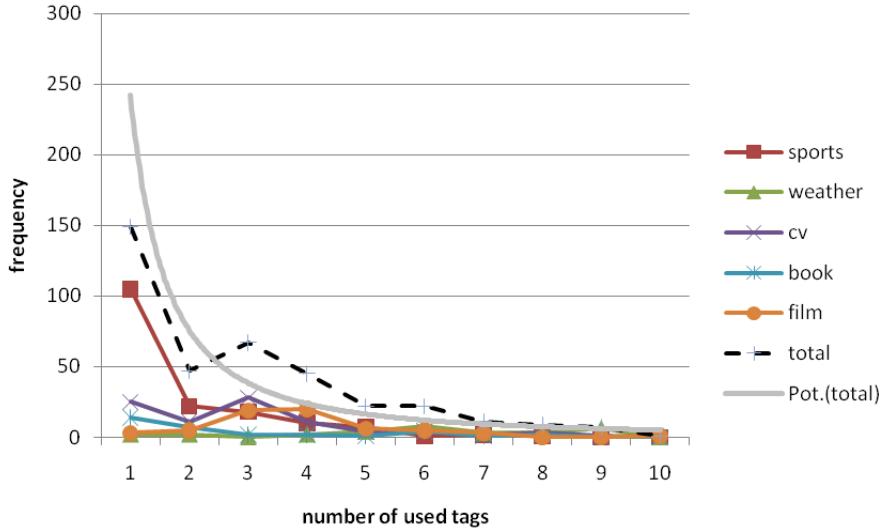


Figure 5.8: Tag distribution in the web experiment (approx. power law) for all user assignments.

Looking first at the number of tags the users assigned to the target web pages (Table 5.2), one result was that experts used far less tags for describing a resource than the novice users. The distribution of the number of assigned tags for the regarded five domains which is visualized in Figure 5.8, approximates the aforementioned typical power law distribution. The curve shows that, for the sports domain for example, most of the resources have been described using one tag (105 from 166 = 63%). Most of the

| | sports | weather | cv | book | movie | Σ |
|----------|--------|---------|----|------|-------|----------|
| type 1 | 7 | 5 | 1 | 3 | 10 | 26 |
| type 2 | 7 | 1 | 15 | 7 | 8 | 38 |
| type 3 | 1 | 0 | 4 | 6 | 8 | 19 |
| other | 0 | 1 | 0 | 0 | 1 | 2 |
| Σ | 15 | 7 | 20 | 16 | 27 | 85 |

Table 5.3: Analysis of web page structure in the five domains (Appendix A.1).

participants used a mere four tags or less to describe a resource. It could also be observed that the structure of a web page had influence on the tagging behavior. Comparing the structured pages (of type 1 and 2) vs. the rather unstructured (type 3) a tendency towards using less tags for more structured resources can be observed (see Table 5.3).

| User Tags | Synset | Description | TF | SD | Rank |
|--------------|------------|----------------------------|------|----|------|
| Darsteller | cast | Actors in play | 0,5 | 6 | 3 |
| Schauspieler | actor | Theatrical performer | 0.17 | 10 | 1,67 |
| James Bond | james bond | secret operative 007 | 0,17 | 10 | 1,67 |
| Bond | bond | elect. force linking atoms | 0.17 | 9 | 1.5 |

Table 5.4: Assigned tags for “James Bond” resource. Chosen tag = “Darsteller”

After pre-processing such as removing spelling errors, etc. the consolidation method was applied in order to obtain the most fitting concept from the set of user assigned tags for a resource. The synsets in Table 5.4 represent selected terms that were found in WordNet for the user tags “Darsteller”, “Schauspieler”, “James Bond” and “Bond”. The example shows the evaluation of a marked sample. Here, the tag “Darsteller” (cast) with the highest rank was selected as the result concept of the consolidation process.

In general, the results showed that synonymous tags are disambiguated well with this method, while there are also cases with falsely assigned tags. The reason therefore lies in the chosen method for selecting hypernyms with the highest occurrence probability obtained from WordNet, which does not provide optimal results for all cases. Hence, the occurrence probability can't be seen as a robust selection criterion and must be replaced appropriately employing enhanced metrics. For example, embedding this disambiguation task into a human computation application or game could resolve this issue by making use of human skills.

Table 5.5 gives an overview of assigned tags, selected synsets from the word net and the number of wrongly assigned synsets/concepts and the percentage of affected tags. For example, for the sports domain, 49 synsets were selected by the consolidation method, while 4 synsets were wrongly assigned.

In overall 306 tags were provided by the users for the sports web pages. The number of tags related to these tags were 41 (13.4%). Looking at the general picture for all assigned tags, false assignments of tags to synsets

| | sports | weather | person | book | movie | Σ |
|----------------|--------|---------|--------|-------|--------|----------|
| total synsets | 49 | 34 | 48 | 34 | 48 | 188 |
| false assigned | 4 | 0 | 5 | 2 | 4 | 15 |
| % | 8.16% | 0% | 10.42% | 5.88% | 8.33% | 7.98% |
| total tags | 306 | 181 | 266 | 94 | 214 | 1088 |
| false tags | 41 | 0 | 33 | 4 | 40 | 116 |
| % | 13.40 | 0% | 12.41% | 4.26% | 16.60% | 10.66% |

Table 5.5: Analysis of Tags and false assigned WordNet synsets.

vary between 0% to 16.6%.

Usability evaluation of the tagging tool

A brief usability evaluation was conducted in order to assess the usability of the social tagging tool “tag2wrap” concerning the parameters convenience, intuitiveness, clarity, visual favor, grade of auxiliary, etc. within the described web-based experiment. In addition, the users had to answer 9 questions from a questionnaire form (Appendix A.1).

Evaluation Results

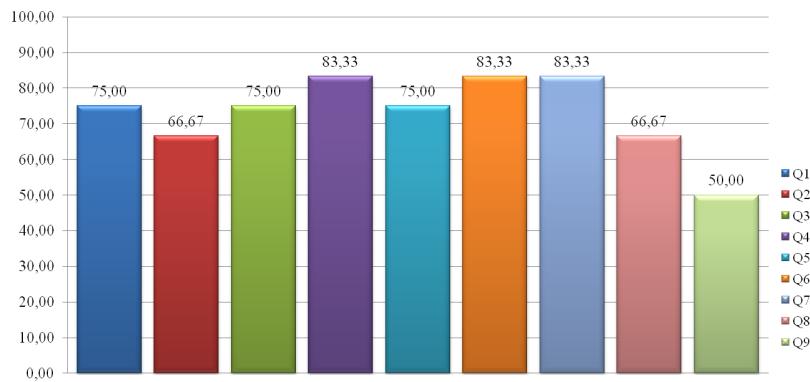


Figure 5.9: Results of the Usability Evaluation.

In Figure 5.9 the SUS (Brooke, 1996) results of the user study are shown. The overall average score was 73.15%. Looking at the categories that dealt with the UI and the selection metaphor (Q4, Q6 and Q7) a score of 83.33% was assessed, while general usability was a little lower (75%). The selection and grouping metaphor (Q6, Q7) obtained high acceptance, while the highlighting function (Q8) and tag recommendations (Q9) did not.

5.1.6 Conclusion

In the tag2wrap study, collaborative tagging was applied for the task of annotating structured data in web pages. The user interface of the presented tagging system utilized dedicated selection metaphors for tagging of single atomic entities and grouping them in order to compose complex structures. Besides that, a consolidation method for identifying conceptual tags in the user-assigned tag sets was applied in order to support the creation of conceptual structures for describing structured web data. The method was able to select the most fitting concept by exploiting occurrence probabilities from WordNet synsets. As most controlled vocabularies or word nets are incomplete, e.g. missing concepts or synonyms, this approach might be limited in real-world environment.

Furthermore, the user evaluation indicated by the influence of the chosen user group on the quality of the annotations due to their different background and domain knowledge, e.g. football fans might be more appropriate for football web pages. The gained results and the consolidated tag structures can be exploited by information agents to discover similar entities or used by an information retrieval or extraction system as samples of the target structures.

The experiences revealed that in order to obtain sufficient user contribution for such types of semantic annotations on the web, besides the used semantic annotation metaphor for promoting conceptual tagging, the incentive model offered plays a major role for obtaining high user contribution.

5.2 Semantic Layering with HC

Social web applications allow for the accessing and interacting with user-generated contents by making use of implicit semantics based on tags from users, which are attached to uploaded or created resources such as images, music, articles in blogs, etc. As discussed before, user contributed tags form narrow (domain-specific) or broad user-created vocabularies, i.e. folksonomies.

As the experiences with the social tagging environment for tagging semi-structured web pages (presented in the previous section) show, employing a social tagging environment for generating semantic annotations (useful for retrieval or extraction tasks) has serious problems with user motivation, as it is difficult to mediate a clear incentive model to the users. Why should users start to tag web page fragments? What are the benefits they gain?

These questions are the basis of a set of challenges to cope with, if the task of semantic layering has to be embedded seamlessly into a social application context on top of strong incentives. The widely adopted incentive models in social networking applications employed so far are of a high personal and social nature, such as adding and describing contents for self-finding or sharing with others. Returning to our use case of tagging web page fragments, which might not really be fun, playing games is likely to be.

What is essential is to profit from the users common sense knowledge and their contextual reasoning capabilities, which could be exploited in form of games, where the task of semantic tagging is hidden behind dedicated gaming actions or embedded in playing a series of game levels. Humans also have good to excellent skills in tasks of aesthetic judgment, interaction with objects in the real world or decision making guided by intuition.

As a result, serious alternatives for addressing challenges related to core semantic web tasks, such as semantic layering of web sources or linking data with conceptual knowledge, could emerge, if successful designs and strategies for creating “games with a purpose” can be provided. One major benefit of linking data from different knowledge sources or services over their semantics (as discussed before in Section 2.2.2) is that new value-added information can be inferred from the individual pieces, which could have been obtained from diverse sources.

In the following section, after providing a basic understanding about human computation and describing related work for the regarded use cases of tagging and question answering, two simple approaches for integrating the semantic layering task for web data into human computation online games are presented. The first approach applies binary verification for labeling extracted web page fragments. In the second approach pairs of questions and answers (web fragments) are generated in the course of playing a “jeopardy” quiz like online game.

5.2.1 Related Work

Human Computation (HC), a discipline that has its origin in evolutionary computation (Dawkins, “The blind Watchmaker”), is based on the idea of making use of the ability of humans to solve computationally complex tasks, e.g. image recognition.

In principle, human computation establishes a *symbiotic* relationship between humans and computers for solving a higher level problem, such

as, labeling all images on the web. Comparable to folksonomy systems, continuously motivating people to contribute is a major challenge that can be dealt with by applying different incentives and gaming strategies.

Human Computation Systems

HC systems differ concerning the used method, quantitative and qualitative criteria, system design and the form of social organization⁵¹. Furthermore, they can be categorized by their underlying incentive model, which can vary across different use cases: voluntary (e.g. Wikipedia), incentives by money (e.g. Amazon Mechanical Turk), incentives by fun (e.g. Games With a Purpose (GWAP)), no-choice systems (e.g. CAPTCHA/reCAPTCHA, (Von Ahn et al., 2008)).

Regarding the fact that people spend a lot of time for playing, e.g. “9 billion hours solitaire played in 2003”⁵², it starts to become plausible that these wasted “human cycles” - for playing a game or waiting ’til the computer or a certain application is ready for input – can be used for human computation purposes. Luis von Ahn was the first to show that computer games can be an adequate way to motivate people to participate in a human computation grid. The ESP game (von Ahn and Dabbish, 2004) is a two-player game for labeling images on the web based on the principle of “input agreement”, where two players are paired over the Internet and decide on appropriate keywords for a shown image. In case of agreement the players collect points moving to the next round. A similar “game with a purpose” is “VideoTag” (S. Greenaway, 2009), where users tag videos by playing a ESP like game. Another interesting application domain for human computation is natural language processing (NLP). A variety of different sub domains were explored by, e.g. Actionary and TwinMinds (Takhtamysheva et al., 2009), that allow for enhancing the linguistic capabilities of interactive games like denoting specific actions in virtual physical spaces.

Games With a Purpose (GWAP)

Games with a purpose are used to solve the outsourced human computation task by letting users play a game. The players are rewarded by getting points and moving up to higher levels etc. As fun is the main motivation for gaining unpaid volunteers, game design is elementary for creating successful ”games with a purpose”. Consequently, successful adoption of HC in

⁵¹http://en.wikipedia.org/wiki/Human-based_computation

⁵²Luis v. Ahn in a Google Talk in July, 2006.

games requires dedicated models and strategies for seamless integration of the outsourced task into the game flow and the underlying concepts, hence, understanding of game theory principles (Jain and Parkes, 2009) is fundamental. In their work, Krause et al. (2010) use an arcade like action game to analyze and explore the impact of game design and game design theory in HC games, embedding the two challenging tasks of *ontology population* and *synonym detection*.

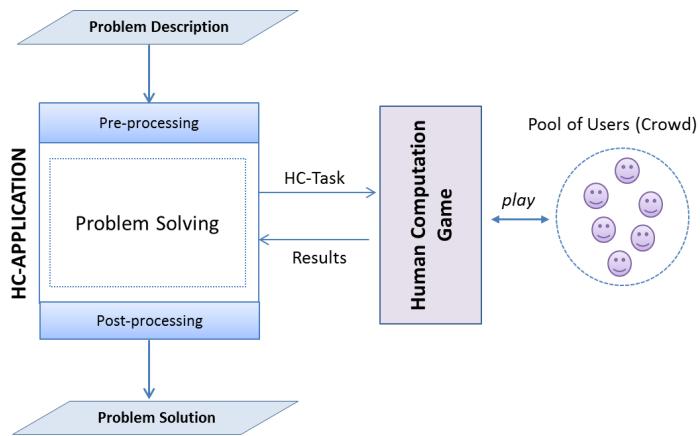


Figure 5.10: Basic Architecture of Human Computation games.

Figure 5.10 illustrates the basic architecture of human computation games that serve a higher level problem solving. The first step consists of a problem description which is processed and translated into an HC algorithm. The algorithm can be executed in the form of one or more task and subtasks. Some computationally complex tasks are split into sub-tasks or parts so that they can easily be solved by humans outsourced to human players who play a human computation game. The returned results are exploited and integrated by the HC application serving the overall problem solution. On the game side, the human computation tasks to be solved are invisible to the users and integrative part of the game play.

GWAPs for the Semantic Web

Games with a purpose for the Semantic Web aim at creating incentive structures and applications for increasing user involvement for core semantic web tasks such as semantic annotation, ontology creation, etc. Although tools such as ontology editors, etc. exist for all these tasks, they are hardly usable by ordinary users and have a steep learning curve. One finding from the social web is, that in order to achieve high user participation for cer-

| Task | Input/Computer | Input/Human | Output |
|------------|---|--|----------------------|
| Annotation | Players are shown a resource (text, image, etc.) and a suitable domain ontology | Players have to select and agree on the appropriate annotation of the resource | Semantic annotations |

Table 5.6: Input-Output relations in a HC Game for semantic annotation (Siorpaes and Hepp (2007))

tain tasks, the users must get valuable rewards for annotating a resource. Furthermore, building shared views of the domain of interest for creating ontologies entails ensuring massive participation in order to be successful. The idea to apply the wisdom of the crowds principle using games to achieve exactly this, is not new, but user-contribution for creating semantic data can only be produced as a by-product implicitly if games serve an intellectual purpose. Therefore, the outsourced tasks have to remain invisible to the users, which is a challenge that game design must reinforce. Hence, gaming should involve reputation for users when exploiting fun, balance contribution with immediate benefits and regard fun and intellectual challenge as predominant user experience, which was explored by Siorpaes and Hepp (2008a).

In conclusion, semantic annotation can be regarded as being in the core of all semantic web tasks. Siorpaes and Hepp (2007) investigated methods for solving a suite of related tasks utilizing multi-player games for the Semantic Web. The regarded resources, e.g. music, videos, image, etc. are embedded into a semantic web game, while the annotation task remains invisible to the user hidden behind a graphical user interface or natural language patterns. Siorpaes and Hepp (2008b) further introduced a generic infrastructure for realizing online games for the Semantic Web, such as, ontology construction, ontology alignment, ontology matching and semantic annotation. For each of the listed game category, a list of tasks is identified to be outsourced to humans. Table 5.6 shows the input-output relations for realizing a semantic annotation game, which is the focus of the next sections.

Question Answering

Existing approaches for answering natural language questions from text are mainly based on NLP methods such as paragraph analysis, named entity

recognition, etc. for extracting appropriate answers. In the SmartWeb project, an open domain question answering system allowing the answering of natural language questions related to persons, locations, and other named entities found in web documents was realized utilizing a web search engine like Google.

An evaluation corpus for the German language was built by exploiting WikiPedia (Cramer et al., 2006). Other approaches such as the Alyssa (Shen et al., 2006) represent statistically-inspired question answering systems. Related experiments were reported for the TREC 2006 question answering track. Furthermore, approaches that are based on ontology-based information extraction from text (Buitelaar et al., 2006) have been researched. Here, the extracted data entities are represented as ontological instances and matched against semantic query representations. The latter realize domain-dependent question answering systems, e.g. for football resources from the web. The Watson system (Ferrucci et al., 2010) represents a combined complex architecture with many of the researched state of the art knowledge inference, analysis, and natural language processing methods.

5.2.2 Playing and Tagging using Binary Verification

Using tags as related keywords for identifying portions of data or information structures on the web can be utilized efficiently for enhancing the retrieval capabilities of search engines or data extractors.

In Section 6.3.3 one main challenge in generating wrappers for extracting structured data from web pages is to identify the most important regions of a page (“semantic regions”). Noise, which results from commercials, banners, navigation menus, etc. can drastically worsen the extraction capabilities of wrappers. As detecting important, content-bearing sections is an easy task for humans, embedding this task into human computation games is of great benefit.

In the following, a simple method for tagging web page fragments or semantic structures similar to the social tagging approach presented in Section 5.1 is described. The basic idea is to embed the collaborative tagging task into a human computation online game (Figure 5.11) based on binary verification. The following sections introduce the concept of binary verification and describe important aspects of the proposed HC game.

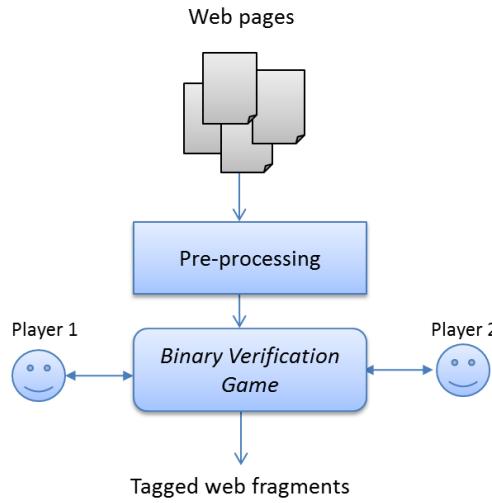


Figure 5.11: Playful Tagging using Binary Verification.

Binary Verification

In binary verification a given object is classified as relevant or non-relevant by the user with respect to one or several other objects. For the regarded semantic tagging task, given web resources such as images, page fragments, etc. are classified as relevant or non-relevant for given terms, e.g. tags or keywords.

A prerequisite of binary verification is given by pre-generated objects and related candidate keywords, which could be generated utilizing search engines, web page segmentation and appropriate automatic data extraction methods. An alternative to the automatic generation of keyword candidates is to apply the human computation paradigm in a training round by showing the users the respective web pages randomly. The entered keywords are collected and used in the second round for disambiguating relevant keywords from non-relevant for given page fragments. In order to automatize the generation of candidate web page fragments, a simple vision-based segmentation algorithm (Cai et al., 2003) could be utilized. The general workflow for applying and exploiting binary verification can be described as follows:

1. Create a corpus of extracted structured data objects from a web page using the methods described in Section 3.2 and a set of related keywords.
2. Utilize a binary verification online game to assign relevant tags for shown resources.

3. Exploit semantic annotations for applications of retrieval, extraction, etc.

Another important issue is game design which has a significant impact on game quality and user acceptance. For the proposed HC game it is important to determine the optimal number of elements to show to the player, the number of elements the user is allowed to interact with and the order of the presented objects. Showing too many elements may confuse the user and lead to reduced interaction speed. If the number of elements that the user can select in one round is too high, players might tend to just randomly select elements. Experiments showed that using around ten to twenty visible elements depending on the available game screen size and the size of each single element is reasonable. The number of selections allowed should be approximately half of this number. Another design aspect is the order of elements shown to the player. Using a random order may help to prevent cheating because fewer assumptions on the game can be made by the players. For instance, players can't always agree on selecting the first five elements because the order is likely to be different for each player.

The FastTag Mock-Up

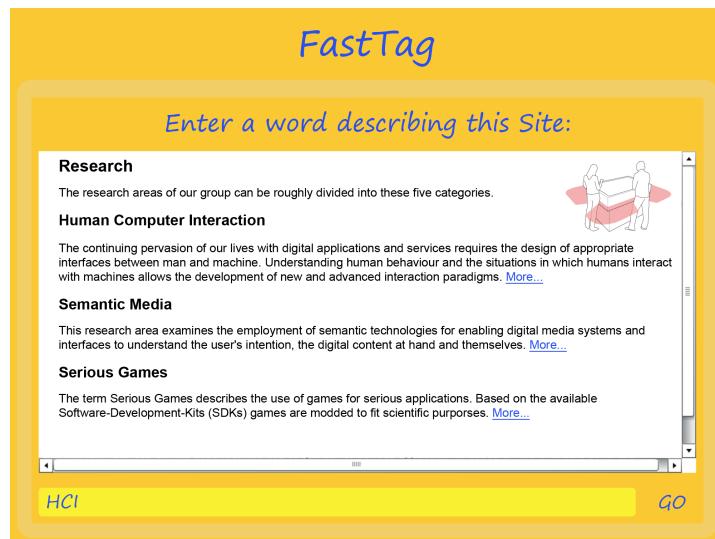


Figure 5.12: Tag Generation Screen.

FastTag, a simple example design for the described approach was proposed by Krause and Aras (2009). The game starts with the first player who needs to enter a tag for a randomly chosen web page fragment as shown

in Figure 5.12. In the next step, a second player joins the game and enters a tag for the regarded resource, i.e. the players are paired.

The game proceeds to the “Agreement Screen” (Figure 5.13). Here a fixed number of extracted web page elements from the formerly shown page are displayed. The task of both players is now to select up to a fixed number of elements that are associated with the previously entered tag. The first round of the game is finished if both players click the “done” - button. Subsequently, each matching selection results in credit points for the players. The game proceeds by changing the roles of the players, i.e. now the second player plays first.

In each round, the first players generate tags for web page fragments and both players associate this tag with a number of page elements. The game is over after a fixed time period, e.g. 5 minutes. During game play, all pairs of associations, together with other statistics, e.g. frequencies are stored. Again, the tripartite model of folksonomies can be utilized to associate players, tags and objects extracted from web pages and analyze their emergent semantics.

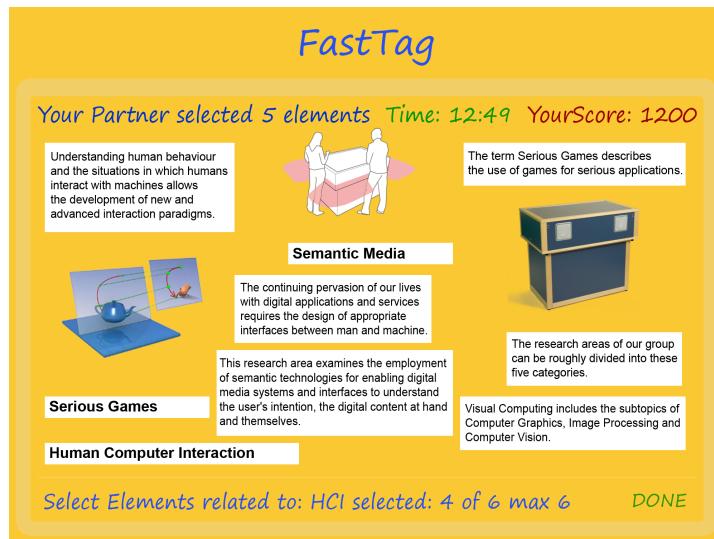


Figure 5.13: Agreement Screen.

5.2.3 Webpardy: Harvesting QA by HC

Answering complex natural language queries beyond those that are factual is still a challenging task for existing Question Answering (QA) systems. Until now researched approaches have used a variety of methods ranging

from statistical analysis to natural language processing, knowledge or logic-based techniques that utilize formal ontologies in order to generate answers for human-entered natural language questions. An outstanding result for question answering has been achieved by the Watson system developed by IBM's DeepQA (Ferrucci et al., 2010) project. The system was tuned for the "Jeopardy Challenge" and was able to beat human players in a TV quiz show.

Although Watson showed remarkable results with the jeopardy question-answering scenario by combining methods such as probabilistic knowledge inference, natural language processing, machine learning and information extraction, as an artificial system nuances of human language, imagination for bridging absent knowledge, neither modeled in a knowledge base nor available at WikiPedia, are still difficult frontiers for machines. For example, during the Jeopardy experiment, Watson failed to find the right question for the following clue: "His daughter and grandson were both premiers, and both assassinated". Obviously the answer ("Who is Nehru?") was easy to find for humans, but challenging for Watson as it could not derive it from the language and the available knowledge alone. Such types of questions and many others can be created by humans within seconds looking at a piece of information, such as a web page or image applying their image recognition, classification skills or using their common sense in addition considering the data in context.

In the following section, a human computation approach for generating question-answer pairs is described. The prototype implementation called "Webpardy" – based on the idea of the popular "Jeopardy" quiz – enables users to contribute to the creation of a corpus of question-answer pairs for harvesting web-based question answering (Aras et al., 2010). Webpardy aims to leverage the "Wisdom of the Crowds" principle for associating information with related questions of any human-thinkable form, which would cover types of questions, e.g. metaphoric knowledge that artificial systems cannot extract or learn from the available data or knowledge sources directly. In contrast to classic question answering systems that work with natural language patterns as answers, in Webpardy answers are represented by web page fragments extracted from the web, while the questions are contributed by the users during game playing.

The Idea of using HC for Question Answering

The basic principle of question answering – in contrast to document level search – is to provide exact or related answers to user given questions.

Chancellor of Germany

From Wikipedia, the free encyclopedia

The **Chancellor of Germany** is the **head of government of Germany**. The official title of the office is *Bundeskanzler* (**Federal Chancellor**).

In German politics the **chancellor** (German: *Kanzler*) is equivalent to that of a **Prime Minister** in other countries. The direct German equivalent of Prime Minister, *Ministerpräsident*, is used exclusively for the heads of government of most German states (called *Länder* in German).

The current Chancellor of Germany is **Angela Merkel**, who was reelected in **2009** after her first election 2005. She is the first female Chancellor. In German she is thus known as *Bundeskanzlerin*. That word was never used officially before Angela Merkel, but it is a grammatically regular formation of a noun denoting a female.

- **Who** is chancellor of Germany?
- **Who** is current chancellor of Germany?
- **What** is german equivalent of prime minister?
- **Who** is head of goverment in Germany?
- **When** was Angela Merkel reelected?

Figure 5.14: A sample web page and associated questions.

So far, existing systems try to extract the answers from a collection of documents applying various techniques ranging from statistical methods, e.g. TF-iDF scheme (Baeza Yates and Neto, 1999, pg. 29-30), co-occurrence analysis (Lux et al., 2007), etc. to natural language processing (Moens, 2006), e.g. paragraph analysis for matching named entities such as of persons, institutions, dates; or analyzing Hearst-patterns (Hearst, 1992) that rely on part-of speech, etc. Figure 5.14 shows an example with answers related to facts from a web page. Questions such as: “Who is the current chancellor of Germany?” could be answered by matching the terms “current”, “chancellor” and “Germany” and looking for named entities of the type person in the proximity of the matches, e.g. whether all three terms are matched, whether the corresponding part-of speech order is correct, etc. In the case of the example question shown, a typical pattern in the form

<TEMPORAL_WORD>-<NOUN/PROFESSION>-<NOUN/LOCATION>

In general, depending on the term matching and paragraph analysis, a confidence score is calculated for each answer candidate. In the case of unstructured text, missing semantic relations and ambiguity of natural language makes it difficult to answer complex questions applying such methods,

resulting in inaccurate or incorrect results.

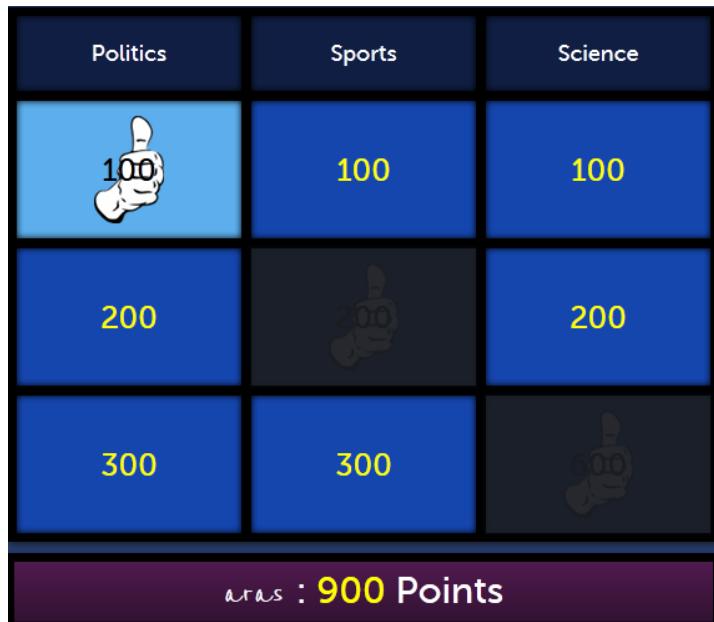


Figure 5.15: The Webpardy “wall” UI for selecting category/difficulty level.

Experiences in picture labeling, classification, etc. using human computation online games have shown that humans are good at applying their common sense, knowledge and intuition to classify data. In Webpardy, the idea is to exploit the capability of a human to associate questions to data or information fragments utilizing a web-based game.

For the regarded question-answering task, the goal is to enable users to contribute to the generation of questions for existing web resources, i.e. web page fragments by implicitly labeling them with “questions” in order to collect points, move to the next level, etc. The idea behind the Webpardy online game was inspired by the Jeopardy TV game, which was also challenged by IBM’s Watson system. The user is first asked to select a category as well as a level of difficulty from a matrix screen as depicted in the left screenshot of Figure 5.15. In the next step, a resource, e.g. a web page fragment is chosen for the player by the game. Appropriate web page segmentation techniques can be used for this task. After that, the player has a limited amount of time (a few seconds) to enter a question for this resource as depicted in Figure 5.16. In order to create the underlying document corpus for a particular domain, which corresponds to a category from where the resources are retrieved, focused or topic crawlers could be utilized (Liu, 2007d, pg. 292).



Figure 5.16: Entering questions for given web page resources/fragments.

Conceptual Framework of a Jeopardy Game for the Web

Webpardy's data model is based on a similar tripartite model as used in collaborative or social tagging. Here, the ternary associations are represented by edges that connect a given user with a particular resource using a particular question. The basic workflow implemented in the prototype system consists of a two steps approach: firstly, generating questions, and secondly, answering questions. Figure 5.17 illustrates the generation of questions.

Existing web pages or resources R are segmented into reasonable fragments or sections F that are loaded by the Webpardy game to be presented to a group of users U . A user u_i once registered to the system can start playing the game by selecting a given category and a level of difficulty, which is chosen randomly for the bootstrapping. Difficulty levels are estimated from implicit user feedback such as, duration for entering the question, length and complexity of the sentence. After clicking the desired cell on the wall a screen with the fragment f_{sj} ⁵³ and an input field is presented. Next, the user has to think of a suitable question during a countdown, e.g. 10 seconds. A provided question sentence q_k is validated according to a multi-step procedure before the question-resource pair is stored in the database together with its confidence score and further parameters. Finally, the created knowledge base with the set of question-resource pairs can be queried

⁵³s: index of the section/fragment, j: index of a resource from where the fragment was retrieved

and matched against a given user question. Here, sentence similarity analysis (Higgins and Burstein, 2007) and additional methods for semantic analysis can be applied for gathering previously stored question-answer pairs containing related answer fragments for the entered question.

Verification of Questions

The validation of questions in Webpardy again can be modeled with a two phase procedure. In the first phase, a trust score is generated for each user. The score is calculated upon a set of existing relevant resource-question pairs. A similarity comparison⁵⁴ of questions given by players with these ground truth questions allows the quality of the questions collected to be estimated. The resulting trust values for a particular user indicate whether the provided questions for unknown resources are valuable or not.

In the second phase, where unknown resources need to be annotated with questions, the following reference procedure can be applied for calculating the overall confidence score S for each (q_k, f_{sj}) pair:

1. Check whether every entered <word> is really a word in a given language, e.g. using a thesaurus.

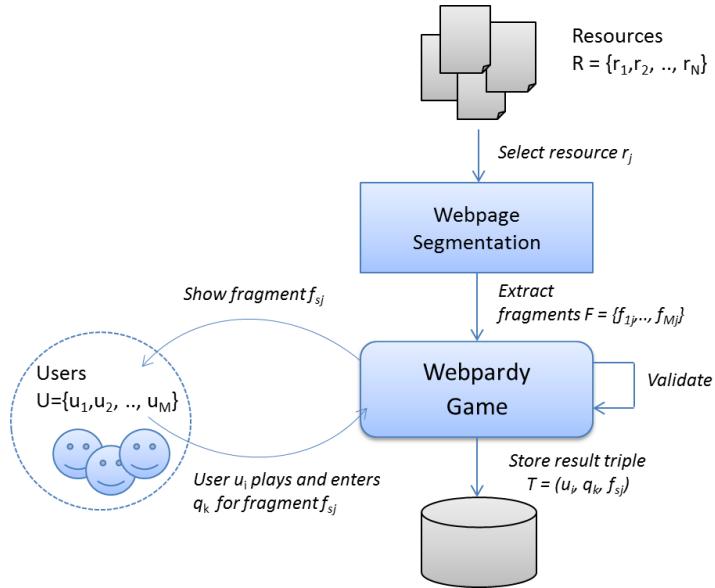


Figure 5.17: Generating question-answer pairs in Webpardy.

⁵⁴Similarity can be calculated based on the content bearing noun phrases or in its simplest form using word similarity. Besides matching the content terms, external knowledge bases, e.g. thesaurus, word net, etc. can be consulted.

2. Check whether the entered sentence contains a question word, e.g. why, who, etc. as well as a question mark.
3. Check the TF-iDF value for each term of the sentence. Verify whether the individual terms are relevant for the entire corpus from where the original page was retrieved and in particular, for the web page itself. Additionally, a word net containing taxonomic relations as well as synonyms could be queried to verify the relevance of the provided terms.
4. Match Hearst-pattern (Hearst, 1992) for typical question templates.

This scoring scheme was implemented in a more simple form, containing the following metrics: word count and length, spelling-grammar analysis and question words analysis. A metric for evaluating the relevance of the content of a web page, e.g. TF-iDF together with stemming, considering WordNet synonyms, etc. and relevance for the regarded resource, was not implemented in the first prototype.

Double Webpardy - Reversed Playing Mode

Webpardy uses the following human computation idea for validating entered questions. A list of alternative result questions is presented to the user for a given section. The candidates are generated by choosing two results for the same section with good confidence scores and a third bad result (low confidence score) from another topic area. The user then has to select the best fitting question from three candidates. Internally, Webpardy maintains a qualification scheme with 3 integer attributes {3 (good), 2 (middle), 1 (bad)}. Depending on the users choice and the previous confidence score, the election results in a double increased score or the score is decreased by the double amount. For the user this means that he/she gets, for example, 400 points instead of the 200 as the value of a given section, or his/her current score is decreased by 400 points.

This method allows the verification of whether the assumption about the three candidates match with the players choices or not. The quality of the given question-resource pairs is evaluated based on the accuracy of the responses provided. The revers mode was integrated as a bonus level allowing the selection of one of the shown questions (see Figure 5.18). The selection currently influences existing results by boosting the confidence score of the selected result, i.e. the question.

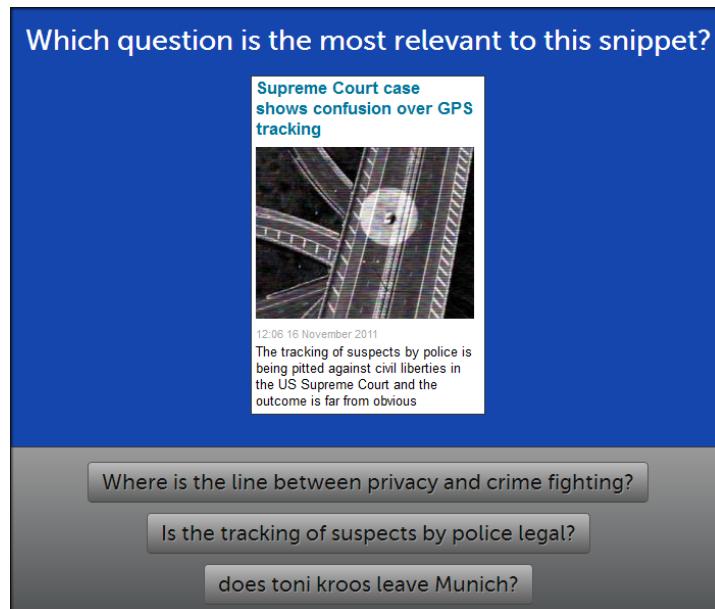


Figure 5.18: Double Webpardy Mode for validating entered results.

Data Collection

For evaluating Webpardy, a corpus of web page fragments (sections) was created via a web scraping tool and integrated into the game. Each section is characterized by a unique id, the visual parameters x, y, height and width of the web page screen and the id (URL) of the belonging resource. Table 5.7 gives an overview of the collected sections for the regarded topics.

Based on this corpus several datasets were collected in 2011 and 2012 by letting users play the game and recording the results for each played section fragment from the first four categories. The datasets in 2011 were gathered by inviting users to participate and try the game via e-mail. Although no data collection could be obtained this way for a consecutive period of time, these spot tests helped to evaluate some of the parameters of the system, such as, question qualification by the reverse Webpardy mode, where the entered user questions are evaluated by other users (human-based verification). In order to avoid biases, power users who gained familiarity with the system were excluded in order to focus on the average players.

Data was collected in a consecutive period of time in a facebook social networking environment in March and April 2012. The collected questions as well as their confidence scores have been analyzed.

| topic | 2010 | 2011 | Σ |
|---------------|------|------|----------|
| politics | 8 | 50 | 58 |
| sports | 3 | 51 | 54 |
| entertainment | 5 | 59 | 64 |
| science | - | 57 | 57 |

Table 5.7: Number of sections per topic field.

5.2.4 Evaluation and Results

The aim of the evaluation was to collect a set of questions, which were entered by the players, for the randomly selected set of web page fragments. General information about the players, the entered questions and related parameters as well as the game scores per player were recorded and evaluated in form of a gaming session. The evaluation resulted in an overall of 432 entered questions for given web page fragments (sections) from the categories of politics, sports, entertainment and science extracted from the web.

Test Parameters and Scoring

After the experiment the following questions served to analyze the quantitative results and find answers for the focused evaluation goals:

- How many questions were entered for each fragment on average? (average number of questions per section, i.e. question contribution per section)
- How many users tagged each fragment? (average participation of users)
- Quality of questions evaluated by the system (average confidence score per section⁵⁵)
- Quality of questions evaluated by the users (Double Webpardy - Reversed Playing Mode)

In order to calculate the confidence scores for each entered question q_k of user u_i for fragment f_{sj} the following general formula⁵⁶ was applied:

⁵⁵For reasons of simplicity and in order to apply a visual scheme the user is familiar with, a three level scoring bad-middle-good was used. This scoring was returned as visual feedback (thump up, -down, -middle).

⁵⁶On the basis of the general formula, a baseline with default weights was implemented in the Webpardy prototype.

$$S(u_i, q_k, f_{sj}) = \frac{\sum_{i=1}^N (w_i \cdot s_i)}{N}$$

f_{sj} stands for the fragment s extracted from the resource (page) j. Furthermore, w_i are the weights of the N(=4) single metrics $s_i \in [0, 1]$. Currently, the four scoring criteria s_i measure grammar, spelling, question word and unique word count in order to verify the newly entered questions. The confidence score $S(\cdot)$ results from summing up the weighted scores of the normalized single metrics s_i and normalizing it in the interval [0,1]. Hence, a score of 0 means bad and a score of 1 good. The weights for the individual scores have been adjusted and optimized manually and set to constant factors for the experiments. Finally, the score that the user obtains as points P is calculated by multiplying the confidence score with the value $val(f_{sj})$ of a fragment that was assigned by the game:

$$P(u_i, q_k, f_{sj}) = S(u_i, q_k, f_{sj}) \cdot val(f_{sj}).$$

Experiment 1: Dataset 2011

The following results in Table 5.8 show the calculated average confidence scores $avg\ S(\cdot)$ for the questions entered during the evaluation period in 2011 and some additional statistics: number of sections (nrOfSections) and number of questions (nrOfQuestions).

| Topic | nrOfSections | nrOfQuestions | avg C(.) | avg S(.) |
|---------------|--------------|---------------|----------|----------|
| politics | 58 | 60 | 1.03 | 0.96 |
| sports | 54 | 57 | 1.05 | 0.96 |
| entertainment | 64 | 22 | 0.34 | 0.85 |
| science | 57 | 30 | 0.53 | 0.93 |

Table 5.8: Evaluation of the data collected 2011.

First, the average number of question per section/per topic (average contribution) $avg\ C(\cdot)$ was calculated. Next, the average confidence scores $avg\ S(\cdot)$ were analyzed together with the user judgements for the entered questions.

For the first two categories, an average contribution per section above 100% and a confidence score of about 95% was calculated by the system. The user contribution for the last two topics was far less. Nevertheless, an average confidence score of above 85% was obtained for the entertainment and science web fragments.

Besides focusing on the average confidence scores for the entered questions, the average qualification of the entered questions through user judgments in the reverse Webpardy mode was analyzed.

Testing Double Webpardy (Reversed Playing Mode):

Table 5.9 shows an example election (voting) $Q(\cdot)$ for the reversed Webpardy mode, where the second and third question were preferred by the users. Considering that in this example at minimum one player voted for each of the questions, resulting in an increased confidence score of about 2.0.

| Question | Section | avg S(.) | Q(.) |
|---|---------|----------|------|
| Does Toni Kroos leave Munich? | 2 | 0.87 | 1 |
| Is Berlusconi still capable of being prime minister of Italy? | 21 | 2.0 | 2 |
| How did he win the elections? | 21 | 2.0 | 3 |

Table 5.9: Double Webpardy election example.

Experiment 2: Dataset 2012

In the following Table 5.10 the results of the data collected in 2012 and during the consecutive experiment (numbers in round brackets) are shown.

| topic | nrOfSec | nrOfQue | avg C(.) | avg S(.) |
|----------|---------|---------|-------------|-------------|
| politics | 58 | 65 (36) | 1.22 (0.62) | 0.92 (0.91) |
| sports | 54 | 67 (36) | 1.24 (0.67) | 0.87 (0.88) |
| science | 57 | 64 (33) | 1.12 (0.58) | 0.88 (0.89) |

Table 5.10: Average scores from the 2012 experiment(s).

The average confidence score in this experiment was calculated as approximately 0.90, which can be interpreted as being good. Despite the deployment in facebook, only a contribution rate around 60% could be achieved regarding the number of sections in play, i.e. not all sections were played during that time. Looking at the number of sections, each user tagged on average (average contribution rate) it can be seen that in the facebook experiment the rate is about 14 sections per user, while at the same time approximately 2.7 users tagged each section on average (average participation rate).

Discussion of the Results

The results of this evaluation show that the game was played successfully (average high score per user: 1590 credit points) several times by the spot players and by 12 players during the consecutive experiment. The outcome of the experiments comprise relevant and interesting questions related to the web fragments shown with good confidence scores.

| question type | 2011 | 2012 | 2012fb |
|---------------|------|------|--------|
| Who | 40 | 16 | 9 |
| What | 59 | 40 | 20 |
| Where | - | 7 | 3 |
| Why | 10 | 30 | 12 |
| Which | 8 | 8 | 2 |
| How | 8 | 24 | 15 |
| Do | 31 | 30 | 17 |
| Did | 13 | 16 | 9 |
| Is | 63 | 52 | 31 |

Table 5.11: Most prominent question types.

Concerning the qualitative value of the questions with respect to their complexity, a deeper analytical and semantic analysis of the question-answer pairs is needed. Bearing in mind the available time of approximately 10-20 seconds to enter a question, the average word count for the given questions was computed as 7 words per question. Table 5.11 shows the most prominent question types entered.

Furthermore, it is necessary to evaluate the user judgements obtained from the reverse Webpardy mode more systematically with sufficient user participation for a variety of entered question types, which could not be focused intensively during the experiments. Table 5.12 shows an excerpt of the results from 2012 with interesting questions, which have been upgraded by the players through the reversed Webpardy mode. The numbers with high user votes lead to increased confidence scores for the questions (6,7,11 and 13), while mixed voting where a question not always received the best vote, have moderately increased scores (3,4 and 6, and 14). It should be noted that the confidence scores of the users influence the scoring as well. Several scoring from the same users lead to a slight increase than voting from distinct users. In order to prevent the score of a resource increasing dramatically, a dampening factor needs to be introduced.

As a result, the short experiments showed the potential of a human

computation game like Webpardy for the regarded question answering task. Despite this, the experience and findings in Webpardy indicate that a long term incentive model for obtaining high quality meta-data in the form of questions attached to web information fragments can only be established within a professional deployment environment and search functionality in the form of a human computation based question answering system, which is a challenge for future work.

| Nr. | Question | avg S(.) | Q(.) |
|-----|---|----------|--------|
| 1 | Can genes relate birds to songs? | 1.0 | - |
| 2 | What can vampire bats sense in their preys? | 1.0 | - |
| 3 | How did the win the elections? | 2.0 | 4 (3) |
| 4 | Is Berlusconi still capable of being prime minister of Italy? | 2.0 | 4 (2) |
| 5 | How did Sepp react to the message that the FIFA is corrupt? | 0.94 | - |
| 6 | What is the name of the FIFA president? | 1.98 | 15 (2) |
| 7 | When will Wayne Rooney leave Manchester United? | 14.0 | 13 (3) |
| 8 | Why is the Wikileaks website under attack? | 1.0 | - |
| 9 | Who was Winston Churchill? | 4.0 | 3 (3) |
| 10 | So smoking while doing daily exercises is healthy? | 0.167 | - |
| 11 | Is professional sports full of corruption? | 13.0 | 15 (3) |
| 12 | How much funding will the government put into disease research? | 1.0 | - |
| 13 | What do tea party members believe? | 8.97 | 7 (3) |
| 14 | Is barack obama a socialist? | 1.0 | 7 (2) |

Table 5.12: Some interesting sample questions entered into Webpardy.

Qualitative User Feedback

After each completed gaming session users were asked about the understandability of the task, its difficulty level, the registration process and whether it would be beneficial to get more credit or bonus points for accomplishing certain things in game play, such as playing a whole round, entering questions without spelling errors, etc. At the same time, users wanted to get feedback from other users and be able to compare their contributions with those from other users. Besides that, factors for introducing competitive challenges like time restrictions, difficulty level, etc. are requested in order to motivate and

excite users for the task at hand. As users wanted to proceed quickly to the next levels, input assistance for formulating user questions was seen as beneficial. Another factor was the language the users are familiar with. German users, for example, preferred to enter questions related to German web sites, as most of the players could express their thoughts more quickly in their mother tongue. Besides that, fun was regarded as the main factor for playing the game, which is closely affected by game design, the graphical interaction and the intellectual challenges in playing.

5.3 Conclusion

In the research described, collaborative tagging and the human computation paradigm was applied to the task of semantic annotation of web page fragments. The proof of concept applications described, demonstrated that it is possible to exploit human skills by employing a social network or crowd of users (e.g. players of a game) for the task of semantic annotation of informative fragments inside web pages. A major benefit of human-based approaches is that they are able to introduce new aspects to existing data as different users can have multiple views and background knowledge. Such additional aspects or complementary contextual information could be beneficial for retrieval or exploration of web resources.

The approach described in Section 5.1 showed that the task for tagging structured or semi-structured data in web pages has two main challenges which have been tackled in a collaborative tagging environment.

Firstly, tagging semi-structured data is more challenging than, for example, assigning user tags to bookmarks. Although in related work, which was described in Section 3.3, visual annotation interfaces for experts have been researched, they are not applicable to a social networking environment due to their complexity.

The approach which has been presented and evaluated, employed a simple two step visual annotation metaphor based on tagging single atomic entities and grouping via dedicated selection metaphors. For atomic tagging, a method for selecting content bearing leafs of the HTML document tree was used, while a grouping metaphor for selecting several previously annotated atomic entities was realized via a bound box metaphor. Consequently, the aim was to support the creation of conceptual structures by tagging single entities and grouping them in order to obtain complex structures.

Therefore, obtained tag structures must be consolidated in order to dis-

ambiguate conceptual tags from tags that represent additional views or a property of the tagged resource. Here, synonyms from an external knowledge base (WordNet) have been used for extracting and disambiguating conceptual semantic descriptions robustly from the assigned user tags. One finding was, that occurrence probability derived from WordNet synsets is not an appropriate selection criterion for disambiguating the correct hypernyms for all cases. Hence, alternative methods based on enrichment and disambiguation employing a collaborative or human-computation based approach should be considered.

Second, in the course of the experiments and the usability evaluation, the user motivation problem and missing incentives was identified as another major challenge for applying social tagging for semi-structured data fragments, which has led to further research of applying a game-based human computation approach.

Human computation games can be regarded as serious alternatives for embedding semantic annotation tasks, as the prototypes and their evaluation in Section 5.2 showed. One finding was, that semantic descriptions of users could go beyond simple labels, if a dedicated social tagging environment, or an appropriately designed HC game is provided.

The Webpardy game – based on the idea of the famous Jeopardy quiz – was employed for collecting question-answer pairs to be exploited by question answering systems. Moreover, the implemented HC games showed that the tripartite model applied in social tagging applications as described by Mika (2005) can be successfully adopted in human-computation tasks as well.

Chapter 6

Semantic QA in a Dialogue System

A dialogue is regarded as a natural form of communication between humans. For this reason it is no wonder that researchers and industry try to develop dialogue systems (e.g. Watson) aiming to imitate a conversational metaphor for realizing human-computer interaction.

Applying a conversational metaphor to interaction with web information is widely based on a knowledge-based process in order to allow for interpreting and reasoning over (spoken) natural language questions of users and finding related facts and answers from the available knowledge sources. But, as elucidated before, the web mostly lacks semantic information.

In a question answering scenario – with the web as the main source of knowledge – users of a dialogue system expect that their natural language questions are understood by the system and answered just in time by exploiting available relevant information on the web. Also, users of a dialogue system expect to get up to date information, as using a regular search engine will rarely return precise answers but just a few links that are more or less related. Hence, the answers to the user's natural language questions must be extracted from one or more web documents accessing the current status of a website of interest, e.g. the status of a web page showing the events in a football (soccer) match that could change any minute. Besides that, as relevant pieces of information can be found on different web sites, a way for maintaining semantic access to these information sources needs to be guaranteed.

In this chapter, it will be shown that software agents that employ wrap-

per technology are able to communicate with a knowledge-based dialogue system by interpreting the user's natural language question and extracting relevant answers by means of formal semantics.

The main objective and challenge of the work described in this study is therefore twofold:

- to investigate how semantic wrapper technology can be employed to answer natural language questions from semi-structured web pages at query time, and
- to show how an agent-based semantic wrapper system can be deployed and integrated into a knowledge-based real-world dialogue system (SmartWeb) for the task of question-answering.

Although data extraction and wrapping techniques employing an ontology (e.g. Thresher/Haystack, see Section 3.2.1) have been researched before, a question-answering pipeline in context of a real-world dialogue system based on a complex ontology has not yet been researched. Breaking this semantic question answering pipeline down into pieces, the focused tasks comprise analyzing the user's natural language query, extracting semantic answer candidates and scoring of (question, answer)-pairs.

Roadmap

In Section 6.1, the context of the research described herein, is first explained by introducing the general functional architecture of the SmartWeb multimodal dialogue system, the dialogue interaction and relevant components that interact and inter-operate with the Semantic Wrapper Agents (SWA) subsystem, which was implemented based on the models and methods investigated herein. Thereafter, the semantic wrapper agents approach based on a wrapper-broker agent architecture is introduced in Section 6.2. Important subtasks dealing with the wrapper generation (Section 6.3), the semantic transformation of the extracted instances to the target representation in the SmartWeb ontology (Section 6.4) and the scoring and selection of the answer candidates (Section 6.5) are explained more detailed separately. The deployment of the SWA system into the SmartWeb dialogue system is presented in Section 6.6. Finally, in Section 6.7 the findings and results of the presented approach are discussed.

6.1 Background: The SmartWeb project

The goal of the SmartWeb project (Wahlster, 2004) - a joint project of several industrial and academic partners funded by the German Ministry for Education and Research (BMBF) - was to realize a multi-modal mobile broadband access to the Semantic Web.

In SmartWeb,⁵⁷ a complex multi-modal dialogue system was developed for allowing users to interact with the Web, using natural language queries based on formal semantics. The focused use case was dialogue-driven question answering for interacting with multiple web-based knowledge sources employing a single complex target ontology for knowledge representation, processing and interaction. The used semantic information sources were: a (factual) knowledge base, open-domain question answering based on passage retrieval from search engine results and analysis based on natural language processing techniques, diverse web services, etc.

The following sections describe first the general SmartWeb system architecture and the dialogue interaction process in SmartWeb. Thereafter, the interfaces to the Semantic Wrapper Agents (SWA) subsystem, comprising the representation of the user's speech input as a semantic query, contextual information, representation of knowledge as ontological instances in the SWIntO ontology and the semantic mediation in the SmartWeb dialogue system, are explained.

The related concepts and components have been developed by partners in the SmartWeb project, and therefore, represent interfaces to and from the researched semantic wrapper agents system. Hence, common work and contributions to the general SmartWeb architecture within this research will be stated explicitly, if it exists. Otherwise, the elucidated aspects are related to inter-operating subsystems and components of the SmartWeb system that interact with the researched agent system. The dialogue and mediation components are relevant, as they are responsible for processing the semantic representation of the user's natural language question - which needs to be analyzed and processed by the agent system in order to find appropriate answers from the wrapped web sources - and reassembling the answers returned to the user. Thereby, the SWIntO ontology represents the *lingua franca* in SmartWeb. Hence, extracted information structures must be encoded with the defined semantics in order to be understood and processed by the dialogue system.

⁵⁷<http://www.smartweb-projekt.de/>

6.1.1 System Architecture and Dialogue Interaction

The SmartWeb system consists of a distributed architecture with three basic processing blocks: the client, dialogue server and the retrieval subsystems, as described by Reithinger et al. (2005). The client-side software is responsible for the graphical user interface and user input (speech, etc.), and runs on suitable mobile devices such as smart phones. On the server side resides a suit of multi-modal recognizers, the dialogue system, speech input, the speech synthesis, and several retrieval subsystems.

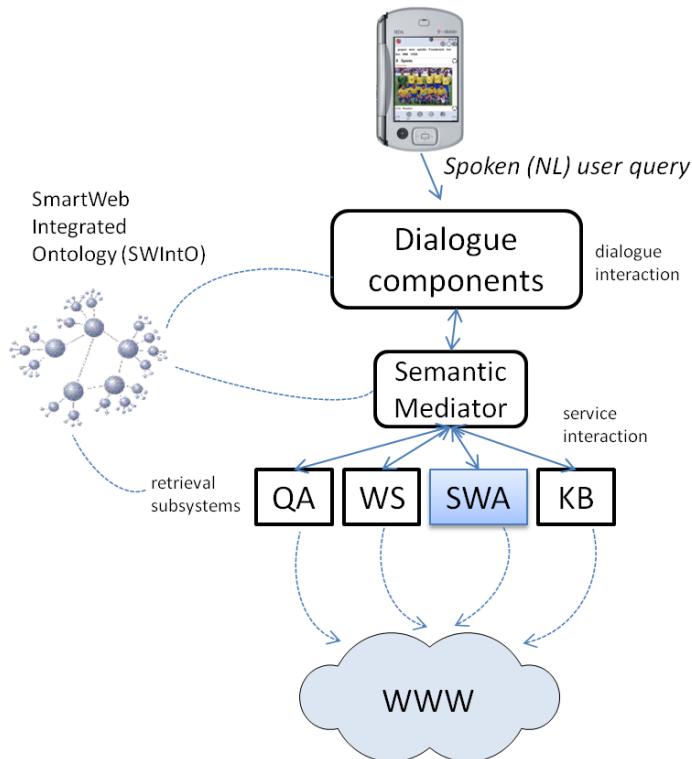


Figure 6.1: SmartWeb - system architecture overview.

Dialogue Interaction

Figure 6.1 shows a simplified view of the technical SmartWeb system architecture, comprising a smartphone with the client-side software, the dialogue components, the service mediator and the information retrieval-extraction subsystems. System-wide interoperability is achieved by employing an integrated ontology covering abstract levels as well as more-specific domain-dependent facts and relations.

A spoken natural language query passes through the speech recognizers, resulting in n-best word chains (best speech hypothesis), which are then parsed by a speech interpretation module (SPIN) (Engel, 2002). The latter generates an ontological representation for the analyzable parts of the n-best word sequence, which is used to form a semantic representation of the spoken or textual user query. The query is then sent by the dialogue system to the semantic mediator component, which is responsible for communicating directly with the retrieval subsystems for open domain Question Answering (QA) based on a web search engine, web services (WS) such as weather service, Semantic Wrapper Agents (SWA) and the SmartWeb Knowledge Base (KB) for the football domain. The semantic mediator reassembles and integrates the answers from the retrieval subsystem in order to obtain a combined and integrated semantic representation of the query results that are returned to the dialogue manager. Usually, the answers are processed by the dialogue system and prepared for a Text-To-Speech (TTS) component that generates the speech output for the client. Table 6.1 shows a sample interaction sequence of the SmartWeb dialog system.

| Turn | Actor | Dialogue |
|------|-------|--|
| 1 | U | "When was Germany world champion?" |
| 2 | S | "1954, 1974, 1990, 2003" |
| 3 | U | "Where?" |
| 4 | S | "Switzerland, Germany, ..." |
| 5 | U | "And Brazil?" |
| 6 | S | "1958, 1962, 1970, 1994, 2002" |
| 7 | U | Pointing gesture on player + "How many goals...?" |
| 8 | S | "Scored none in championship 2002" |
| 10 | U | "What's the state Germany against Italy at the moment?" |
| 11 | S | "0:2" |
| 12 | U | "..." |

Table 6.1: A sample dialogue in turns in SmartWeb (Sonntag et al., 2007) between a user (U) and the system (S) as the participants.

Looking at the dialogue, it can be seen that different tasks have to be resolved by the dialogue components for speech interpretation, analysis and the retrieval subsystems connected, in order to resolve the questions posed to the system. In the example above, the first two questions are resolved by using deductive reasoning via the ontological knowledge base modeled in

OWL (Krotzsch et al., 2006). While utilizing the knowledge base allows the use of rich but static implicit knowledge, user question number 10 requires the consultation of online sources that are highly topical, such as sports results web sites and other topical information sources on the Web. As will be shown later, the semantic wrapper approach allows the current state of web pages to be queried by extracting and analyzing their contents almost at query time.

6.1.2 Speech Recognition and Semantic Queries

Prior to using the semantic wrapper agents (SWA) subsystem for querying wrapped web pages, a semantic representation of the user query must be created by the SmartWeb system. As stated earlier, the speech recognition hypothesis (n-best word list) is processed by the SPIN semantic parser applying appropriate transformation rules in order to obtain a semantic representation of the natural language query by utilizing the system wide ontology SWIntO, i.e. a set of ontology instances that represent the user utterance. SPIN, therefore, operates via 544 rules and 2250 lexicon entries and more in order to process an utterance and to create a semantic representation (Engel, 2006).

Considering that in the dialogue example shown, contextual information has to be taken into account implicitly, the dialogue system consults a dedicated component of the dialogue manager called SitCom (Porzel et al., 2006), which is responsible for modeling situation and context of the system by taking into account pragmatic knowledge. The acquired contextual information from the systems sensors are given by domain of discourse, time and place, weather conditions, etc.

| Nr | Input | Semantic Translation |
|----|------------------|---|
| 1 | Germany | <code>Country(name:GERMANY)</code> |
| 2 | \$C=Country() | <code>Team(origin:\$C)</code> |
| 3 | when | <code>TimePoint(variable:QEVariable(focus:text))</code> |
| 4 | \$TP=TimePoint() | <code>QEPattern(patternArg:</code> |
| 5 | was \$TM=Team() | <code>Tournament(winner:\$TM, happensAt:\$TP))</code> |
| 6 | world champion | |

Table 6.2: A sample semantic translation of a recognized natural language question.

The example in Table 6.2 (see Sonntag et al. (2007)) illustrates the se-

mantic translation of the user question “*When was Germany world champion?*” by the speech interpreter (SPIN), applying 4 distinct rules. The first rule translates the word “Germany” to the ontology instance Country. In the second step, countries are transformed to teams as each country represents a team in the football domain. The dialogue context and other context is resolved a priori by the dialogue subsystem. The word “when” is recognized as an instance of the class **TimePoint** which is marked as a question. The most complex part of the semantic translation is the verbal phrase “<TimePoint> was <Team> world champion”, etc.

Ontologically described results, i.e. instances of the SWIntO ontology gathered via the retrieval subsystems use the same SPIN parser and a TAG grammar⁵⁸ in order to verbalize the query results.

6.1.3 Knowledge Representation and Ontology

Within a complex dialogue system like SmartWeb it is important to ensure system wide interoperability by using established standards for structured and semantic knowledge representation and processing. This is true for dialogue communication and messaging as well as the exchanged or processed contents. In SmartWeb, a response-request scheme was implemented as dialogue acts for the question-answering use case.

SWEMMA

The Extensible MultiModal Annotation markup language EMMA⁵⁹ was utilized to provide containers for the ontological instances of the user query, that are based on the ontology used system-wide. Besides that, in order to allow the representation of the systems result instances – using **swemma:result** and **swemma:status** – retrieved from the retrieval subsystems, an extension called SWEMMA was developed. EMMA/SWEMMA was modeled using RDF-S on the ontological level. The example in Figure 6.2 shows the semantic representation of a natural language question in SWEMMA, as part of the discourse ontology, based on the W3C EMMA standard in order to model dialogue interaction.

For the semantic query processing, relevant fields are given by the focus of the query (**:focus**), which can be exploited by the semantic wrapper agent system for identifying the information parts of interest in the answer instances, here: a **:DivisionNationalTeam** ontology instance. The major

⁵⁸<http://www.cis.upenn.edu/~xtag/>

⁵⁹https://de.wikipedia.org/wiki/Extensible_MultiModal_Annotation_markup_language

```

<rdf:RDF
  xmlns:jms="http://jena.hpl.hp.com/2003/08/jms#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:j.0="http://smartweb.org/ontology/emma#"
  xmlns:j.1="http://smartweb.org/ontology/discourse#"
  xmlns:j.2="http://smartweb.org/ontology/sportevent#">
  <j.0:Emma>
    <j.0:container>
      <j.0:Interpretation>
        <j.0:container rdf:resource="http://smartweb.org/ind#i2"/>
        <j.0:lang rdf:datatype="#string">de</j.0:lang>
        <j.0:end rdf:datatype="#long">1114605785</j.0:end>
        <j.0:turnId rdf:datatype="#string">42</j.0:turnId>
        <j.0:start rdf:datatype="#long">1114605781</j.0:start>
      </j.0:Interpretation>
    </j.0:container>
    </j.0:Emma>
    <j.0:OneOf rdf:about="http://smartweb.org/ind#i2">
      <j.0:container>
        <j.0:Interpretation>
          <j.0:container rdf:resource="http://smartweb.org/ind#i4"/>
        </j.0:Interpretation>
      </j.0:container>
      </j.0:OneOf>
      <j.1:Query rdf:about="http://smartweb.org/ind#i4">
        <j.1:text rdf:datatype="#string">wer war 1990 Weltmeister</j.1:text>
        <j.1:dialogueAct>
          <j.1:Question/>
        </j.1:dialogueAct>
        <j.1:focus>
          <j.2:DivisionNationalTeam rdf:about="http://smartweb.org/ind#i5"/>
        </j.1:focus>
        <j.1:content>
          <j.2:WorldCup>
            <j.2:heldOn rdf:datatype="#string">1990</j.2:heldOn>
            <j.2:winner rdf:resource="http://smartweb.org/ind#i5"/>
          </j.2:WorldCup>
        </j.1:content>
        <j.0:confidence rdf:datatype="#float">0.75</j.0:confidence>
      </j.1:Query>
    </rdf:RDF>
  
```

Figure 6.2: “Who won the soccer world championship in 1990?” as RDF.

semantic query structure is derived from the ontology instance encoded under `:content`, which indicates instances of `:WorldCup` that took place in the year 1990. At the same time this instance corresponds with the *semantic class* to be exploited by the retrieval subsystems. For this example, the wrapper agents extract and generate semantic instances for the semantic class `:WorldCup`, containing available information related to the regarded tournament to be queried by the users. Bear in mind that the reference under `:winner`, is the same as under `focus:`, i.e. the focused semantic structure in a particular answer instance.

SWIntO

Knowledge in SmartWeb, e.g. web content is represented via the newly developed SmartWeb Integrated Ontology (SWIntO), which is based on two well established foundational ontologies, namely the highly axiomatized Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) (Gangemi et al., 2002) and the Suggested Upper Merged Ontology ontology (SUMO) (Niles and Pease, 2001) that form the SmartWeb foundation ontology SmartSUMO (Cimiano et al., 2004). Therefore, DOLCE and SUMO had to be aligned and integrated following some necessary modifications. The DOLCE part of SmartSUMO required minor modifications, while featuring also various DOLCE extensions, e.g. the ontology of plans and a module called *Descriptions and Situations* (Gangemi and Mika, 2003). In addition, a domain-independent layer that consists of a range of branches from the less axiomatic SUMO ontology has been included, which is known for its intuitive and comprehensible structure. For describing specific system-relevant domains such as the discourse ontology, the intensively supported *sports-event*, and *navigation* domain, dedicated ontologies have been modeled under SmartSUMO and used to describe domain-specific classes, their properties and domain-specific ontology instances.

Moreover, SWIntO features a special ontology design pattern for denominations, since every entity can have different names and abbreviations. Furthermore, there is a distinction between non-physical endurants and physical endurants. On the one hand, there are the physical endurants, such as **Man** and on the other hand, there are roles, such as **FootBallPlayer**, which are non-physical endurants. A man can play different roles during his life. This results in a rather complex model for a **FieldPlayer** scoring a goal (see also later Figure 6.20), with regard to the football (soccer) domain ontology, which was modeled as a part of SWIntO.

6.2 Semantic Wrapper Agents for QA

In a knowledge-based dialogue system, answering natural language questions by consulting a variety of web pages can be realized by employing wrapper technology for extracting the target information structures and an agent-based system for maintaining semantic access to these information sources and analyzing the results with respect to related user questions. In general, the following tasks are fundamental:

1. Extracting the target information structures from the respective web sites (Section 6.3).
2. Transforming the extracted information structures to be represented as valid ontological instances (Section 6.4).
3. Evaluating the semantic relevance of the extracted instances with respect to a semantic user query (Section 6.5).

In SmartWeb, these tasks have been realized via a wrapper-broker agent infrastructure and integrated into the SmartWeb dialog system. The purpose of such a wrapper-broker system was to add a semantic indexing layer to information structures hidden inside semi-structured (syntactic) web pages. Herewith, access to content can be established at query time resulting in an increase of topicality of the information to be queried and retrieved.

The workflow and deployment of the SWA system into the knowledge-based dialogue system SmartWeb is described in Section 6.6.

The following Section 6.2.1, first, gives a brief overview of related work on semantic wrapper agents, before resolving the above listed subtasks of wrapper generation, semantic transformation and scoring of question-answer pairs.

6.2.1 State of the Art

The agent-oriented software paradigm has been shown to be well-suited for building complex and distributed systems. In the context of the Semantic Web, Tim Berners-Lee's understanding of intelligent agents is that of software components being able "to roam the web of data, allowing machines/software agents to understand the semantics, or meaning of information on the Web" (Berners-Lee et al., 2001).

Information agents that extract and assemble information from the World Wide Web using semantic information had been introduced earlier by Arjona et al. (2002). In their *WebMeaning* approach, semantic wrappers as agents have been described in the context of non-heterogeneous environments that are characterized by many individual ontologies yet missing translation schemes among them, i.e. semantic interoperability. The semantic wrapping system called Thresher (see also Section 6.3.2) used also ontologies for wrapping web content and was part of the Haystack system, a platform with a semantic browser application for end-user authoring. Haystack is focused on the semantic layering aspect in order to give ordinary users the ability to interact with the Semantic Web.

Integrating external data and services into a dialogue system by using an agent-based architecture (Aras et al., 2006) was introduced earlier in the SmartKom⁶⁰ project. The multi-layered agent architecture comprised an interface, - mediation and service layer. While the interface layer was responsible for translating queries and responses, the mediation layer served to link brokers and semantic translators to web repositories and databases.

In contrast to the previous works, the semantic wrapper agents (SWA) approach focused on employing semantic wrappers for the purpose of answering the user's natural language questions at query time, and their deployment in the context of a knowledge-based real-world dialogue system. Moreover, employing an integrated ontology like SWIntO in agents systems as the basis for communication between agents has great benefits for the semantic integration of heterogeneous information sources and their interoperability.

Agent technology and Multi-agent Systems (MAS)

Jennings and Wooldridge (1998) described a variety of application areas for intelligent agents that are “situated in some environment, and that are capable of autonomous actions in this environment in order to achieve their goals”. Besides autonomy and proactivity, agents can be utilized for the decomposition of complex problems/tasks into reasonable sub-problems. Furthermore, in Kirn et al. (2006) intelligent agents have been described in the context of commercial or industrial applications, e.g. autonomous logistic processes (Gehrke et al., 2010). It was also stated by Wooldridge (1999) and Jennings (2001) that agents are well suited for acquiring and processing information in a collaborative way.

⁶⁰<http://www.smartkom.org/>

MAS Platform and Agent Framework

In multiple initiatives intelligent software agents have been developed to mediate between user requests and a “society of service agents”. The idea is that there are various software components that specialize on certain services such as restaurant information, events, cinemas, etc. These service agents can also be specialized for certain regions. So-called broker agents can then in turn, search for agents that provide useful information for a user that suits his location. Some initiatives have started to build such agent networks and to demonstrate the feasibility of this idea (Moreno and Pavón, 2007). The underlying agent middleware has to provide basic services such as look-up and directory services for service identification and a communication language that allows agents to negotiate requests and service offers. But the agent platform also has to implement a communication framework for message exchange between the agents. The way of dealing with communication is the most important difference of agent platforms to other distributed computing environments such as CORBA⁶¹ or JINI⁶². In a multi-agent system, the agents are software-components that share a knowledge-representation, i.e. an ontology, and communicate on a semantically higher level of abstraction than method invocation. This makes agent systems highly interesting as underlying middleware for question-answering systems as well as location-based service, where loosely coupled software components are to be integrated. One standard for such agent platforms is specified by the Foundation for Intelligent Physical Agents⁶³ (FIPA). Multiple FIPA-conformant agent management systems have been implemented, e.g. JADE⁶⁴, FIPA-OS⁶⁵, RAJA (Ding et al., 2001, 2003), etc.

For the integration of various information types, it is necessary to build a common discourse of understanding among the involved agent community. An ontology encodes the world-knowledge from the agent’s perspective in order to reason about the environment the agents are interacting with (Burg, 2002). Here, we are speaking about information integration on a higher semantic level that suits the needs of human users who need flexible and intelligent services. The ontology-driven communication paradigm is not only used in agent systems but also evolves to be the new paradigm for the Web itself, as discussed in earlier sections about the ‘Semantic Web’. Here,

⁶¹<http://www.corba.org>

⁶²<http://river.apache.org>

⁶³<http://www.fipa.org/>

⁶⁴<http://jade.tilab.com/>

⁶⁵<http://sourceforge.net/projects/fipa-os/>

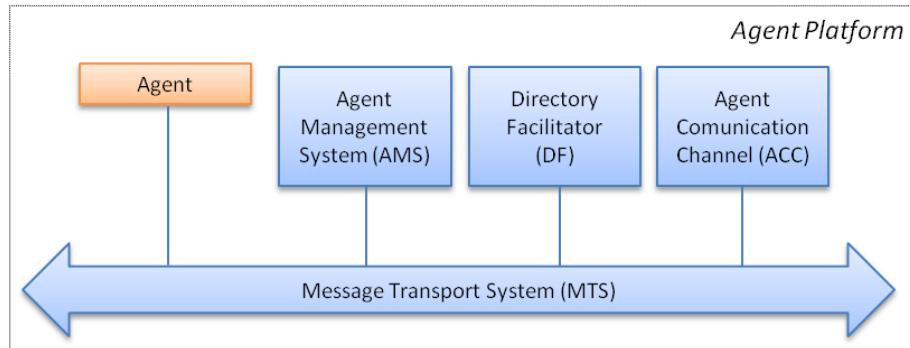


Figure 6.3: FIPA reference model of a MAS (see FIPA-OS described in Poslad et al. (2000)).

ontologies are utilized to describe data and services on the Web (Horrocks, 2008).

The basic agent framework (Figure 6.3) consists of the parts DF, AMS, MTS and ACC. The Directory Facilitator (DF) and Agent Management System (AMS) are specific types of agents, which support agent management, and the Agent Communication Channel (ACC) is a lower-level entity that is part of the Message Transport Service (MTS). The DF provides “yellow pages” services to other agents. The AMS provides platform management functionality, such as monitoring agent life-cycles and ensuring correct behavior of entities within and upon the platform. The ACC supports interoperability both within and across different platforms; therefore, it is viewed as a component of the MTS. The MTS provides a message routing service for agents on a particular platform. Such agents must be reliable, orderly and adhere to the requirements specified in the FIPA-MTS specification. The AMS, MTS and DF form what is called the agent platform (AP). These are mandatory, normative components of the model.

6.3 Semantic Wrapper Generation

Semantic access to web sources in the sense of the Semantic Web, i.e. using formal semantics and ontologies, necessitates models and methods for accessing and transforming existing data on the syntactic Web, i.e. the non-semantic Web. In Section 4.4 a general concept and related tasks for establishing an ontology-based semantic access to semi-structured web sources have been outlined focusing on the problem of “querying the online state of domain-specific web pages”. This Chapter deals with the first two tasks, namely: ontology-based semantic layering and semantic wrapping of information structures inside HTML web documents through visual interaction.

6.3.1 Introduction

As introduced earlier in Section 3.2, semantic wrappers employ a knowledge-based representation (Section 2.2) for encoding the extracted target information structures for a focused domain.

The wrapper creation process generally starts by marking sample information of interest by using concepts and relations from an ontology. While labeling of sample instances of the target information structures for creating wrappers can be performed manually, employing visual interaction is more convenient, less error-prone and less costly.

The idea behind visual wrapper generation is to use a dedicated graphical user interface for generating wrappers for visited web sites utilizing a browser-based tool in a semi-automatic way. The users select sample pieces of information structures they are interested in, e.g. books, weather forecasts or sports results on the visited web page and assign to it semantic tags or labels. From the marked sample instances of the target data structures a semantic wrapper is generated allowing the extraction of all other unlabeled instances of the same type or class from equal or similar pages. The contributions described in this chapter can be listed as follows:

- A semantic tagging metaphor which is efficient to use, is employed in order to allow for building of complex semantic instances to be represented, e.g. in a lightweight-ontology or taxonomy.

The semantic tagging interaction engages the concept of selecting and tagging of single data items (atomic entities) that can be grouped and re-used to form more complex instances beyond simple entities or a list, as known from widely applied extraction approaches with

visual assistance. In addition, linguistic information from the ontology is exploited in order to realize a more user-friendly and easy-to-use interaction for semantic tagging.

- The target information can be extracted efficiently by mapping the marked (hierarchical) sample structure to corresponding extraction rules, depending on the type of instance (atomic, sequence, group, etc.) marked in a recursive manner.
- The scope of information structures can be restricted to content-bearing sections by means of landmarks or semantic region delimiters, allowing noise e.g. menu items, commercials, etc. in the respective web pages to be skipped.

The aim of the visual wrapper generation framework presented herein, is to allow for high accuracy extractions of information structures in order to be represented as ontological instances, which is a main requirement in question answering in a knowledge-based dialogue system. The alternatives, wrapper induction or automatic data extraction are regarded less feasible for covering a variety of semantic structures and structural types, while high accuracy for the generated ontological instances cannot be guaranteed either, as discussed in Section 3.2.3.

In the following section, a brief discussion of alternatives and related work is first given. Then, subsequent sections describe the concept of the visual semantic wrapping approach in Section 6.3.3. Section 6.3.4 gives insight into system design and Section 6.3.5 describes the evaluation of the wrapper generation and extraction components. A description of how the generated semantic wrappers are employed in the knowledge-based dialogue system SmartWeb for the purpose of question answering is presented later in Section 6.6

6.3.2 State of the Art

Forms of visual assistance in wrapper generation is provided by semi-automatic extraction systems such as Thresher (Hogue and Karger, 2005), the commercial system Lixto (Baumgartner et al., 2001) and the W4F system (Sahuguet and Azavant, 1999).

The Thresher system supports using ontologies for layering the extracted information. Its wrapper generation component is based on tree alignment between sub-trees in DOM. The system also allows the creation of wrappers using partial user provided examples to create a generalized wrapper.

Furthermore, the test results of Thresher show that it works well for one class pages, e.g. one movie page, having difficulties with pages where two or more different semantic classes exist. For W4F on the other hand, extraction queries must be formulated via a XML query language, which is a further barrier for generating wrappers by end users.

As elucidated before, complex and dynamic retrieval systems such as dialogue systems benefit from direct forms of semantic interaction via wrappers that access web data at query time, as they are usually limited by turnaround times of a few seconds of dialogue. Hence, a solution based on the offline crawling pattern, where the target information is crawled and the index later updated a few times, cannot be tolerated by the users. Besides that, the lack of semantics in crawler indices is the main reason why complex structured and semantic queries are not possible yet, which is beginning to change with the growth of the Linked Data Web and semantic indices (see Sindice⁶⁶), where information is directly encoded based on formal languages and controlled vocabularies based on RDF.

In contrast, online semantic access to web sources by extracting structured data applying the “on-the-fly access pattern” is reasonable if the mark-up structure, i.e. the schema of the data, does not change very often.

6.3.3 Visual Interaction for Creating Wrappers

The following sections first describe the prerequisites and the workflow for the visual wrapper generation system.

The presented approach is elaborated on in more detail in subsequent sections describing the entire wrapper creation process starting with the ontology-based semantic tagging concept for marking sample information structures in semi-structured web pages.

Prerequisites

A visual semi-automatic wrapper generation system requires methods for intuitively selecting sample instances of the target information structures utilizing a graphical visual user interface, which has to balance the trade-off between ease of use for semantic tagging and expressiveness for the target information structures to be marked. Furthermore, looking at the domain of the regarded content, the usage of an appropriate knowledge representation, i.e. domain ontology, plays a central role. In order to cope

⁶⁶<http://sindice.com/>

with varying layout parts - though generally it can be assumed that similar data structures have been formatted using similar templates - appropriate generalization methods for marked patterns in HTML must be developed.

Besides that, visual interaction for wrapper generation requires transparency, where created wrappers can be (re-)validated against the wrapped data sources, by allowing the user to verify if the extracted data is equivalent to what was wanted. Ontology Editors, such as Protégé⁶⁷ allow for creating concepts, relations and hierarchies together with semantic instances of an ontology, which can be used as reference instances for validating the semantics of extracted information structures.

Finally, wrapping web-based data sources following established Semantic Web standards helps to achieve data and service interoperability. A key advantage of data interoperability based on machine-understandable semantics is the prospect of inference of higher-level knowledge from extracted facts (Hitzler et al., 2009).

Workflow

In the following, the general workflow for creating wrappers through visual interaction and grammar-based generalization is described. Figure 6.4 illustrates the individual steps and related tasks of the wrapper generation process.

The semantic wrapper generation process comprises two main tasks. In the first - *semantic layering* - task, users tag single data entities and group them for marking coherent semantic structures. More complex structures can be created by re-using existing atomic or grouped annotation patterns and combining them on a conceptual level by building for example *is-a* or *part-of* hierarchies or exploiting additional semantic relations. The second - *semantic wrapping* - task comprises the adaptation and generalization of the structure of the underlying HTML fragment for the selected information structures. From the resulting pattern structure the wrapper generation algorithm generates extraction grammar rules through a recursive procedure. The resulting wrapper is applied to the source documents in order to extract data instances that comply with the selected samples. Data records that match the generated grammatical patterns – modeled via attributed grammars – are extracted by the wrapper. The extracted instances can be stored either as XML or RDF format.

The created wrapper is generally capable of extracting similar data

⁶⁷<http://protege.stanford.edu/>

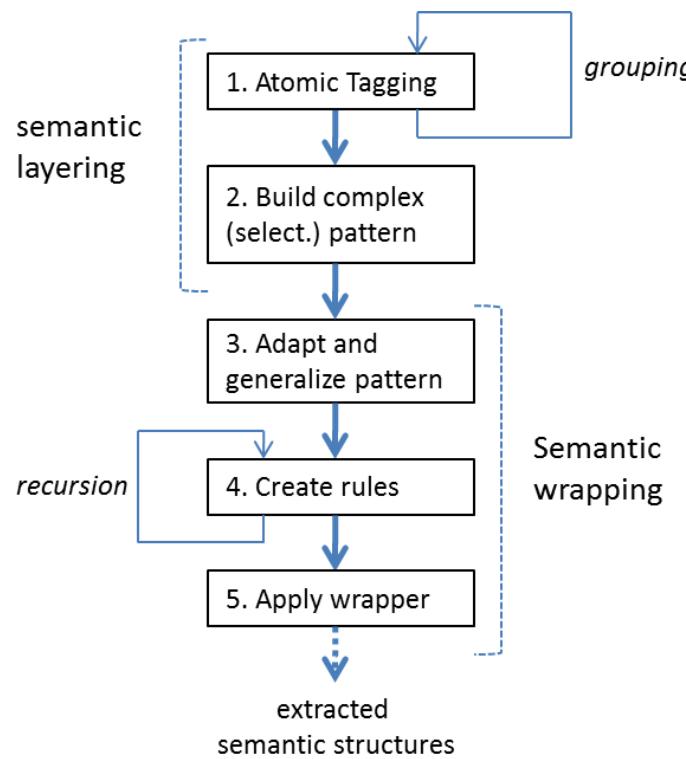


Figure 6.4: Visual wrapping - workflow.

records from web pages that belong to the same page class and are generated by the same script or web service. The similarity metric complies with structural similarity, while the semantic similarity of the target information structures must be verified as being a priori by the user.

Task 1: Lightweight Semantic Tagging

Ontology-based annotation allows the use of expressive models for describing web data by attaching concepts and relations to data structures inside web documents.

Individual data structures can be tagged with concepts, relations between two or more instances can be established forming complex semantic structures for describing web information. In order to reduce complexity in interaction, simpler representations can be used at the surface of the wrapping interface. Simpler terms for tagging the data items can be used by exploiting linguistic knowledge. In SmartWeb, domain ontologies modeled, e.g. SportsEvent were enriched with linguistic information in the form of terms that were directly attached to corresponding concepts (classes) and

properties. The LingInfo approach (Buitelaar et al., 2006) therefore uses instances of the introduced meta-class `LingInfo`. For the wrapper generation system, a word-2-concept lexicon was generated in re-engineering from the underlying ontological models, i.e. the a and t-box of the SmartWeb ontology. The extracted linguistic instances and related terms for given concepts served to provide and suggest tags for marking pieces of data by using the graphical tagging user interface. Figure 6.5 shows the LingInfo model with example domain ontology classes and corresponding linguistic instances.

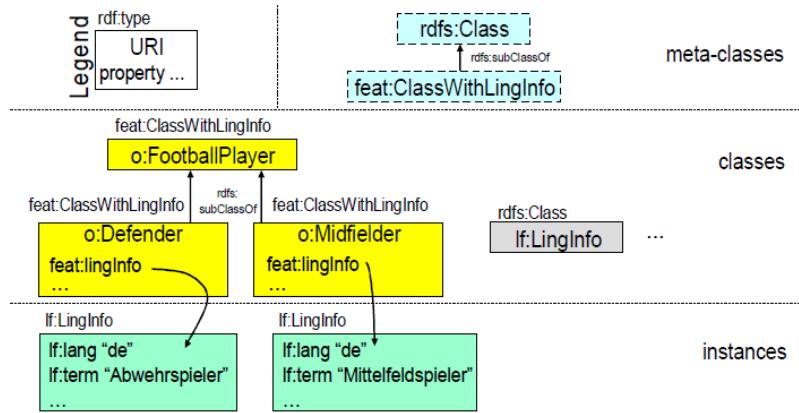


Figure 6.5: LingInfo example for the soccer ontology (Buitelaar et al., 2006).

As illustrated by the workflow shown in Figure 6.4, the tagging process is accomplished by firstly tagging individual data items using semantic predicates and secondly by grouping previously tagged elements using available conceptual terms derived from the used domain ontology. In the following, a tagged single data items is referred to as an “atomic entity”.

The tagging interaction is performed by using a browser-like graphical user interface and a “highlighting and marking” metaphor. Tags can be selected by consulting a pop up menu that comes up when marking an item and clicking on the screen for initiating subsequent interactions for choosing appropriate atomic and conceptual tags. The pattern composition is driven by the visual perception of the template for the given data objects that are shown in the browser. This fact was exploited by several web page segmentation and information extraction algorithms, e.g. VIPS⁶⁸ (Cai et al. (2003)). The user interface of the wrapping tool is described in more detail in Section 6.3.4.

⁶⁸Vision based Page Segmentation

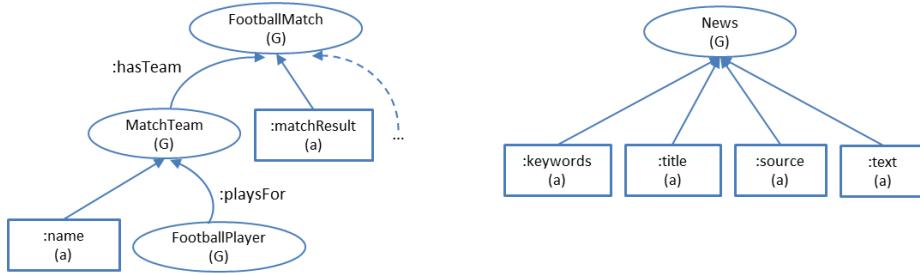


Figure 6.6: Hierarchies in a soccer match (left) and news article (right).

Applying the described tagging process, *subject-property-object* triples known from the RDF triple notation are created implicitly. The creation of conceptual representations is driven by the grouping interaction, either by grouping of atomic tags or binding subgroups to their parent group(s). Hereby, the grouping interaction establishes references between the regarded group concept and its contained elements, i.e. atomic or conceptual entities. A simple preferred strategy (*concept-property-concept*) would be to use an approach to create first atomic entities, group them via a concept, and use dedicated property tags to bind them to higher level concepts, etc. In other words, traversing the tree from the root as the main concept each child node is interpreted alternately as concept or property. For example:

```

FootballMatch ->
    inTournament: ->
        WorldCup ->
            hasName: "WorldCup 2006 Germany"

```

Moreover, by re-using the created atomic and grouping structures, it is possible to create more complex semantic patterns that are used in the following wrapper creation step.

Figure 6.6 (left) shows a simple example structure from the football (soccer) ontology, describing a football match which has a match result and consists of teams, which have names, field players, etc. The corresponding RDF statements of an example football match are given in Table 6.3.

Semantic Pattern Regions and Specific Delimiters

Web pages contain non-relevant data or noise such as advertisements or menu items that do not represent content information. For this reason, it is more efficient to use specific delimiters to skip this kind of information instead of trying to create a wrapper on the basis of such data structures

| Subject | Predicate | Object |
|---------|--------------------------|-------------|
| <_:xid> | <rdf:type> | <rdf:Match> |
| <_:xid> | <sportevent:hasTeam1> | ”Germany” |
| <_:xid> | <sportevent:hasTeam2> | ”Portugal” |
| <_:xid> | <sportevent:matchResult> | ”3:1” |

Table 6.3: RDF statements describing a football match result.

and finally trying to filter out “incorrect instances”. Two types of delimiters can be identified which help to filter out unintentional information in the pre-processing step before creating a wrapper: *specific semantic landmarks* and *semantic region delimiters*.

Marking specific text elements in the context of the tagged entities can help to restrict or skip the source from where semantic structures are extracted. In this case a kind of a *landmark* tag is inserted to indicate that the wrapper-generation algorithm should not generalize the tagged fragment, but instead insert this specific data element as a delimiter into the extraction grammar rules. While landmark delimiters are static elements that focus a certain context, region delimiters consist of two landmarks for restricting the regarded source fragment. As a result, region delimiter tags filter out the HTML code that should be relevant for the extraction process and throw away superfluous parts as described before.

With these few precautions, cases where the wrappers extract similar structural patterns from the same page that do not suffice for the semantics specified during the semantic tagging process can be avoided.

An example for such a semantic region is a list with the same type of items. In the previously shown example, the HTML code that contains the list of players represents the data region for “players”, identified by the specific word “Players” that appears on the web page before the list of the players (see FIFA world championship match report examples⁶⁹). The end of the region could be marked by a landmark at the start of the upward following semantic region.

The problem of filtering non data regions will be alleviated in future by the rising usage of HTML 5, which allows for semantic structuring of the document on the structural level, e.g. using the tags header, nav, article, footer, etc. as well as on textual, e.g. through the mark tag.

⁶⁹<http://www.fifa.com/tournaments/... /matches/match=9994/report.html>

Task 2: Wrapper Creation

The task of wrapper creation comprises the steps for pre-processing, square fragment detection and generalization, and creating the rule tree and the wrapper. The pre-processing of the marked samples has the goal of extracting a sufficient expressive and valid pattern (tagged HTML fragments) for the subsequent generalization, prior to creating the grammar rules for the wrapper.

Task 2.1: Pre-processing

As marked HTML fragments often do not represent valid HTML code, e.g. opening/closing tags might be missing, etc. a cleaning and transformation step is required in order to obtain valid HTML code. Therefore, the tagged web page is first retrieved and transformed from HTML to XHTML. Then, the wrapper creation process starts by first seeking the marked data records or sections (Figure 6.7, left) and correcting their HTML fragment (Figure 6.7, right) by finding the minimum surrounding tag structure of the underlying HTML fragments for the visually selected data items.

```

<title>Wachsender ... </title>
</h2>
<source>DPA</source> -
<p class="cnt">
    Berlin (dpa) - Der Druck ...

```



```

<h2>
    <title>Wachsender ... </title>
</h2>
<source>DPA</source> -
<p class="cnt">
    Berlin (dpa) - Der Druck ...
</p>

```

Figure 6.7: Adaption of selected and tagged HTML-Fragment.

This is needed as the adapted pattern fragment represents a more expressive and discriminative pattern for the subsequent generalization step based on the tagged HTML markup structure. The adaptation can be performed using pushdown automata (Sipser, 1996).

In order to calculate a generalization from the surrounding of the individual tagged data items (atomic entities), a method based on square detection is applied for capturing the general group structure of the individual entities in the HTML code. The respective algorithm is described in detail in Task 2.2.

Task 2.2: Detection of Square-Fragments and Generalization

After pre-processing the tagged data items, a generalization technique based on HTML square detection is applied for each atomic entity⁷⁰ a_1, \dots, a_k .

```

...
<li>
    <h2 class="ynw-sect"><keywords>Ausland</keywords></h2> S1
    <div class="ynw-story clr">
        <h2><title>Wachsender Druck auf Wikileaks in den USA</title></h2> S2
        <source>DPA</source> - vor 1 Min.
        <p class="cnt">
            <text>Berlin (dpa) - Der Druck auf Wikileaks wächst von ... </text> S4
        </p>
        <ul class="moreLnks">
            <li class="bul">
                Für abgetauchten Wikileaks-Gründer Assange wird es eng
            </li>
            <li class="bul">
                Haftbefehl gegen Wikileaks-Gründer bestätigt
            </li>
            <li class="bul more">Ausland: Alle Nachrichten »</li>
        </ul>
    </div>
</li>
...

```

Figure 6.8: Example of square detection (HTML news from Yahoo.)

Starting at linear⁷¹ token index positions $\text{index}(a_i)$ of tagged atomic entities, tag mismatches are detected by expanding the tagging pattern to tag positions before (prefix) and after (suffix) the regarded atomic entities. The mismatch positions determine a range which is called a square S_i around the focused atomic entity a_i .

From the squares of individual atomic entities, a square tree is built by generalizing the regarded XHTML fragments. The purpose of the square detection and generalization algorithm is to obtain expressive HTML pattern fragments for the subsequent extraction rule generation.

The example in Figure 6.8 illustrates the idea of square detection in XHTML documents for the tagged data items a_1 (keywords), a_2 (title), a_3 (source) and a_4 (text) that represent a tagged group of news articles from the Yahoo! News website.

The algorithm finds the squares S1, S2, S3 and S4 that are shown in Table 6.4. For example, the square range of the example square S4 – created around tagged data item $<\text{text}> \dots </\text{text}>$ in Figure 6.8 is given by

⁷⁰Here, for the sake of simplicity, atomic entities are regarded as pure text (PCDATA) elements in HTML, although other leaf elements such as pictures (e.g. IMG) are imaginable.

⁷¹For the linear representation of an HTML document see Section 2.1.2.

the linear tag token index position at the first (11) and the last (14) tag of an XHTML (square) fragment.

In the example, the atomic square fragments are drawn as blue boxes, while the square S3, which is subject to further generalization is depicted as a green box.

| Square | Range: [start, end] | Position of a_i | Pattern |
|------------|---------------------|-------------------|--|
| S1 | [3, 5] | 4 | <h2> a_1 </h2> |
| S2 | [7, 9] | 8 | <h2> a_2 </h2> |
| S3 | [6, 26] | 10 | <div> ... a_3 ... </div> |
| S4 | [11, 14] | 12 | <p> a_4 ... </p> |
| S3' | [6, 26] | 10 | <div> S2 a_3 S4 </div> |

Table 6.4: Detected square pattern candidates and parameters.

As the tagged atomic entity a_3 at index position 10 generates the square S3 (as depicted by the arrows in Figure 6.8) that contains S2 as well as S4, the pattern fragment is generalized by calculating the hull square fragment by removing all irrelevant parts (see `` block in the Figure below) around the embedded square fragments. This is intended, as these parts do not contain any relevant data marked by the user. The resulting **generalized square pattern** is listed as **S3'**.

From the obtained square fragments, a square tree is constructed as the basis for the succeeding rule creation step, which follows a recursive process that is determined by the tagging tree from Task 1 (Figure 6.6, right).

In the following, the details of the square detection algorithm⁷² and the generalization of the tagged XHTML fragments are described.

Algorithm - Square Detection

A square S_i can be defined as a structure having a range $r_i = [start, end]$, a sequence of associated atomic entities $A = a_1, \dots, a_k$ that it contains and a mismatch m_i , which has been unveiled applying the square detection algorithm described next.

A mismatch $m_i = [\text{pre}(s), \text{suf}(s)]$ is a tuple having the mismatch position of the prefix iteration (beforeTag mismatch) as the first element, and the mismatch position of the suffix iteration (afterTag mismatch) as the second element starting at position s. For example Square S3 has a range = [6,26],

⁷² a_1, \dots, a_k : token indices of atomic entities; N: length of document fragment; k,i,j: iterators; c_i : index of closing tag for i-th before tag (prefix) token

associated atomic entities a_2 , a_3 , a_4 and a mismatch $m = [5,27]$ after the generalization step, which is described more detailed later.

Algorithm 1 Square Detection by Tag Mismatch

```

Require: atomicIndicesList=( $a_1, \dots, a_K$ )
    squareList  $\leftarrow ()$ , mismatchList  $\leftarrow ()$ 
for all  $a_k$  in atomicIndicesList do
    set i=1; mismatch=false
    while ((  $(a_k-i) > 0$ ) and (not mismatch)) do
        beforeTag=nextBefore( $a_k$ , i);
        if (isOpenTag(beforeTag)) then
            set j=1;
            while ( $N-(a_k+j)>0$ ) do
                afterTag=nextAfter( $a_k$ , j)
                if (isTagPair(beforeTag,afterTag)) then
                    set s = createSquare();
                    if (squareList.hasNot(s)) then
                        squareList.add(s)
                    end if
                    beforeTag=nextBefore( $a_k$ , i+1)
                    afterTag=nextAfter( $a_k$ , j+1)
                    if (tagMismatch(beforeTag,afterTag)) then
                        m = createMismatch()
                        mismatchList.add(m)
                    end if
                    mergeDuplicates(squareList)
                    exitLoop
                end if
            end while
        end if
    end while
end if
end while
end for
mergeOverlaps(squareList)
return squareList

```

The Square detection algorithm (Algorithm 1) developed herein relies on iterating prefix and suffix tag pattern before and after each individual atomic entity and analyzing surrounding tags whether they have been closed correctly or not. Surrounding tag squares are characterized by a preceding opening tag and the corresponding closing tag after the regarded data item. In principle, four distinct cases can be distinguished:

- A: Atomic elements are surrounded by regular open/close tag pairs, e.g. $\langle a \rangle a_k \langle /a \rangle$ (No mismatch)
- B: No direct closing tag, but an opening tag before, e.g. $\langle p \rangle a_k \langle a \rangle \dots \langle /a \rangle \langle /p \rangle$ (Mismatch)
- C: No direct opening tag, but closing tag after; in this case the corresponding closing tag has to be found, e.g. $\langle p \rangle \dots \langle a \rangle \dots \langle /a \rangle a_k \langle /p \rangle$ (Mismatch)
- D: Neither direct opening tag, nor direct closing tag after, e.g. $\langle h2 \rangle \dots \langle /h2 \rangle a_k \langle p \rangle \dots \langle /p \rangle$ (Mismatch)

Looking at the types of mismatches above, two general kinds of mismatch can be found: a) direct tag mismatch b) indirect tag mismatch at level k around the regarded data item.

Algorithm - Generalizing Square Fragments

Square detection has to consider embedded atomic entities in order to find the correct boundaries for each atomic square by generalizing the surrounding patterns. As shown above, several similar or overlapping squares can be generated by the general iterative square generation algorithm. Therefore, equal and overlapping squares have to be merged appropriately by the algorithm.

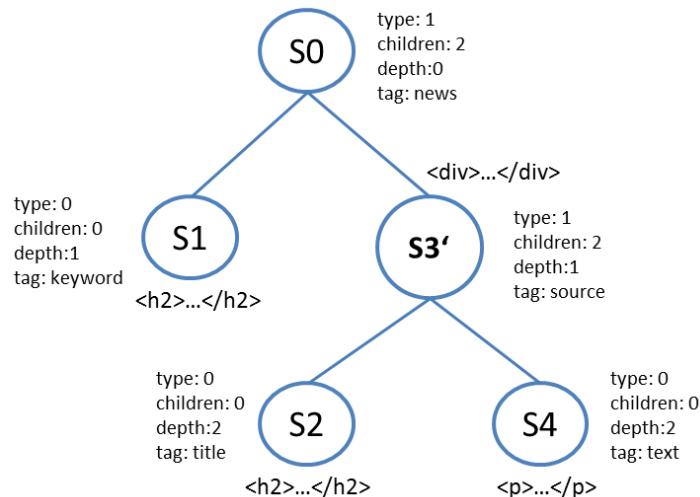


Figure 6.9: Square Tree for the Yahoo news example.

Higher level squares that contain one or more atomic square elements are merged following the minimum hull fragment pattern, which is calculated

by Algorithm 2.

Algorithm 2 Generalizing HTML Fragment Structure of a Square

- 1: **Get** incomplete hull fragment from **head** to **tail**;
 - 2: **Find** all corresponding close tags in square fragment;
 - 3: **Insert** missing close tags for opened tags in minimum square range determined by embedded squares, by iterating backward and analyzing tag pairs;
 - 4: **Create** generalized HTML square fragment;
-

For the HTML fragment shown in Figure 6.8 the following square (S3 complete) is modified in token range from 6 to 26 in order to obtain the minimum hull fragment as a form of a structural generalization. The parameters **head** (position 6, `<div>`) and **tail** (position 14, `</p>`) depict the boundaries of the regarded square with respect to the atomic elements that are contained (here: a_2 , a_3 and a_4).

```

<div>
  <h2>#PCDATA</h2>#PCDATA<p>#PCDATA#PCDATA</p>
  <ul>
    <li>#PCDATA</li>
    <li>#PCDATA</li>
    <li>#PCDATA</li>
  </ul>
</div>

```

The obtained generalization for the example is as follows:

```

<div>
  <h2>#PCDATA</h2>#PCDATA<p>#PCDATA#PCDATA</p>
  .*
</div>

```

Note that the generalization has removed the sub-fragment `...`, as it represents a separate square not directly related to the tagged fragments. The resulting overall square tree with the generalized square fragments for the tagged sample structure of the example is shown in Figure 6.9. It is the starting point for the subsequent wrapper generation based on attributed grammar rules (see definitions in Task 2.3),

Task 2.3: Rule Creation Using Context Free Grammars

The previously created and generalized patterns of selected HTML fragments are encoded by using context free (attributed) grammars that were first formalized by Noam Chomsky in 1956 (Sipser, 1996). In this work, the Jedi java library (Huck et al., 1998) is used to describe the syntactical structure of textual sources such as HTML by defining grammar rules that consist of a name, a return value and a body as shown in Figure 6.10:

```

rule <NAME> : <returnVar> is
    <GRAMMAR_EXPRESSION>          | Pattern identifier
    do
        ...
        <returnVar> = ...;         | Processing logic
    end
end

```

Figure 6.10: Attributed grammar rule definition.

The body of the rule consists of the grammar productions – the *pattern identifier* – for matching and extracting the data, and the processing logic for creating the semantic instances for matched data. The pattern identifier results from the generalization step applied to the detected square fragments.

The wrapper generation process succeeds by generating extraction rules for all selected pattern fragments using grammar productions.

Jedi Attributed Grammar Productions:

In the following, the basics of attributed grammars and rule production as described by Huck et al. (1998) that are used to implement the wrapper rules defined next are presented.

Simple productions can be used to match a single, or a set of characters by using a regular expression like syntax:

| Production | Description |
|------------|-------------------------------|
| . | any character ⁷³ |
| [A – Z] | uppercase letters A to Z |
| ^ [A – Z] | any character but A to Z |
| 'text' | the character sequence 'text' |

More complex rule productions can be composed from simpler ones by

applying the operators shown in Table 6.5 for representing sequences, optional productions, optional repetitions/repetitions, alternatives and grouping:

| Production | Description |
|------------|---------------------------|
| E1 E2 E3 | sequence of E1, E2 and E3 |
| \hat{E} | any but E |
| $E?$ | optional E |
| E^* | optional repetition of E |
| E^+ | repetition of E |
| $E1 E2$ | E1 or E2 |
| (E) | grouping of E |

Table 6.5: Complex grammar productions.

Besides this, rule productions can refer to themselves or other rule definitions using recursion.

Task 2.4 Wrapper Rules Generation

A wrapper consists of a set of rules that are used to parse the target HTML sources or web pages for extracting information structures that comply with the tagged sample(s). In the implemented wrapper generation system, the following four types of rules are distinguished:

- atomic rule (**a**): created from the adapted square pattern for a marked data item i_1 , e.g. the *price* of a data record describing a book
- group rule (**g**): parses subsequent atomic fragments a_1, \dots, a_K by calling the corresponding atomic rules for extracting single data items. The grouping serves the purpose to build simple structures with a list of properties, e.g. $news = (keywords, title, content)$.
- group iterator rule (**g+**): iterates over a sequence of simple group rules (a sequence of data items)
- group joiner rule (**j**): allows distinct groups of items to be parsed independently by parsing the entire HTML source document and joining them with a higher concept.

Example:

In the following, a more complex structured web page (Appendix B.1) serves to illustrate the generation and purpose of the above defined wrapper rules

(a, g, g+, j) implemented as rule productions⁷⁴.

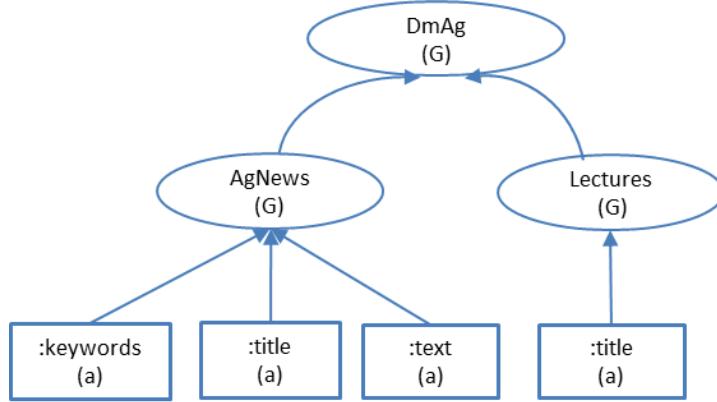


Figure 6.11: Conceptual hierarchies in the dmag-news site.

In the regarded example page, a user aims to extract two conceptual groups (agnews and lectures) from the regarded website of a research group (dmag). The news is tagged with the predicates `keywords`, `title` and `text` for the agnews concept, while the title of a lecture is tagged and grouped to form a list accordingly. The corresponding tagging tree is shown in Figure 6.11.

The atomic rule below (Example R1) named `ont_keywords_0` is used to match atomic data items tagged as “keyword”. The pattern identifier expression derived from the corresponding (generalized) square pattern fragment is defined in the rule header using the attributed grammar productions. As it can be seen by looking at the pattern identifier, the pattern was generalized by replacing the content of the atomic tags with wildcards (“*”). Moreover, in order to avoid ambiguities or strict patterns, the direct context of the relevant data is analyzed, in order to avoid being too specific.

Example R1: Atomic rule for matching keywords.

```

rule ont_keywords_0 : res is
( '^<div' '[']* '^'class="template_list_date"'[']*'>' x_0 = .*'^</div' '[']*'>' )
  do
    res=x_0;
  end
end
  
```

The other atomic entities “title” and “text” are matched via their corre-

⁷⁴In the following examples, the rule naming for the previously defined rule types (see rule headers) follows the conventions listed in the Appendix B.4.

sponding atomic rules. A sequence of data items can be matched by using a group rule (Example R2) that calls the corresponding atomic rules matching the individual data items, e.g. keywords, title, text. The semantics of the target information structure (here: news) is composed in the body of the group rule via XML as the intermediate representation⁷⁵.

Example R2: Group rule for matching a single news structure.

```
rule agnews : res is
(x_0 = ont_keywords_0() .* x_1 = ont_title_1() .* x_2 = ont_text_2() )
do
  res = "<agnews>" +
    "<keywords>" + x_0 + "</keywords>" +
    "<title>" + x_1 + "</title>" +
    "<text>" + x_2 + "</text>" +
    "</agnews>";
end
end
```

For each group rule, a dedicated iterator rule (Example R3) is created by the wrapper generator in order to match a list of (conceptually) grouped data items in a web document.

Example R3: Iterator rule for matching a list of news.

```
rule agnews_iterator : res is
( .+ | list += agnews() )+
do
  listdata = "";
  forall i in list do
    if( i <> null ) then
      listdata = listdata + i;
    else
      end end
  res = listdata;
end
end
```

This is possible, as repeated occurrences of grouped data items follow a similar token pattern in HTML as the tagged examples.

In general, matching flexibility can be guaranteed for certain optional or missing elements exploiting the corresponding operator for optionality (E?) in rule production, while significant structural changes require new tagged samples for creating and validating a functional wrapper.

⁷⁵Other representation languages e.g. RDF can be used accordingly by encoding the corresponding statements in the rule processing body.

Furthermore, complex target semantic structures can consist of one or several other semantic structure(s), which are located in different positions or sections of a web page. For example, a web page could consist of two distinct types, e.g. news and lectures on the same web page. Therefore, in order to indicate to one or several conceptual groups (i.e. semantic structures that may follow different HTML pattern fragments) during the semantic tagging step (see Task 1), groups are bound together with a top level concept. As a consequence of such a higher level grouping above the sequential organization of the patterns to be matched, additional rules beyond the grouping and iterator rule concept are necessary.

Example R4: Main (joiner) rule for matching one or more grouped semantic structures; news and lectures.

```

rule dmag : res is
(^'<html'[^>]*'>' x = .*'^</html'[^>]*'>')
do
    x_0 = agnews_iterator.parse(x);
    x_1 = lectures_iterator.parse(x);
    if ( x_0.length<>0 and x_1.length<>0 ) then
        res = "<dmag>" +
            x_0 +
            x_1 +
        "</dmag>";
    else
    end
end
end

```

For this, joiner rules (Example R4) are utilized to parse the source documents to match the distinct (sub-level) groups. In the rule example below, 2 types of semantic structures are matched by calling two distinct iterators (news, lectures) using a joiner rule for parsing a list of news articles and a list of courses on a sample page.

Recursive Rule Creation

The input for the rule generation process is the pattern tree derived from the corresponding tag tree (Figure 6.11), which is built from the generalized patterns of the selection fragments. The overall rule tree for parsing the semantic structures (as defined by the pattern tree) is built through a recursive procedure.

Note that the pattern identifier grammar expressions (see rule body of the examples) for matching atomic elements or higher level groups have

Algorithm 3 GenericWrapperGenerator

```

Require: deduceRule(Tree t, Depth d):
1: ruleList  $\leftarrow$  ()
2: for all children i of the root of t do
3:   if (i isType atomic) then
4:     skip
5:   else
6:     deduceRule(t.getChild(i), d+1)
7:   end if
8:   createRule(t)
9: end for
10:
11: createRule(Tree i):
12:   if (i isType atomic) then
13:     ruleList.add(createAtomicRule(i))
14:   else if (i isType group) and (nrOfGroups>1) then
15:     ruleList.add(createJoinerRule(i))
16:     if (i is globalRoot) then
17:       ruleList.add(createMainRule(i))
18:       ruleList.add(createMainIteratorRule(i))
19:     end if
20:   else
21:     ruleList.add(createGroupRule(i))
22:     ruleList.add(createGroupIterator(i))
23:   end if
24: return ruleList

```

been calculated by the square detection procedure a priori and assigned to corresponding atomic or higher-level grouping rules by the recursive rule generation procedure appropriately.

The pseudo code of the wrapper generation algorithm is illustrated in more detail in Algorithm 3. The recursion starts at the root of the tree and generates rules for the atomic pattern first, before ascending the tree in the recursion and composing grammar rules for the groups (breadth-first search), group iterators and the joiner rules (see sample tree in Figure 6.12). The atomic rule productions are evaluated at the leaf level, resulting in rules r_1, r_2, \dots, r_k for the atomic elements a_1, a_2, \dots, a_k .

Group rules and their corresponding iterator rules are built from atomic rules when ascending the tree. If one or more groups are bound together with a conceptual tag (as its the case in the example in Figure 6.11), they are bound together with a joiner rule. In case of having reached the root ele-

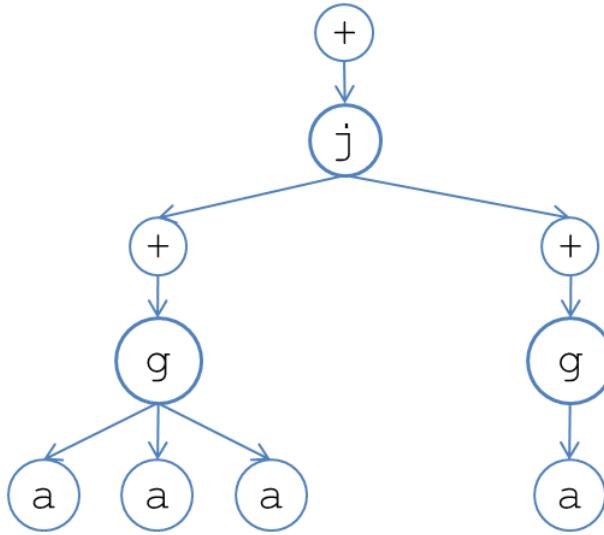


Figure 6.12: Rule Tree Example.

ment, a main (joiner) rule and a corresponding iterator rule is created at the top level. A main rule is equivalent to a joiner rule with the difference, that its pattern identifier has the entire HTML source code as its scope, whereas sub-group joiner rules are restricted according to their square boundaries.

As every rule production knows the semantic entity it represents from the tagging tree, the semantic instance of the regarded class is also produced by this process by inserting the semantic predicates and concepts when building the grammar expression. This evaluation process also includes the insertion of the semantic landmarks that were generated according to the explanations given in Section 6.3.3.

In the following section, the design and implementation of the visual wrapping environment (JEFF⁷⁶) is described.

6.3.4 JEFF - System Design

The implemented wrapper generation system consists of the main components for retrieving the HTML documents, the user interface for tagging sample instances of semantic structures, interactions for creating wrappers, wrapper execution and results management.

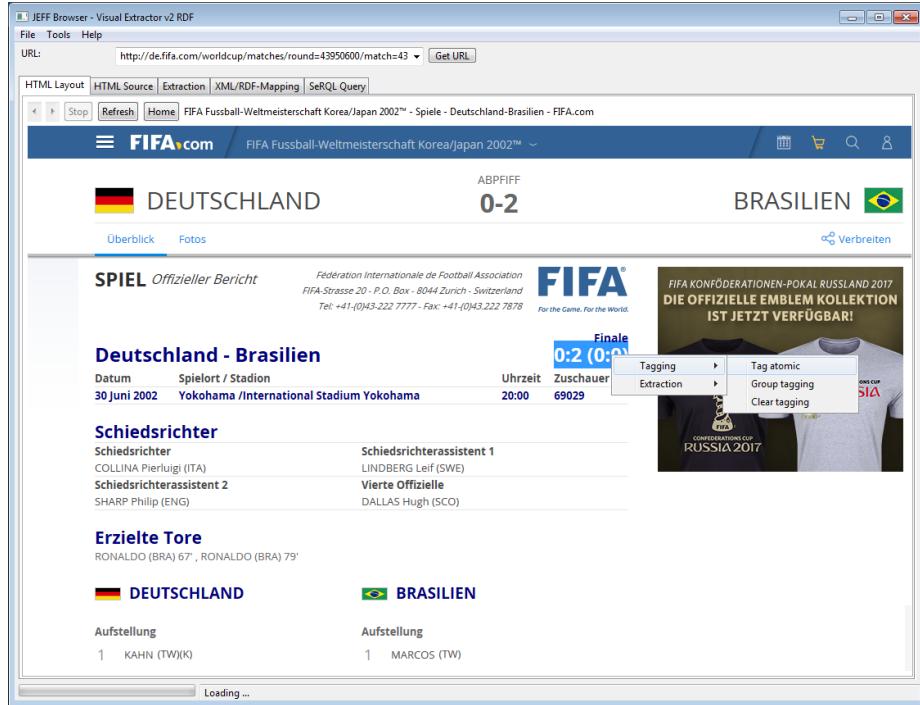


Figure 6.13: JEFF - Semantic wrapper generator UI.

Data Access and Tagging

Figure 6.13 shows the browser-like application based on the eclipse SWT⁷⁷ browser API and additional parsing tools⁷⁸. Target pages are first loaded into the browser.

The underlying retrieval mechanism is responsible for loading and mapping the target web page to an intermediate representation by filtering out non-relevant parts of mark-up and avoiding the loss of descriptive markup for identifying the target information structures.

Bear in mind that web browsers generally modify the original HTML source slightly while adding additional mark-up in order to improve the visual presentation of a particular web page. For this reason, it must be ensured that generated wrappers comply with the original HTML code. Filtered representations have to be created and exploited directly by the wrapper generation framework and its HTML processing components avoiding modifications by an external browser application.

In a second step, the sample web pages need to be annotated and the

⁷⁶Java Extraction Facilitator Framework

⁷⁷eclipse.org/swt

⁷⁸htmlparser.sourceforge.net, jtidy.sourceforge.net

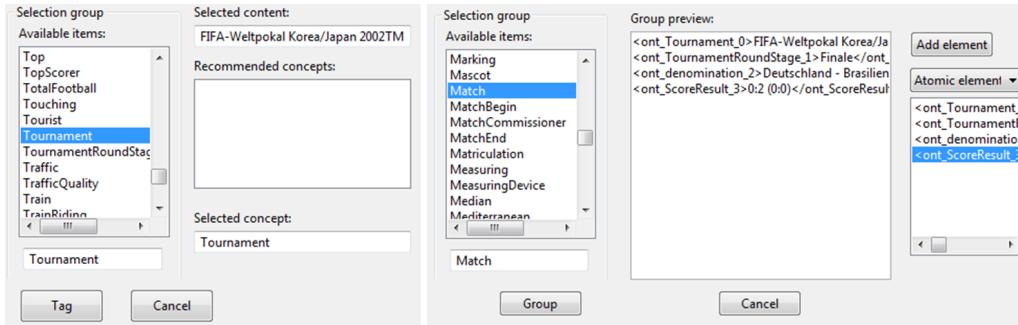


Figure 6.14: Atomic and grouping tagging dialogues.

scope of the extraction determined, i.e. a list of URLs from where data has to be extracted. Therefore, a target web page is loaded into the browser by entering its URL and relevant information structures be tagged consulting a pop-up menu and selecting pieces of information on the page as shown in Figure 6.13.

Over a pop-up menu the dialogues for tagging single data items, grouping of atomic or group entities can be called. In Figure 6.14 (left) the atomic tagging dialogue for selecting semantic labels derived from an ontology or entering tags manually, is shown. The grouping dialogue in Figure 6.14 (right) allows more complex groups to be built from previously created annotations that have been stored before. In general, single data items are first linked forming conceptual groups. In a second step, more complex groups are created by reusing and linking previously created groups.

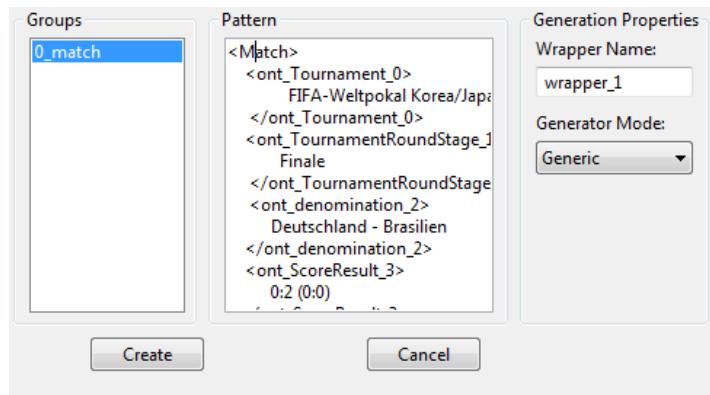


Figure 6.15: JEFF - visual wrapper creation dialogue.

Wrapper Generation

Once a target semantic structure is built, i.e. the target information fragment is annotated, a wrapper can be created by using the “create wrapper” dialogue (Figure 6.15) of the user interface. Hereby, one of the target semantic structures can be selected from previous interactions and assigned to a named wrapper object. The selected wrapper works with the underlying tagging tree of the target semantic structure(s) for mapping the semantics directly to the corresponding extraction grammar rules.

```

<wrapper>
  <srcUri>http://de.news.yahoo.com/</srcUri>
  <ruleUri>data\wrappers\w11\wrapper_11.jedi</ruleUri>
  <title>w11</title>
  <id>11</id>
  ...
</wrapper>
```

Figure 6.16: Wrapper definition in XML.

After instructing the dialogue to create a wrapper, the grammar rules are generated and the target wrapper saved to a directory structure, containing the corresponding XML file for the wrapper definition (Figure 6.16), the extraction grammar script and additional information such as the wrapper id, creation-time, the URI of the HTML source page for this grammar production, the main ontology class, etc. If necessary, the code of the generated grammar rules can be modified and enhanced by consulting the graphical user interface (Appendix B.2) of the wrapper generation framework. Moreover, additional wrapper generation algorithms can be developed or existing ones enhanced and integrated easily into the system in order to cover requirements from future development, as the HTML markup language and its structural and semantic features will evolve and change in future versions.

The developed `GenericWrapperGenerator` (Algorithm 3) based on the previously described square detection and generalization algorithms, allows wrappers for joining and matching different composed structures on the same web page (multi-class wrapping) to be generated. Besides that, the algorithm is able to generate less complex and easy-to-read extraction rules, while at the same time providing more flexibility by generating multi-level grammar rules. Basic rules are created for the atomic entities and combined in higher-level rules to form more complex structures. By considering optional grammars and supporting enumerations, it is possible to deal with

varying layouts and list structures.

Wrapper Execution and Data Storage

Each created wrapper is registered in a wrapper table, which can be queried for finding appropriate extraction wrappers for the focused domains and relevant web sources. Similarity scoring of stored semantic structures will play a central role in “localizing” appropriate wrappers in request-response or question-answering use cases, as will be elucidated later in Section 6.6.

The created wrappers can be looked up over their corresponding semantic class pattern and executed by third party applications via the classes `JediWrapper` and `WrapperExecutor` from the JEFF API in order to extract and process the target data structures. An application of the wrapper technology for answering questions from web data within a dialogue systems is described in Chapter 6 in the SmartWeb project.

6.3.5 Evaluation and Results

For evaluating the wrapper generation system, a collection of documents from two distinct domains, the soccer and news domain, was used. The evaluation followed a 3-step process:

1. Annotating the target semantic structures
2. Generating and executing the wrappers
3. Evaluating the extraction results applying IR/IE evaluation metrics

In contrast to IR, IE systems have to deal with partially correct, incorrect or missing data. Although different additional measures have been introduced for the MUC (Message Understanding Conferences) since late 1980s, precision and recall remained as the primary evaluation metrics (see Section 2.6) in revised form. For evaluating the extraction task described herein, precision (P) is defined as the fraction of the extracted data that is correct and recall (R) as the fraction of the wanted data that was correctly extracted. A balanced metric for evaluating the extraction performance is given by the F_1 measure - the harmonic mean of P and R .

Data Collections

In the context of the SmartWeb project a sample subset from the entire FIFA WorldCup⁷⁹ web corpus (1930-2006) consisting of 708 documents provided in the form of match reports was used. The sample consisted of the reports from the years 1990 to 2006, each containing 52 or 64 match reports depending on the year. In order to evaluate different structural entities wrappers were created on the basis of randomly selected web pages and applied to the rest. The objective of the first experiment was to extract headlines of the match reports, the participating national teams and the goals scored section with high precision. The selected samples were either structured in horizontal or vertical tabular form or as a horizontal list structure.

| i | Source | NrOfDocs | Structure Type |
|---|---------------|----------|-------------------------|
| 1 | www.fifa.com | 296 | div blocks, table, list |
| 2 | dm.tzi.de | 27 | div, paragraph, list |
| 3 | news.yahoo.de | 22 | paragraph, nested div |

Table 6.6: Characteristics of the used datasets.

For the football domain, real-world tests have been accomplished for the semantic class `FootballMatch` and related semantic classes such as `FootballPlayer`, `MatchTeam`, `ScoreGoal`, etc. of the SmartWeb ontology. The overall goal was to form a set of wrappers for extracting the complex target semantic structures of the match reports with maximum precision and to build up a knowledge-based representation of each web page for the purpose of semantic querying. The example in Appendix B.2 shows a sample target structure stored as XML for the semantic class `FootballMatch`, which was generated by the system after wrapper execution.

Additionally, target structures from different news web sites have been extracted and evaluated with generated wrappers for this domain. Table 6.6 lists some characteristics of the used corpuses.

Experiments

Besides the football domain, datasets from the news domain served to evaluate the wrapper generation and the extraction accuracy of the generated wrappers via the precision and recall metrics.

⁷⁹<http://www.fifa.com/worldcup/archive/>

As the focus of the experiments was the extraction performance of the generated wrappers, an intrinsic evaluation was performed. This was necessary, as there was a requirement to ensure reliable extraction, before integrating the information extraction components as part of a higher level information retrieval task in the SmartWeb question-answering system. In summary, the evaluation obeyed the methodology defined by the following questions:

- How many instances of the annotated target sample exist in the regarded documents (N)?
- How many instances have been extracted correctly by the wrapper? (TP)
- How many instances have been falsely extracted? (FP)
- How many instances have not been extracted? (FN)

In general, false positives could be detected easily by applying simple instance filters based on true positives examples, which have been extracted previously.

Another important criterion was the number of wrappers needed to extract the target information structures, which was optimized by applying the rule structure definitions introduced in Task 2.3 of Section 6.3.3. The less wrappers were needed for extraction, the less additional user effort is necessary for the wrapper generation. It must be noted, that wrapper generation corresponds to the generation of a set of rules for extracting a target information structure. Thus, a more direct qualitative metric for measuring complexity is given by the number of produced rules to extract a target information. Nevertheless, from the user's perspective interaction efficiency can be measured by the number of wrappers he/she needs to create.

While one wrapper per structural class was generally sufficient to extract the target structures, significant structural changes could necessitate the generation of additional wrappers in order to deal with partly inaccurate elements in the target semantic structures.

Football (Soccer) Domain Results

The evaluation of the FIFA Soccer WorldCup match reports was conducted on the basis of several interdependent semantic object types which compose

the semantic structure `FootballMatch`. In general, one wrapper per World-Cup year was sufficient.

| Type | Atomic | Group | TP | FP | P | R | F_1 |
|----------------|--------|-------|-------|--------|--------|-----|-------|
| matchInfos | 8 | 1 | 296 | 0 | 1.0 | 1.0 | 1.0 |
| matchOfficials | 4-6 | 1 | 296 | 0 | 1.0 | 1.0 | 1.0 |
| scoredGoal | 2 | 1 | 730 | 0 | 1.0 | 1.0 | 1.0 |
| matchTeam | 1 | 2 | 592 | (1184) | (0.33) | 1.0 | (0.5) |
| footballPlayer | 2 | 1 | 13408 | 0 | 1.0 | 1.0 | 1.0 |
| footballMatch | 0 | 1 | 296 | 0 | (0.87) | 1.0 | 0.9 |
| Σ | 17-19 | 7 | 15618 | (1184) | | | |

Table 6.7: Overall results for the semantic structure `footballMatch` and sub-types in the FIFA corpus 1990-2006.

Moreover, depending on the type of the data source and its contents, each target structure to be extracted could have different structural complexity. In the case of the FIFA football (soccer) match reports, for example, `MatchInfo` consists of two and `ScoredGoal` consisted of one conceptual type with additional attributes, while the `FootballMatch` semantic type comprised a hierarchy of several conceptual entities (groups), which is shown in Appendix B.2.

| 'til year | NrOfDocs | TP | TP_{cum} | FP_{cum} | $N_{cum} = TP_{cum} + FP_{cum}$ |
|-----------|----------|------|------------|------------|---------------------------------|
| 1990 | 52 | 2663 | 2663 | 208 | 2871 |
| -1994 | 104 | 2689 | 5352 | 416 | 5768 |
| -1998 | 168 | 3430 | 8782 | 672 | 9454 |
| -2002 | 232 | 3425 | 12207 | 928 | 13135 |
| -2006 | 296 | 3411 | 15618 | 1184 | 16802 |

Table 6.8: FIFA Wrapper Evaluation: Cumulated number of documents per WorldCup year and corresponding number of true positives (TP).

Table 6.7 shows the results for the distinct conceptual types that were extracted. The first two columns list the number of direct and distinct atomic and group elements each wrapped semantic type comprised. In addition, the number of true positives (TP) and false positives (FP) is listed for each type allowing the calculation of P, R and the F_1 -Score. From the results, it can be seen, that the system was able to extract all instances successfully, while only in one case did the precision of the extracted instances dropped to 33%. Here, 4 additional false instances of the type `MatchTeam`

(FP) were extracted by the system due to structural similarity with other data sections.

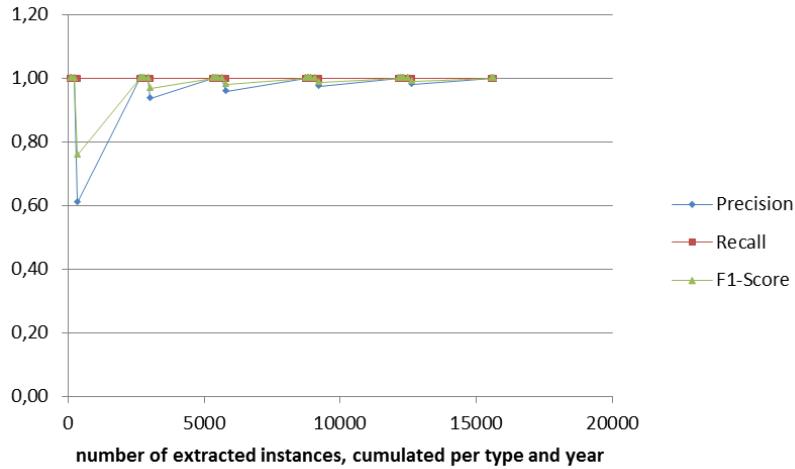


Figure 6.17: Aggregated results of P,R and F-Score for the FIFA corpus.

Using a noise filter in the form of a semantic region delimiter resolved this issue, which required one additional interaction for inserting the region delimiters via the visual annotation interface. Due to high overlap in the similarity of the target structures, the grammar-based wrapper generalization algorithm does not retrieve false negatives or true negatives. The result with false positives before applying the filter is given in round brackets, considering the number of false positives (FP) in the original wrapper. In the case of structures with varying sub-structures, extraction accuracy might be affected by leaving out optional elements not-covered by the provided samples. In such cases, a more complete example can be used by defining the additional atomic entity as an optional element. This case is also covered by the grammar definitions described in Task 2.1 of Section 6.3.3.

Figure 6.17 shows the aggregated mean average results of P, R and F-Score for the FIFA corpus. On the horizontal axis the number of extracted instances of the match report types (`matchInfos`, `matchOfficials`, etc.), which are listed in Table 6.7, are cumulated for each WorldCup year, allowing for looking at the extraction of the specific instances that form the higher level semantic structure `footballMatch`. The extraction behavior for these individual types instances is reflected by the regular pattern in the diagram. The drops indicate the occurrence of false positives (FP) for the instances which have been extracted thus far, which is smoothed with increasing number of correctly extracted instances. As a result, with the increasing number of documents by aggregating, the number of instances per type

and year (see Table 6.8) the F-Score tends to 100%, while only precision is affected by false instances extracted due to partly structural similarity with the target structure.

News Domain Results

The results of the experiments for the two news domain sites are described in Table 6.9 and 6.10, which provided similar good results as the football domain does. Without applying semantic filters, the F-Score is around 93%, while precision may drop in cases where on the source page structures exist with (partly) similar markup structure.

| Dataset | NrOfDocs | NrOfAtomics | NrOfGroups | TP | FP |
|----------|----------|-------------|------------|-----|----|
| dm_exp1 | 21 | 63 | 21 | 33 | 0 |
| dm_exp2 | 4 | 12 | 4 | 80 | 0 |
| dm_exp3 | 1 | 4 | 2 | 13 | 9 |
| dm_exp4 | 1 | 5 | 2 | 13 | 3 |
| ynews1 | 20 | 4 | 1 | 281 | 25 |
| ynews2 | 2 | 13 | 2 | 2 | 0 |
| Σ | 49 | 101 | 32 | 422 | 28 |

Table 6.9: News domain wrapper evaluation.

The first news web site was the site of the Digital Media Research Group at the University of Bremen. The table shows the results of 4 datasets with different page structures. The first two datasets contained only pages with lists of news with the attributes keywords, title and text, while the last two datasets contained pages with two conceptual types: news and additionally a list of courses with a title grouped under lectures.

| Wrapper | Precision | Recall | F_1 -Score |
|-------------|-----------|--------|--------------|
| dm_exp1 | 1.0 | 1.0 | 1.0 |
| dm_exp2 | 1.0 | 1.0 | 1.0 |
| dm_exp3 | 0.59 | 1.0 | 0.74 |
| dm_exp4 | 0.81 | 1.0 | 0.90 |
| ynews1 | 0.92 | 1.0 | 0.96 |
| ynews2 | 1.0 | 1.0 | 1.0 |
| \emptyset | 0.89 | 1.0 | 0.93 |

Table 6.10: News wrapper evaluation results (P/R/F-score).

Overall, one wrapper per experiment was needed to obtain optimal re-

sults. For example, the wrapper for experiment 1 extracted an overall of 63 atomic entities in 21 groups from the overall set of 21 web documents. Figure 6.18 shows the extraction results for the 27 research group news pages, showing a drop of precision at document 25 and 26 due to missing values in the regarded pages. Here, the reason turned out to be a field with information about two aspects - keywords and publication date of the news, stored in the name HTML tag structure, where in the case of the false positives which occurred, only the date had been stored at page creation time.

The Yahoo news web site was analyzed using 2 wrappers for extracting the target news structures. Both page types (datasets ynews1 and ynews2) did not contain further structural levels and were simple list types, i.e. a list of news with the attributes of title, source, date and text and varying layouts.

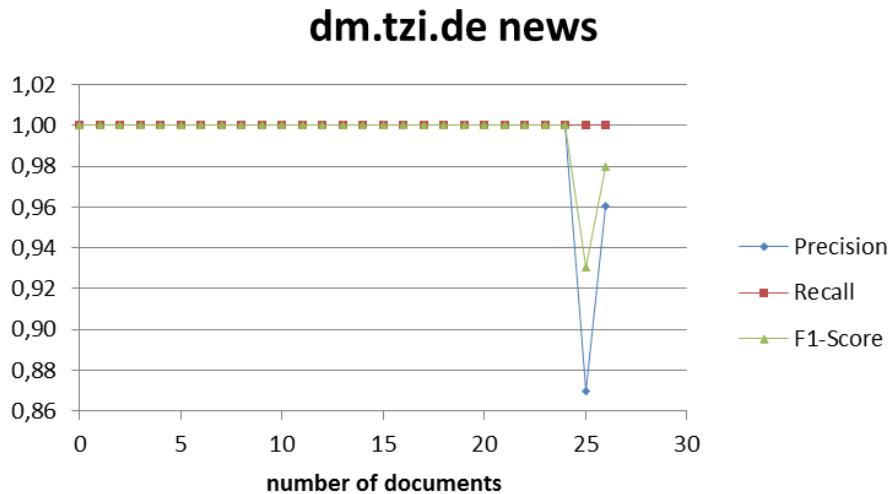


Figure 6.18: Mean results of P,R and F-score for the dm-tzi news pages.

Here, drops in precision (Figure 6.19) occurred due to missing values at the end of the extracted news list and one falsely extracted news instance at the end of the extracted list.

6.3.6 Conclusion

This section described a semi-automatic wrapper generation system for extracting instances of labeled information structures from semi-structured web content. The presented visual selection and generalization algorithm based on visually created semantic pattern allows the generation of wrappers for classes of similarly structured target web pages quickly, in a few

steps. Besides that, restricting the search for semantic instances to a marked region, wrapper rules can be created efficiently. As knowledge-based real-world dialogue systems have high requirements for the quality of the extracted information structures – as it was the case in SmartWeb – wrapper induction or automatic data extraction (Section 3.2) was regarded less feasible in order to cover a variety of semantic structures and structural types, while guaranteeing high accuracy and actuality for the queried, extracted and transformed semantic information structures.

The experiments described in the previous section served to evaluate the wrapper generation approach with several datasets and structural types from two domains. Looking at the structural differences of the wrapped domains, it can be noted that the football dataset contained pages with the most structured elements, having one top concept and several sub-concepts wrapped with the help of additional group and joiner rules. Regarding the hierarchical levels of the wrapped semantic structures, the `footballMatch` structure contains up to 3 or more structural levels, for example: `footballMatch -> matchInfos -> scoredGoal -> scorer`.

The Digital Media news pages have 2 semantic structures (news, lectures) with simple sub-groups as described before. From the semantic complexity, the Yahoo news site represented the most simplistic structure, consisting only of one single semantic news structure.

The results discussed previously show that the presented wrapper gen-

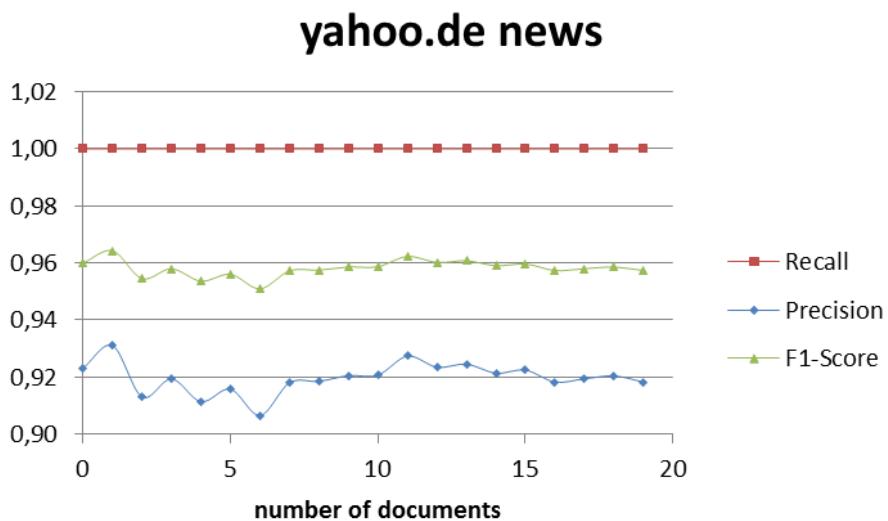


Figure 6.19: Mean results of P,R and F-score for the yahoo news pages.

eration system is capable of handling all the tested structural variations. Drops in extraction accuracy only appear due to structural similarities that need to be covered by additional user interaction for creating modified wrappers for the regarded target structures. It should also be noted, that the chosen web pages contained – besides the target structures – additional elements consisting of menus, commercial sections, links and other non-relevant structures, having partly similar structures as the target semantic structures. Therefore, semantic markup allowing the identification of the target sections for the annotated parts was required, hence additional user interactions could become excessive when regarding more sites and variations. Therefore, additional work for involving user contributions for marking needs to be investigated.

Furthermore, an ontology-based approach allows the expression and modeling of complex knowledge structures to be exploited in knowledge-based dialogue systems. Nevertheless, the high precision approach presented has its drawbacks when working on a site or web-level, where structural changes from one page to another are more drastic and require additional wrappers, hence, additional user interactions for wrapper creation. For the purpose of this work, where the target system was a web-based dialogue system focusing on specific domains, this approach was well suited, as the major requirement of such systems are high precision and recall for the extracted semantic instances.

6.4 Semantic Transformation

In SmartWeb, interoperability between distributed and heterogeneous data sources and the dialogue system is achieved by means of semantic integration based on formal semantics. For the semantic wrapper agents system, to be part of the SmartWeb ecosystem, the agent system must be able to communicate with the natural language understanding components in a common target knowledge representation, which is achieved by transforming extracted data structures to valid ontological instances of the SmartWeb integrated ontology SWIntO. Without semantic consistency of the extracted semantic structures, subsequent steps such as the scoring and ranking of answer candidates cannot be performed.

Therefore, the aim of the framework presented in this section is to translate structured data extracted, e.g. as XML (input structure) into RDF-S/OWL instances of the SmartWeb ontology (target structure) using a rule-based approach. In contrast to existing approaches the framework employs a semi-automatic adaptive translation method triggered by the input structure exploiting the ontology's hierarchy and axioms. Besides that, it allows for manual modifications of the transformation through a flexible and efficient rule set for complex cases.

In the following, after describing some related work briefly, the input/output data representation models and the workflow of the semantic transformation engine from the input representation model to the target representation using a set of rule definitions is explained.

6.4.1 State of the Art

Matching techniques coming from database or XML schema matching research use lexical and structural components to find correspondences between two input schemas or XML structures. In contrast to that, ontology-based techniques exploit semantic relationships, i.e. subclass-of or part-of relations, assignment of properties to classes, domain and range definitions of class properties. Furthermore, ontologies have more specified constraints than schemas. While schemas do not provide explicit semantics for their data, ontologies obey formal semantics.

The TARTAR system by Pivk et al. (2006) transforms arbitrary HTML tables into frame-structures, which can be easily transformed to RDF. TARTAR integrates the wrapping and analyzing of tabular data with RDF transformation. The transformation process is similar to the approach described

herein. In much the same way the SPIN parser (Engel, 2006) implements a working-memory-based production system for parsing natural language text directly to frames or RDF instances. Here the source is natural language sentences where terminal symbols (words) are replaced by ontology concepts using a production system. While the approaches described above have mastered bridging the gap from XML to RDF by using a complex overall ontology, the work presented in Cruz et al. (2004) generates a global ontology from individual ontologies that are created from single XML documents through merging.

The well-known TRAINS system (Allen et al., 1996) uses different representations in its basic components for parsing, action planning and generation which was reported by Ferguson et al. (1996). In fact, even central concepts such as *city* are modeled differently by different components.

A solution for the issues described above was investigated by Porzel et al. (2003) and Gurevych et al. (2003), who showed the benefits of an approach that employs a single target knowledge representation. Their work has been deployed successfully within the SmartKom (Wahlster, 2006) multimodal dialogue system. The benefits of employing one single target ontology like the SWIntO ontology are exploited by the semantic transformation approach developed for the SWA subsystem in SmartWeb in order to seamlessly process and integrate the semantic structures extracted from various web sources.

6.4.2 Approach

Knowledge processing and semantic interaction in the SmartWeb system is based on complex ontological representations of data and the information exchanged via the SWIntO ontology. In the SWA system described in Section 6.2, semi-structured information is extracted from a variety of web sources employing wrappers. In the course of the wrapper generation process the extracted information structures are encoded via XML-based representation or RDF. This representation can also be regarded as a form of instance representation in a lightweight ontology, where simple concepts and their relations derived from the SmartWeb domain ontology for the Soccer World Cup have been used to tag the target information structures in a web site. From the perspective of the SmartWeb system such lightweight representations are less expressive and do not represent valid ontological SWIntO instances, which are more complex and have rich semantic relations.

As a fully automatic semantic translation of such web data into an ontological representation in a complex ontology like SWIntO is still a challenging and serious task, a rule-based method for mapping between structured specifications of individual data instances, e.g. as XML, and instances of a target ontology in RDF-S/OWL was developed.

The basic tasks implemented by rules consist of mapping XML tags onto ontological concepts and associated properties, generation of additional RDF instances to conform the axioms of the target ontology, merging of instances and their validation. The semantic transformation rules are exploited by the individual wrappers for encoding the extracted instances as valid SWIntO instances.

6.4.3 Data Representation: Input/Output

The following sections describe the input model (XML) and the target model (RDF-S) for the semantic transformation.

Input Model

The semi-automatic creation of RDF instances is heavily influenced by the inherent structure of the input, e.g tree or other structured specification.

```

<footballMatch>
  <name>Deutschland–Saudi Arabien</name>
  <inTournament>
    <worldCup>
      <heldIn>
        <country>Japan</country>
      </heldIn>
    </worldCup>
  </inTournament>
  <matchEvents>
    <goals>
      <scoreGoal>
        <committedBy>Ballack</committedBy>
        <atMinute>40</atMinute>
      </scoreGoal>
    </goals>
  </matchEvents>
</footballMatch>
```

Figure 6.20: “Ballack scored a goal in a WorldCup soccer match ... ” as XML representation generated by a Wrapper.

For tree or graph-based structures that can be expressed via *subject-predicate-object* relations, a minimum manual intervention is needed to

transform the data to RDF. For that reason, a semantic tagging strategy for promoting the creation of simple concept-property-concept structures over repeated atomic tagging and grouping was directly integrated into the user interface of the wrapper generation system described in Section 6.3.

In this way, it is possible to create XML trees following a structure that can be translated efficiently. Expressions beyond this concept need additional rule definitions by modeling transformation rules to cover the exceptional cases.

In the following, the transformation process is explained with the XML example input data structure shown in Figure 6.20. This sample XML structure was created by the wrapper generation and extraction process. XML was used as an intermediate representation for the wrapper generation and extraction, as it is easy to process and allows an efficient way to encode the wrapped information structures via a taxonomy-like structure. Here, concepts as well as semantic predicates were modeled by means of appropriate conceptual mark-up in XML.

Target Model

The example in Figure 6.21 shows the transformed ontological representation of the input XML structure presented in Figure 6.20. The shown instance encodes the fact of “a football player called Michael Ballack, scored a goal in a football match between Germany and Saudi Arabia at the World-Cup in Japan, in the 40th minute. It is clear that an ontological representation for this example is far more complex than just mapping the XML tag names to ontological concepts and properties. For instance, an axiomatic naming of a person in SWIntO (here: football player) who is a man, is encoded via a cascade of semantic relations⁸⁰ :impersonatedBy -> :Man > :HAS-DENOMINATION -> :denomination > :NAME.

The following sections describe the rule-based semantic transformation approach and the available rule types.

6.4.4 Transformation Engine *2RDF

In the described application scenario wrapper agents extract tables or other semi-structured data from the web as described in the work of Porzel et al. (2006). The extracted data is processed further by the *2RDF transformation engine resulting in semantic instances in RDF-S/OWL and stored in

⁸⁰For the sake of simplicity, the abbreviations of the namespaces of the involved ontologies are ignored.

a persistent semantic repository such as the Sesame engine or sent directly to other reasoning components.

1. Tree Processing

All processing steps depend on a tree or graph structure, e.g. encoded as XML which has to be implemented by the source of structured data. The transformation engine then calls this interface and retrieves all nodes of this tree. This way adapting to another source can be regarded as implementing methods for tree traversal. Nodes of height $2n, n \geq 0$ are mapped to con-

```

<sevent:FootballMatch rdf:about ="2000">
  <sevent:inTournament>
    <sevent:FIFAWorldCup rdf:about="">
      <sevent:heldIn>
        <sdolce:Country rdf:about="">
          <sdolce:HAS-DENOMINATION>
            <ssumo:country-denomination rdf:about="">
              <sdolce:NAME>Japan</j.0:NAME>
            </ssumo:country-denomination>
          </sdolce:HAS-DENOMINATION>
        </sdolce:Country>
      </<sevent:heldIn>
    </<sevent:FIFAWorldCup>
  </<sevent:inTournament>
  <sdolce:HAS-DENOMINATION>
    <sdolce:denomination rdf:about="">
      <sdolce:NAME>Deutschland-Saudi Arabien</j.0:NAME>
    </sdolce:denomination>
  </sdolce:HAS-DENOMINATION>
  <sevent:matchEvents>
    <sevent:ScoreGoal rdf:about="1912">
      <sevent:committedBy>
        <sevent:impersonatedBy>
          <ssumo:Man rdf:about="2006">
            <sdolce:HAS-DENOMINATION>
              <sdolce:denomination rdf:about="2007">
                <sdolce:NAME>Michael Ballack</sdolce:NAME>
              </sdolce:natural-person-denomination>
            </sdolce:HAS-DENOMINATION>
          </ssumo:Man>
        </sevent:impersonatedBy>
      </sevent:committedBy>
    <sdolce:HAPPENS-AT>
      <sevent:MatchTimePointRelative rdf:about="1916">
        <sdolce:OFFSET>40</sdolce:OFFSET>
      </sevent:MatchTimePointRelative>
    </sdolce:HAPPENS-AT>
  </sevent:ScoreGoal>
  <sevent:matchEvents>
</sevent:FootballMatch>
```

Figure 6.21: “Ballack scored a goal in a football match” as target instances of the SWIntO ontology.

cepts while node of height $2n + 1$ are interpreted as properties. Hence, the nodes are interpreted alternately as concept or property by default (*concept-property-concept*), while this order can be modified by one of the defined rules.

The transformation process is controlled by a set of rules defined in a rules file (a sample definition for the FIFA 2006 corpus is given in Appendix B.6), which are applied for each input data source and used by the associated wrappers for transforming extracted information structures.

2. Rule Syntax

Each rule is identified by a naming keyword followed by predefined parameters e.g. tags or concepts, it will operate on. The general syntax is defined as follows:

```
ruleName parameter1 parameter2 ...
```

A rule for mapping a tag “<Match>” onto an ontology concept “FootballMatch” is written simply as: `concept Match s:FootballMatch`. If a tag has the same name as its corresponding concept in the ontology, this rule would be obsolete. The same is true for matching properties, while the usage of equally named properties may vary for different conceptual contexts, which can be covered by additional transformation rules. Table 6.11 shows all available rule types with an example transformation.

| Rule Syntax | Example Transformations |
|---|---|
| <code>concept <t> <c></code> | <code>concept Match s:FootballMatch</code> |
| <code>property <t> <p></code> | <code>property scoredBy s:committedBy</code> |
| <code>cmap <t> <c> <p></code> | <code>cmap name s:FieldPlayer s:hasUpperRole</code> |
| <code>insert <c> [<p>] <c> [<p>]</code> | see example in Table 6.22 and 6.23 |
| <code>list <t></code> | list goals (see Appendix B.6) |
| <code>maplist <t> <p></code> | <code>maplist matchEvents s:MatchEvents</code> |
| <code>merge <c> <p1> <p2></code> | <code>merge ss:Man sd:has-denomination</code> |
| <code>rename <c> <c> <p> <p></code> | <code>rename s:WorldCup[sd:YEAR]</code> <code>sd:HAPPENS-AT</code> |

Table 6.11: Example transformation rules.

⁸⁰Note, that in the examples ‘s’ is an abbreviation for the namespace of the SportEventOntology, while ‘ss’ stands for the SmartSumo and ‘sd’ for SmartDolce ontologies.

The transformation process involves several steps, while the rule set is associated with one of these steps. All rules at the same level are executed in the order in which they are defined within the rules file. In the following, the individual steps are described and some rule examples given.

3. Mapping Rules from Tags to Concepts or Properties

The transformation process starts by defining simple mappings to create RDF instances for each node. If the tag-set is identical to the ontology's naming of concepts and properties no additional mappings are needed. The mapping-engine tries to derive the appropriate concepts and properties from the ontology. This works if the wrapper has been designed with the ontology in mind, i.e. if the terms for semantic annotation rely upon terms derived from a target ontology as described earlier in Section 6.3.3 (Tasks 1).

The rule syntax shown in the first row of Table 6.11 allow to map a tag (*t*) manually onto an ontology concept (*c*) or property (*p**) via a dedicated mapping rule. In some cases identical tags refer to different properties depending on the parent node. In order to handle this, conditional mapping rules can be defined via a `cmap` rule. In other words, properties are mapped depending on the domain concept, i.e. the last mapped upper concept.

4. Transforming List Structures

Lists of instances can be converted if declared as being a list. Lists may violate the hierarchical concept/property order and have to be declared using a list rule. The parent node of a regular list should be a property. Note, that the regarded tag is not mapped to any concept or property, but is only used to describe the list. Therefore, the rule *list <tag>* just marks such a list.

<tag> indicates that all nodes beneath this list-node will be interpreted as concepts which are fillers for the property defined by the parent node of the list tag. Additionally, there is a *maplist* rule, which has a concept tag as parent node. Here, the property has to be provided with the rule, the list node will be mapped to this property, i.e. for each item in the list a property of the specified type is inserted. The parent node should be mapped to a concept.

5. Creating Instances with Additional Elements

Most ontologies model the world in a more fine grained manner than it is described on web pages. Ontology axioms, for example, may require a distinction between roles and types, endurants or perdurants. In order to cover such cases, expansion/insert rules can be declared. These are production rules, which anchor on concept or properties. When these anchors occur during transformation, additional instances are inserted at place.

```
<goal>
  <committedBy>Ballack</committedBy>
  <score>1:0</score>
</goal>
```

Figure 6.22: A goal scoring example

The simple structure in Figure 6.22 comes from a HTML table listing goals. The goal in this case is scored by a **FieldPlayer** that has an upper role named **FootBallPlayer**, which is impersonated by a **Man**, which has a **denomination** which has the value “Ballack”. This means, that this XML snippet maps to the three nested RDF instances generated by the three insert rules shown in Figure 6.23.

```
insert s:FieldPlayer[s:hasUpperRole] s:FootballPlayer[s:impersonatedBy]
insert s:FootballPlayer[s:impersonatedBy] ss:Man[sd:has-...-denom.]
insert ss:Man[sd:HAS-DENOM.] sd:has-...-denom.[sd:NAME]
```

Figure 6.23: The player encoding of the goal scoring example.

6. Merging of different aspects in the input structure

Depending on the web site which has been wrapped, information about a single instance may occur in different places on the site. A “substitution” for example, can be split into an “in” and “out” part, while the ontology models this as one event. The merging or *smushing* matches the “time” of this event and merges all substitutions taking place at the same “time”. This also applies for all **denomination** and **Man** instances which were created during the expansion phase. While OWL has *inverse functional properties* which imply uniqueness of instances and can be used for smushing, RDF doesn’t support such properties. Therefore, the properties used for merging have to be stated within the rules. Each instance of a type to be merged

has to be compared with all other instances of this type. The corresponding rules will merge football players after having merged “names” and “man” instances. Additionally, all substitutions happening at the same time are considered being the same. Since many instances depend on each other, the order in which the smushing occurs, is important. As described above, a football player is modeled as a `Role a Man`, he himself having a name. Therefore, all `denomination` instances first have to be merged, after this the `Man` and `FootBallPlayer` and lastly the `FieldPlayer` instances.

This, however, may work only in a local context, for example, substitutions are only unique during a single football match. For this reason, the transformation has to be done for each information unit, i.e. each source separately. The merging step has to be iterated until no more instances can be unified.

7. Instance Validation

A validator checks whether the resulting instances comply with the given target RDF-S ontology or not. In worst case, the instance transformation is interrupted and an error message thrown. At this point, additional analysis or manual intervention may be required to resolve the occurring issues. It may also be needed to update the corresponding wrappers via the wrapper generation user interface described in Section 6.3. Sample reference instances created manually using an ontology editor can be used for validation as well.

6.4.5 Conclusion

The last sections showed that for embedding information extraction into a complex semantically grounded dialogue system, such as SmartWeb, there is a need for a flexible bridge between these worlds. It has been shown that this bridge can be built with a rule-based semantic transformation engine following a minimalistic approach, i.e. the less rules that are needed to encode the transformation of the input semantic structures, the more conveniently and efficiently the transformation process and future modifications can be performed. The transformation rules for wrapped web information sources are exploited by the wrapper agents in order to convert the extracted information structures to their corresponding ontological representation in RDF-S on-the-fly. Furthermore, the rule-based system reduces manual effort by leveraging the inherent structure of the input model, which is created by the visual wrapper generation system during semantic label-

ing applying the concept-property-concept rule as a convention for building complex structures.

In order to further automate the described mapping and transformation process, future work could integrate approaches from the field of automatic ontology matching and ontology learning. Indeed, even without adhering an underlying formalized ontology, structured XML data defines an *implicit* ontology (Klein et al., 2000). This implicit ontology could be extracted from the instance data by simply collecting all occurring properties for a given concept. This inferred ontology could then be mapped to the target ontology by using approaches such as proposed by Kalfoglou and Schorlemmer (2003) or Ehrig and Staab (2004).

Future work might also focus on automating the expansion process. The expansion rules might, for example, be automatically created from the ontology axioms. Another promising approach would be to find the proper chain of instances as done by the OntoScore (Gurevych et al., 2003) system. One question here is how to handle ambiguities if more than one path between e.g. a **FieldPlayer** and his/her **denomination** is possible and the shortest path does not reflect the semantics unambiguously. In this case either a special selection rule is needed, or some other automatic selection function to ensure that the transformation bridge does not create incorrect instances.

After the instance extraction and transformation into the SWintO ontology, the extracted instances need to be scored and ranked with an appropriate method for selecting the best answer for a given semantic query. As it will be shown in the following section, wrapper selection and the extraction process can also benefit from a semantic analysis and scoring of previously extracted answer instances.

6.5 Semantic Scoring of (Q,A)-Pairs

The user-system interaction in a knowledge-based dialogue system is characterized by a sequence of single question-answer dialogue turns. As elucidated before, semantic queries are ontological representations of the user's natural language questions. Encoding the answer candidates as ontological instances by the retrieval subsystems via the same knowledge representation language allows the calculation of semantic similarity scores between pairs of questions and answers.

In the SWA subsystem of SmartWeb, registered wrapper agents are employed for extracting one or more answer instances from the target web information sources. In order to find the best answer from the extracted candidates, the answer instances must be scored and ranked. Instance scoring is also applied for selecting one or several wrappers that extract the semantically related candidates. Having prior knowledge about the type of the instances to be extracted from a particular web site and their semantic context, on the one hand, helps to reduce the amount of unnecessary extractions, and on the other hand, increases the confidence for returning a fresh answer⁸¹.

Therefore, the semantic relatedness or similarity scores of question-answer instance pairs are stored in a semantic similarity scoring matrix (Table 6.12) and looked-up by every time a query arrives from the dialogue system. For each query q_k the matrix saves its semantic class c_k and the question focus f_k , together with a score s_{ik} for the best fitting wrapper(s).

| query | w_1 | w_2 | ... | w_i | ... | w_N |
|------------------|----------|----------|-----|----------|-----|----------|
| $q_k : c_k, f_k$ | s_{1k} | s_{2k} | ... | s_{ik} | ... | s_{Nk} |

Table 6.12: Semantic similarity scoring matrix.

The wrappers are selected by the broker agent, which administers registered wrappers in a wrapper pool, as explained in Section 6.3.4. As a result, an algorithm for evaluating ontological instances extracted by the wrapper agents with respect to a given semantic query aims at resolving the following task:

- Find the best possible answer instance to a given semantic query from

⁸¹As discussed previously, utilizing semantic wrappers allows the posing of semantic queries to the online state of the "wrapped" web pages.

the list of answer candidates matching a general semantic class with the specified target or focus, e.g. `WorldCup` as a semantic class and `sevent:winner` property as the focus of the query, which is defined as a `NationalTeam` ontology instance.

6.5.1 State of the Art

According to Narayanan and Harabagiu (2004a) answers from large text collections can be extracted by applying either classification based on the answer type, using question keywords or patterns associated with the question or ranking the candidate answers in order to find the appropriate one(s).

Answer selection - which is closely related to scoring ontology instances as answers - is regarded an important subtask in question answering. Answer selection from candidate answers extracted from natural language text was investigated by Narayanan and Harabagiu (2004b) as well. In the context of this thesis, answer selection is defined as the task of choosing the most likely answer from a pool of previously gathered answer candidates for a given question (Xu et al., 2003). Furthermore, Sinha and Narayanan (2005) believe that the use of relations allows irrelevant answer candidates to be efficiently eliminated.

Semantic Coherence Scoring

Ontology-based concept-scoring algorithms are applied to disambiguate speech recognition hypotheses or in word sense disambiguation. In general, speech recognition systems produce n-best lists for possible utterances from the user (speech recognition hypotheses). The OntoScore algorithm can be used to calculate the semantic coherence of speech utterances (Gurevych et al., 2003). In case of pairs of questions and answers, a high score would indicate a good match.

In SmartWeb, an enhanced version of this algorithm called *AnswerScore* was developed in order to calculate the semantic relatedness between two concept sets, i.e. the average distance of concepts over their relations. For example “football player”, “ball” and “goal” are semantically related. A semantic similarity measure describes similarity on a taxonomic level, i.e. “restaurant” is similar to “pub”. In *AnswerScore*, concepts that are “semantically similar” are filtered out as they repeat the question and distort the score. Further comparisons at statement level improved the scoring results. In contrast to *OntoScore*, *AnswerScore* uses different weights for the

semantic relations used and ranks paths on the same level higher than paths across levels or even domains.

While the aforementioned algorithms are well suited for ontologies that are mostly homogeneous and provide concepts widely at the same level of granularity, despite the enhancements in AnswerScore, it turned out that the methods do not produce good results for the SWIntO ontology, where the integrated domain ontologies have grown historically, i.e. the most common concepts are built in a very detailed way, while others have less properties, etc.

Therefore, a new algorithm for scoring ontological instances based on the idea of semantic density has been developed and evaluated in the agent-based semantic question answering system in SmartWeb employing wrapper technology. The idea of semantic density is based on overlaps of properties and instances in the question and the answer instance. The compared instances are regarded as being more semantically related or similar the more overlap between properties and instances exist.

Instance overlap is calculated by looking at the corresponding [subject-property-object] triple of the answer predicates and validating the content of the object part, e.g. concept, literal, etc.

In the work of Pretzsch (2006), he analyzed the drawbacks of OntoScore for the SWIntO ontology, and investigated the semantic density idea as a reference implementation for scoring pairs of ontological instances. The semantic density approach was implemented and evaluated in the SmartWeb dialogue system as part of the SWA subsystem.

6.5.2 Approach

The idea behind the semantic density approach is that the best suited answer for a semantic user query can be identified by looking at matching properties in the answer candidates and additionally comparing the semantic content of an answer instance, which is stored in the object part of an RDF triple⁸².

The example in Table 6.13 shows that all properties in the question match in the answer, while the focus of the question (see also Figure 6.2) is filled with the wanted information.

In the following, for a list of returned answer candidates $A = a_1, a_2, \dots, a_k$ from the wrapper agents, each a_i is compared with the query q .

⁸²[subject, predicate, object], e.g. in RDF N-triple notation $x2 \text{ sevent:committedBy } x1$.

| Type | Example Instance |
|----------|--|
| question | x1 rdf:type svent:WorldCup x1 sevent:heldOn "1990" x1 sevent:winner x2 x2 rdf:type svent:DivisionNationalTeam |
| answer | x1 rdf:type svent:WorldCup x1 sevent:heldOn "1990" x1 sevent:winner x2 x2 rdf:type svent:DivisionNationalTeam x2 sdolce:HAS-DENOMINATION x3 x3 sdolce:denomination x4 x4 sdolce:NAME "Germany" |

Table 6.13: A sample question-answer pair as N-triples.

Semantic Density Algorithm

Similarly to the OntoScore approach, the ontological domain model is converted into a directed graph, whereas nodes represent concepts and edges relations, i.e. properties.

Be $Q = q_1, q_2, \dots, q_{N1}$ a list of questions collected in the course of user dialogues in SmartWeb and $A = a_1, a_2, \dots, a_{N2}$ the list of the answer candidates extracted by the wrapper agents, with q_i being questions and a_j answers.

1. Property Score Calculation - Comparing Properties

For calculating the amount of overlapping properties in each question-answer pair (q_i, a_j) , how many properties the answer shares with the question is analyzed.

The input to the *property slot filler* algorithm is a list of question properties $P_i^q = \{p_{i1}^q, p_{i2}^q, \dots, p_{i,K1}^q\}$ and a list of answer properties $P_j^a = \{p_{j1}^a, p_{j2}^a, \dots, p_{j,K2}^a\}$. In other words, the properties of a question q_i are collected and build a semantic property density D_{q_i} comprising the set of question properties. The size of such a property density is defined as

$$d_{p_{q_i}} = \sum_{k1=1}^{K1} p_{q_{k1}}$$

, whereas $p_{q_{k1}} = 1$ is the size of the k1-th question property, $k1=1,..,K1$.

Then, each density D_{q_i} for a question q_i is compared with the properties of the answer candidates by checking whether or not an answer property

$p_{j,k2}^a$ is in the property density⁸³ of the question. Then the PropertyScore (PS) is calculated as follows:

$$\text{PS}(Q) = \frac{\sum_{k1=1}^{K1} \delta^p(P(q_i), P(a_j))}{d_{pq}}$$

where

$$\delta^p = \begin{cases} 1, & \text{if } p_{i,k1}^q = p_{j,k2}^a \\ 0, & \text{if } p_{i,k1}^q \neq p_{j,k2}^a \end{cases}$$

with $k1=1,\dots,K1$ and $k2=1,\dots,K2$.

The $P(\cdot)$ function denotes a function for selecting the $k1$ -th property in question q_i or the $k2$ -th property in answer candidate a_j , while the δ function represents the matching in the property comparison.

Properties that do not appear in the question but the answer instances have discriminating character, while additional instances in the answers can also indicate specialization of a certain answer. Property scoring disregards these properties for now. A semantic distance analysis as in the OntoScore approach could be integrated in future work.

2. Instance Score Calculation - Comparing Answer Instances

In a similar way the InstanceScore (IS) is calculated in the *instance slot filler* step by checking whether the instance or literal of the question o_q is equal to the instance o_a of the answer or not. Similar to the PropertyScore calculation each instance o_q in the question is compared to the answer instances o_a .

With o_{all} as the size of all instances in the question, the Instance Score is calculated as:

$$\text{IS}(Q) = \frac{\sum_{i=1}^n \delta^o(O(q_i), O(a_i))}{o_{all}}$$

where

$$\delta^o = \begin{cases} 1, & \text{if } o_q = o_a \\ 0, & \text{if } o_q \neq o_a \end{cases}$$

⁸³The set of possible question properties. Consider, that no duplicate properties are taken into account. The semantic density approach operates on the unique set of properties in an ontological instance.

The instance scoring allows the essential information in the answer candidate instances such as important literals, etc. to be considered in order to differentiate the correct answer to a question between similar candidate instances.

3. Final ProperScore PS'

The overall score ProperScore (PS') is the sum over the PropertyScore PS and the InstanceScore IS dived by 2:

$$\text{PS}'(Q) = \frac{\text{PS}(Q) + \text{IS}(Q)}{2}$$

6.5.3 Evaluation

The previously described algorithm for scoring ontological instances has been evaluated as being exemplarily via a corpus of 14 question (q) - answer (a) pairs, each representing a particular question type, obtained from the SmartWeb system demonstrator. The questions types used are listed in Table 6.14, while the answer instances have been gathered via the semantic wrapping system introduced earlier. For reasons of simplicity of the evaluation, the index of the correct answer has been assigned the same index as the question.

| Nr | Example Question Type |
|----|--|
| 1 | Gegen wen spielte Deutschland in Spanien |
| 2 | Wann war Brasilien Weltmeister |
| 3 | Wo fand die WM 1990 statt |
| 4 | Welche Spiele laufen gerade |
| 5 | Welche Tore schoss Michael Ballack bei der WM 2002 |
| 6 | Welche Tore schoss Ronaldo |
| 7 | Wo fanden die Spiele bei der WM in Deutschland statt |
| 8 | Wie steht es im Spiel Niederlande und Tschechien |
| 9 | Wer war 1990 Weltmeister |
| 10 | Welche Tore schoss David Beckham |
| 11 | Welche Spiele finden in Stuttgart statt |
| 12 | Wer schoss die Tore im Spiel Deutschland gegen Brasilien |
| 13 | Welche Position in der Tabelle belegt Finnland |
| 14 | Zeige mir den Tabellenstand in der Gruppe A |

Table 6.14: Questions used for the evaluation of the ProperScore algorithm.

The selected example instances cover all types of questions that have been directed to the SWA subsystem from the dialogue system, i.e. preferably different instances have been selected, while a few similar questions served to measure and verify whether they also get reasonable scores.

The scoring and ranking results for the best 3 instances in Table 6.15 show, that the overall average precision for selecting the best answer candidate is about 99%. The results of the second and third instance (<60%) further show that disambiguating the question-answer pairs works well for most of the cases.

| Query | Rank 1 | Rank 2 | Rank 3 |
|------------|-----------|----------------|--------------------|
| q_1 | 1.0 (a1) | 0.41 (a4, a11) | 0.3 (a12) |
| q_2 | 0.98 (a2) | 0.59(a9) | 0.5 (a1) |
| q_3 | 1.0 (a3) | 0.53 (a6, a10) | 0.51 (a1, a7, a11) |
| q_4 | 1.0 (a4) | 0.55 (a1) | 0.54 (a11) |
| q_5 | 1.0 (a5) | 0.55 (a10) | 0.46 (a6) |
| q_6 | 1.0 (a6) | 0.59 (a10) | 0.54 (a5) |
| q_7 | 1.0 (a7) | 0.56 (a1, a11) | 0.47 (a3, a5, a10) |
| q_8 | 0.93 (a8) | 0.44 (a1, a11) | 0.4 (a13,a14) |
| q_9 | 1.0 (a9) | 0.64 (a2) | 0.52 (a3, a6, a10) |
| q_{10} | 1.0 (a10) | 0.57 (a5) | 0.52 (a6) |
| q_{11} | 1.0 (a11) | 0.44 (a1) | 0.42 (a7) |
| q_{12} | 1.0 (a12) | 0.42 (a1) | 0.4 (a8) |
| q_{13} | 1.0 (a13) | 0.93 (a14) | 0.53 (a8) |
| q_{14} | 1.0 (a14) | 0.93 (a13) | 0.53 (a8) |
| Mean Score | 0.99 | 0.58 | 0.47 |

Table 6.15: ProperScore results for Dataset 1 (Appx. B.7).

A more detailed analysis of the results has been performed in order to investigate the effects of the instance scoring (IS) compared to the property scoring (PS) and their influence on the overall score PS'.

The spider diagrams of PS, IS and PS' in the Figures 6.24, 6.25, and 6.26 visualize the best selected answer candidate for a given query as peaks. The diagram with the PS scores shows, that the semantic range of the properties best fits the corresponding correct answer candidates, while similar instances have similar values.

Note that question 5 and 6 seem similar at first glance, while question 5 represents a more specific query directed at a specific tournament (Soccer WorldCup in 2002), while the other query is too general and would return

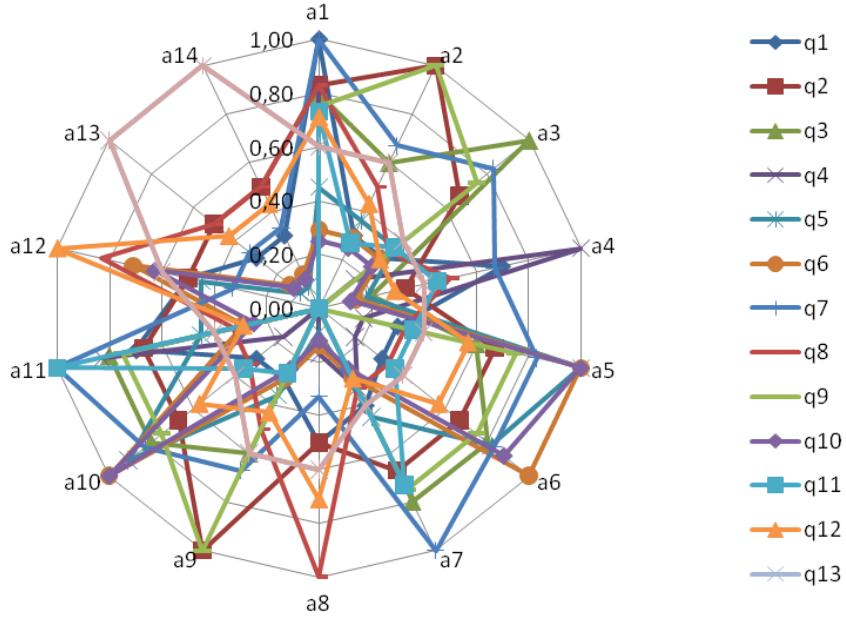


Figure 6.24: Evaluation scores of PS for 14 (q,a) pairs.

all goals “Ronaldo” has scored which can be found on the wrapped web sources. Despite this, the aforementioned context module (SitCom) should usually enrich the query 6 (and 10) with the contextual information about the current football (soccer) tournament. In this way, both query instances would conform to each other to a greater extent. Nevertheless, the queries 5, 6 and 10 are most similar which is shown by the scoring of the corresponding answer candidates. All three question have both of the other answer instances ranking at 2 or 3. The differences between answer instance 10 and 6 (whose questions are most similar) result from the differences of the returned dataset with additional individual information related to the football events (match, goals scored, etc.) obtained from the wrapped sources.

Looking at the results of the instance scoring, a clearer picture can be observed. All query-answer pairs are correctly disambiguated which is illustrated by the star form of the spider diagram. The answer instances for questions 13 and 14 seem to be scored close to each other. Despite this, the disambiguation also seems to work here sufficiently. The reason therefore, is the fact that answer 14 contains the answer instance returned by instance number 13 as well, here the additional information (position information of a soccer team) seems to be semantically very similar. Hence, again the results seem to be reasonable here as well.

In order to confirm the findings of the first experiment, two additional

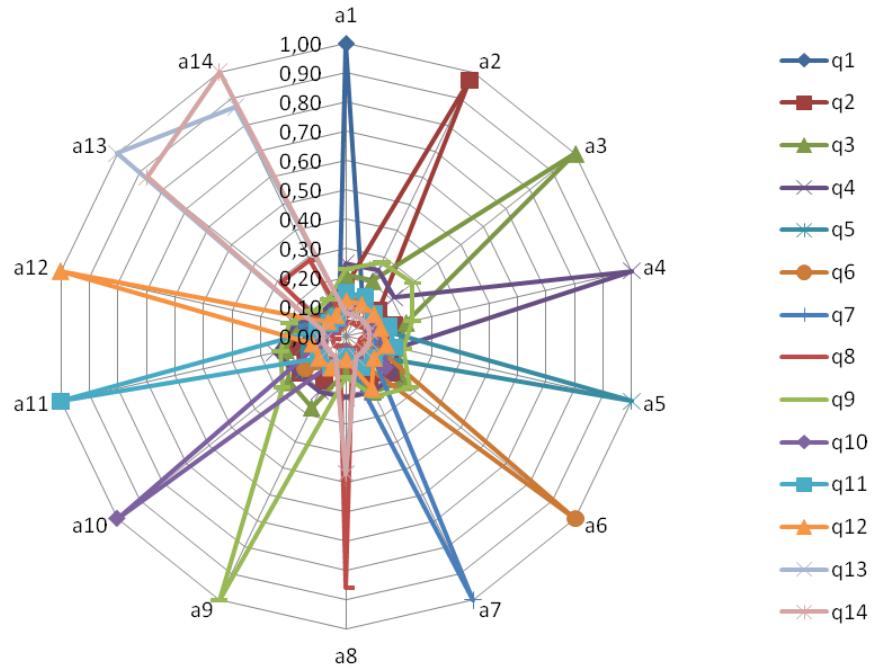


Figure 6.25: Evaluation scores of IS for 14 (q,a) pairs.

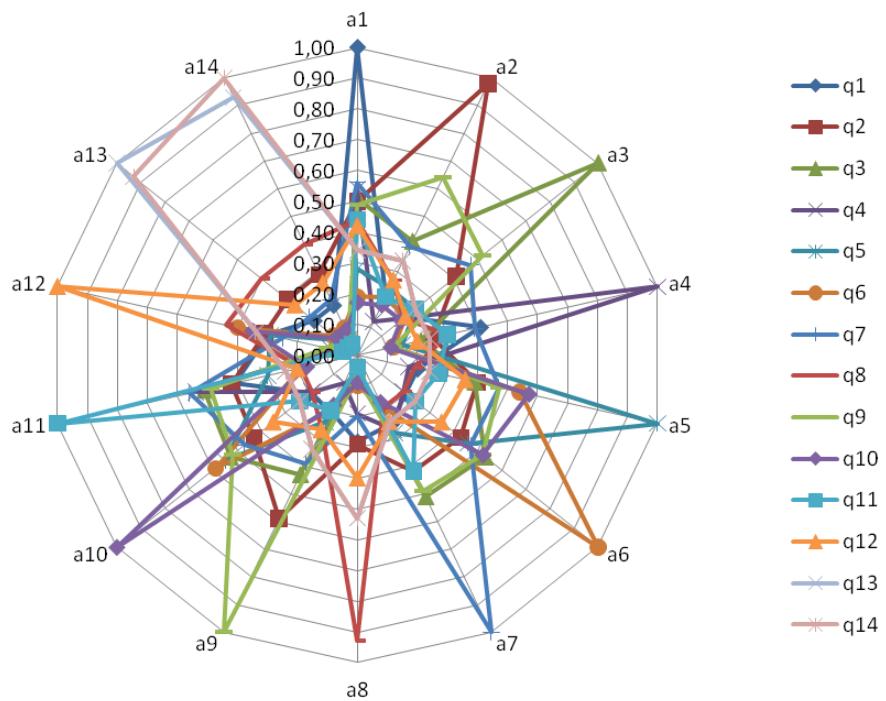


Figure 6.26: Evaluation scores of PS' for 14 (q,a) pairs.

datasets (Dataset 2 with 15 (q,a) pairs and Dataset 3 with 24 (q,a) pairs) were evaluated and the answer selection for different types of (q,a) pairs were analyzed. The related questions are listed in the Appendices B.5 and B.6.

| Average Rank Precision | Dataset 1 | Dataset 2 | Dataset 3 |
|------------------------|-----------|-----------|-----------|
| Rank 1 | 0.99 | 1.0 | 1.0 |
| Rank 2 | 0.58 | 0.76 | 0.57 |
| Rank 3 | 0.47 | 0.75 | 0.54 |

Table 6.16: Ranking results (PS) for the two other datasets.

In Appendix B.7 the detailed results for the overall comparison of the question-answer pairs are given as a colored matrix. The color green indicates best match, while gradations towards red point to the matching answer candidates matching the least. In dataset 3, the comparison of question 4 and the 25 answer candidates show that certain instances have a common ground of similarity resulting from overlapping subgraphs in the RDF graph of the regarded answer instances. In the aforementioned case, the general semantic pattern of the user question is matched under the RDF subgraph attached to the `inTournament` property of the SWIntO concept `Match` (see Appendix B.8). Despite the high score of the property density, the pair is disambiguated as the instance score (IS) shows only a similarity of 0.22.

The results (Table 6.16) show, that in all cases the first rank – the selection of the best answer candidates – again has a precision of 100%, while the other two ranks hint at similar data instances with varying parts.

Finally, the baseline evaluation for Dataset 1 (Table 6.17) applying the enhanced OntoScore algorithm (AnswerScore) confirmed the previously stated findings.

One observation about the baseline result is that the mean scores for the first three ranks are very low compared to the semantic density approach. In general the results also show, that the baseline method is not able to disambiguate the top three answers well. About 28% of the questions have no correct answers within the top 3 ranks, while the fraction of matched answers with the top 3 varies from 33% to 50%. Only in two cases (q_8 and q_{12}), a top match at rank 1 can be observed. The mean average precision (MAP) scores for rank 1 to 3 of the baseline are shown in Table 6.17, taking rank 1 selection of the ProperScore method as the human judgments or gold standard.

| Query | Rank 1 | Rank 2 | Rank 3 |
|------------|----------------|----------------|---------------|
| q_1 | 0.37 (a12) | 0.34 (a8) | 0.30 (a9) |
| q_2 | 0.245 (a12) | 0.241(a8) | 0.20 (a9) |
| q_3 | 0.27 (a8) | 0.25 (a12) | 0.20 (a9) |
| q_4 | 0.21 (a12) | 0.17 (a8) | 0.169 (a6) |
| q_5 | 0.30 (a12) | 0.27 (a8) | 0.278 (a6) |
| q_6 | 0.31 (a12) | 0.30 (a8) | 0.28 (a10,a6) |
| q_7 | 0.21 (a12) | 0.16 (a6) | 0.158 (a8) |
| q_8 | 0.49 (a12, a8) | 0.47 (a9) | 0.36 (a13) |
| q_9 | 0.24 (a12, a8) | 0.20 (a9) | 0.179 (a2) |
| q_{10} | 0.30 (a12, a8) | 0.28 (a6, a10) | 0.22 (a5, a9) |
| q_{11} | 0.20 (a12) | 0.16 (a6) | 0.158 (a8) |
| q_{12} | 0.547 (a12) | 0.538 (a8) | 0.51 (a9) |
| q_{13} | 0.267 (a12) | 0.217 (a6) | 0.2167 (a8) |
| q_{14} | 0.267 (a12) | 0.217 (a6) | 0.2167 (a8) |
| Mean Score | 0.30 | 0.27 | 0.246 |
| MAP | 0.14 | 0.10 | 0.29 |

Table 6.17: Baseline results (AnswerScore) for Dataset 1.

6.5.4 Conclusion

The evaluation of the presented method based on 3 datasets showed, that the semantic density method is able to disambiguate ontological question-answer pairs with high precision. The method works well as long the semantic variations or interpretations of the extracted instances are not dramatically, i.e. the set of relevant properties that match can be deduced from the ontological model. For example, in case of metaphoric meaning “not modeled exactly”, there might be less possibility to deduce such density overlaps from an incomplete ontology, but for factual knowledge, the results can be regarded appropriate.

It was also clearly shown that considering semantic content in the case of semantically similar structures (as it was the case in the used sports-event ontology of the SmartWeb system) is a viable approach, as more efficient extractions can be performed exploiting additional semantic information about the wrapped web contents itself.

Nevertheless, the method needs some improvements with respect to the problem of non-scored properties and non-overlapping information in the answer instances. As additional information in the answers might either represent non-relevant information or information that provides a detailed

interpretations of the extracted facts, more work on disambiguating such entities by exploiting taxonomic relations as well and evaluating semantic relations by considering their semantic granularity, is needed.

Furthermore, the results showed that for a generalizable approach, semantic relatedness how it is scored in OntoScore (Gurevych et al., 2003) and the semantic density approach should be integrated, in order to cover more complex ontological instances where semantic variations and interpretations can be resolved over more specific relations, while investigating general topics or semantic levels could be evaluated differently.

6.6 Question-Answering Workflow and Deployment

Suzuki et al. (2002) state, that question answering systems comprise at least the following 3 steps: document retrieval, extraction of answer candidates and answer selection. The SWA subsystem processes the received semantic representation of the user query from the dialog system in order produce relevant answer instances applying the following workflow:

1. Analyze the semantic query, i.e. identify semantic class c_i and focus f_i of a query q_i , which are derived from the SWEMMA representation (see Section 6.1.3) of the user query.
2. Lookup semantic similarity scoring matrix and select appropriate wrappers from the "pool of wrappers" $W = \{w_1, w_2, \dots, w_K\}$.

The wrappers are created by the users in the course of the semantic layering and wrapper generation process for the target web sites using the visual wrapper generation system described in Section 6.3.

In SmartWeb, wrappers for the FIFA WorldCup page, the T-Online soccer sites and other sport-related pages have been created accordingly, allowing the extraction of the target information structures. Thus, each wrapper w_k is associated with one or more URLs for extracting data instances from. And depending on the semantic class, which is stored in the :content section of the SWEMMA query one or more wrappers are chosen for a particular query.

| query/wrapper | w_1 | w_2 | $\dots w_i \dots$ | w_k |
|------------------|-------|------------|-------------------|------------|
| $q_1 : c_1, f_1$ | 0.2 | 0.4 | 0.8 | 0.1 |
| $q_2 : c_2, f_2$ | 0.3 | 0.7 | 0.1 | 0.2 |
| ... | ... | ... | ... | ... |
| $q_k : c_k, f_k$ | 0.4 | 0.2 | 0.7 | 0.9 |

Table 6.18: Semantic similarity scoring matrix for selecting wrappers.

Table 6.18 shows the example selection scores for existing wrappers in a wrapper pool W . For example, for query q_1 the wrapper with the highest score is w_i (0.8).

3. Extract data from online sources via the selected wrappers.

A selected wrapper w extracts data instances d_1, d_2, \dots, d_n from the URLs it knows about.

4. Transform extracted data to instances of the SmartWeb ontology in RDF-S and store to the temporary semantic repository applying the method described in Section 6.4.

5. Send a generalized form of the semantic query to the semantic repository and get answer candidate instances.

In this step a generic RDF query based on the semantic class of the query and its focus is sent to a semantic repository – which was created *on-the-fly* by the wrapper agent system. The temporarily created repository is used to collect and store all extracted and transformed ontological instances before the answer selection step.

6. Score and rank the stored instances by calculating the semantic similarity and relatedness scores of (query, answer)-pairs using the answer selection method described in Section 6.5.

7. Return the n-best list of result instances and their confidence values to the semantic mediator component of the SmartWeb dialogue system. The semantic mediator analyzes and creates text-to-speech representations for the answers returned to the user immediately.

6.6.1 Deployment in SmartWeb

The general deployment of the semantic wrapper agents approach in SmartWeb can be described in three phases. Each phase implements a corresponding layer: Wrapper, Mediation and Query Interface, which is illustrated in Figure 6.27.

Phase 1: Generating Semantic Wrappers (Wrapper Layer)

In the first phase, wrappers are generated using the approach that was described in Section 6.3. A user of the system visits a page with semi-structured data and wraps relevant content using the visual selection, labeling and wrapper generation capabilities of the visual extractor tool JEFF (see Appendix B.2). The generated wrappers are added to the wrapper table maintained by a dedicated broker agent that manages a collection of wrappers deployed in the agent system.

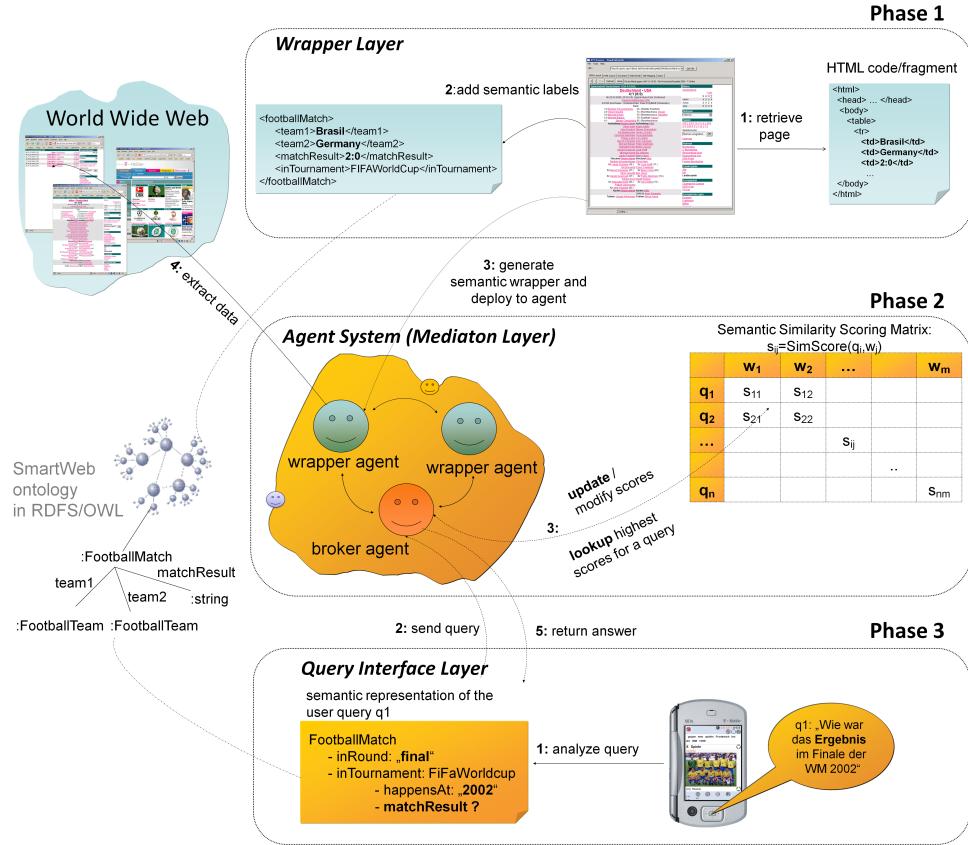


Figure 6.27: Big Picture showing the Semantic Wrapper Agents approach.

The generated wrappers are managed by a wrapper table and can be used by the agents to semantically access the web content and do further semantic processing. The mapping to RDF is done based on the corresponding domain ontology, e.g. the SmartWeb sport-event-ontology for annotating/wrapping web pages from the football domain. The extracted RDF triples (subject, predicate, object) are stored in a semantic data repository using RDF and can be queried via semantic web query languages like SeRQL provided by the used Sesame framework⁸⁴.

Phase 2: Deploying Wrappers as Agents (Mediation Layer)

Bootstrapping the agent-based system by deploying various wrappers as agents can be regarded as the second phase of the overall goal for realizing the agent-based semantic access to semi-structured web pages.

The multi agent system manages several agents that work as brokers or

⁸⁴Sesame: <http://www.openrdf.org/>

wrappers. While wrapper agents are responsible for processing extraction requests for assigned web sources utilizing specialized focused crawlers (Liu, 2007d, ch. 8), the semantic broker agent is responsible for processing the semantic user request and orchestrating the wrapper agents for extracting the desired information instances and translating the instances into valid ontological instances of the SmartWeb domain ontology. Furthermore, the broker agent manages the wrapper table (wrapper pool) and the semantic similarity scoring matrix of (Q,A)-pairs. Appropriate answer candidates for the current query are found by analyzing the semantics, e.g. semantic class, focus, etc. of the user query and consulting the scoring matrix and the wrapper pool.

```
(request :sender userAgent1
         :receiver brokerAgent
         :content
        (
          <rdf:RDF ...>
            <j.1:Query rdf:about="http://smartweb.org/ind#i4">
              <j.1:text rdf:datatype="#string">wer war ... </j.1:text>
              <j.1:dialogueAct>
                <j.1:Question/>
              </j.1:dialogueAct>
              <j.1:focus>
                <j.2:DivisionNationalTeam
                  rdf:about="http://smartweb.org/ind#i5"/>
              </j.1:focus>
              <j.1:content>
                <j.2:WorldCup>
                  <j.2:heldOn rdf:datatype="#string">1990</j.2:heldOn>
                  <j.2:winner rdf:resource=".../ind#i5"/>
                </j.2:WorldCup>
              </j.1:content>
              <j.0:confidence rdf:datatype="#float">0.75</j.0:confidence>
            </j.1:Query>
          </rdf:RDF>
        )
        :ontology SWInt0
        :language RDF)
```

Figure 6.28: FIPA-ACL request message.

The semantic similarities in the scoring matrix are calculated via the aforementioned instance scoring algorithm used for answer selection which was described in Section 6.5. Hereby, the results of each (question, answer)-pair evaluation are added to or updated in the similarity scoring matrix. Furthermore, for each new query the semantic similarity score to existing queries is calculated via the same answer selection method and utilizing a similar matrix as shown in Table 6.18. The difference is that the matrix

of rows and columns are queries, while the value of the cells are semantic similarity scores. The queries having the highest similarity to the current query are selected and their respective URLs are used as the sources for data extraction for answering the current question. The intuition behind this is that similar questions can be answered by posing questions to previously accessed web sources with similar or related semantic classes. Finally, a threshold serves to constrain or judge the similarity of user incoming queries to the existing one.

During the “warming-up” of the agent system, initial scores are added to the scoring matrices and updated after each dialogue turn accordingly. The processed query-response instances are bound to the SmartWeb ontology, hence, agent communication using ACL is performed via the SWIntO ontology as the content language. Figure 6.28 shows the ACL request message for the example user question sent from the *userAgent* (which communicates with the dialogue system) to the *brokerAgent* and Figure 6.29 the corresponding answer returned by the wrapper to the brokerAgent.

```
(inform :sender WrapperAgent1
        :receiver brokerAgent
        :content
        (
            <swemma:Result rdf:about=".../answerFor-s0-001">
                <emma_:confidence>0.31</emma_:confidence>
                <emma_:container>
                    <emma_:OneOf rdf:about=".../answerFor-s0-002">
                        <emma_:container>
                            ...
                            </emma_:container>
                        </emma_:OneOf>
                    </emma_:container>
                    <discourse:dialogueAct rdf:resource=".../sitcom#sc4.670"/>
                    <emma_:derivedFrom>
                        <discourse:Query rdf:about=".../sitcom#sc456908078.653">
                            ...
                            <discourse:dialogueAct
                                rdf:resource=".../sitcom#sc456908078.670"/>
                            <emma_:id>1182859965201</emma_:id>
                        </discourse:Query>
                    </swemma:Result>
            )
            :ontology SWIntO
            :language RDF)
```

Figure 6.29: FIPA-ACL inform message.

Phase 3: Question Answering (Query Interface Layer)

The third phase, is initiated by a user sending out a natural language query to the system.

In the first step, the user query is translated to its semantic representation in RDF and arrives (passing several other components) at the semantic broker agent of the SWA web service component.

The semantic broker agent analyzes the SWEMMA query in order to select appropriate wrappers supporting the respective semantic class. Therefore, the broker agent computes the semantic coherence (relatedness or similarity) between the query and the semantic classes stored in the wrapper table.

Besides that, the focus of the query is identified and used for the refinement of the extracted answer instances as a previous extraction process might have extracted a richer semantic structure having additional properties and other linked instances. The broker then advises the semantic crawler component to follow associated/linked web sources in order to extract the data instances executing the selected wrappers. Thereafter, the broker agent translates the extracted results to RDF and stores the extracted data instances in the Sesame⁸⁵ RDF repository. The updated semantic repository is queried via the SeRQL query language of the Sesame framework. Therefore, the semantic broker has to translate the EMMA query to SeRQL (see Appendix B.5). Basically, the semantic pattern of the query is embedded into a SELECT clause in SeRQL and a WHERE instruction used for indicating the focus of the query. The SeRQL query language is described by Broekstra and Kampman (2004). The SeRQL language allows the definition of CONSTRUCT queries which return RDF graphs as a set of triples and is therefore suitable for extracting ontological instances from a candidate repository conveniently, and was therefore preferred over alternative languages such as SPARQL.

In this instance, the Sesame RDF repository serves as an intermediate cache, storing exchanged semantic instances coming from the single wrappers. Querying a central repository has the benefits of allowing broader queries over a collection of semantic RDF instances, which is due to performance reasons. The broker then sends the SeRQL query to the semantic repository and retrieves the results that are scored by the answer selection method. In the final step, the answer selection method is applied in order to calculate and rank the n-best result instances, which are then returned to the SmartWeb dialogue system via the semantic mediator component. The (normalized) confidence scores have values in the interval [0,1].

⁸⁵<http://www.openrdf.org/>

6.7 Conclusion

The aim of the case study presented was to examine how semantic wrappers can be employed as part of a semantic question answering pipeline in a knowledge-based real-world dialogue system (SmartWeb) in order to answer the natural language questions of users. Focusing on an online question answering scenario a general concept and a workflow for semantic labeling, extraction and querying of information structures inside semi-structured web pages was investigated.

Answering the natural language questions of users of a dialogue system is a challenging task, requiring the interpretation of the user queries by means of semantic analysis and providing and maintaining semantic access to several target web sources in order to extract and identify relevant answers. In the approach presented, semantic wrappers generated by a visual wrapper generation system have been employed in order to extract and transform complex information structures of interest from different web pages for a focused domain.

Therefore, sample instances of the target information structures had to be labeled with concepts and relations from the used system-wide ontology (SWintO) employing a visual user interface. Although visual assistance in wrapper generation was also applied in other systems (see Section 6.3.2 and 3.2.1), the first important contribution of the presented approach was the use of linguistic information (e.g. in a certain domain language) for labeling and composing complex annotations (e.g. events in a football match) that are exploited in information extraction and semantic transformation for creating valid ontological instances.

In general, the contributions related to the wrapper generation approach comprise means for:

- exploiting linguistic information for user-friendly semantic tagging of complex information structures beyond simple entries and lists.
- integrating an efficient to use semantic tagging metaphor (atomic tagging and grouping selection) for building complex instances of a lightweight-ontology to be extracted.
- creating generalized extraction rules for a tagged (hierarchical) semantic structure (semantic wrapping) on-the-fly by means of visual interaction.

Besides that, deploying semantic wrappers into an agent system required the maintenance of semantic access to several web sources in order to access and integrate additional aspects of information by means of “a pool of specialized wrappers”. Therefore, the created semantic wrappers had to be integrated into a semantic question-answering pipeline in the SmartWeb dialogue system based on the system-wide ontology (SWintO). Hereby, an essential step comprised the semantic transformation of the extracted information structures to valid ontological instances and a scoring method for pairs of questions and answer instances.

In addition, a reduction of the instances to be extracted was achieved employing a form of semantic indexing on previously extracted semantic structures by employing a semantic similarity matrix in broker agents that store a score of similarity for previously executed semantic queries. The semantic similarity score allows the selection of appropriate wrappers for extracting the answer instance candidates.

Furthermore, a semantic representation of the target information allowed to pose semantic queries online (e.g. using any RDF query language), hence, accessing the current status of a target web page. Particularly in case of accessing frequently changing web pages, this type of access is of major importance for freshness of the data instances to be extracted. The demonstration of the system shown at the Soccer World Cup in 2006 in Berlin, proved to be able to query football (soccer) match results, where the contents are updated frequently due to important events during game play, such as goal scoring, penalty, substitutions, etc.

With the presented proof of concept study it was shown, that the applied workflow based on the building blocks and the conceptual design provided by the SI framework introduced in Chapter 4 is viable for a focused domain and applicable in the context of a real-world dialogue system for the regarded task of question answering. The overall evaluation of the major components for extracting and scoring of ontological instances (Sections 6.3.5 and 6.5.3) showed that the used methods are able to generate accurate results for the given question answering use case. The evaluation focused on the main question types provided by the SPIN parser and the dialogue interaction components described in Section 6.1, whose expressiveness was restricted to a range of question types that were relevant for the regarded use case and question answering scenario.

The used instance scoring method worked well for semantically similar question-answer instances with a restricted focus (e.g. question: “Who scored in the 40th minute?”, answer: “Ballack scored in the 40th minute!”).

In the case of big differences between the question-answer instances, the method did not work well. It can generally be applied well for frame/slot-based information extraction approaches.

Answering the natural language questions of users of a dialogue-system consulting semi-structured web sources, requires a robust and consistent way to extract and semantically express the meaning of the returned answers. Obvious problems resulting from the applied wrapper generation approach can be expected in the case of frequently changing web page structure. While structural change could be resolved through re-creation of the wrappers, semantic changes would be more severe. Here, an on-the-fly approach would no longer be feasible. For classic web pages, the applied semantic wrapping methods for structured data will remain without many alternatives.

Chapter 7

Semantic HCI for Visual Exploration

In human communication, metaphors are widely used to provide a short-hand description of complex facts, events or circumstances. The prominent metaphor by Shakespeare of “*All the world’s a stage ...*” in his comedy “As You Like it”, supposed to have been written around 1600, is a well-known example.

In the world of computers, metaphors play a central role in human-computer interaction as part of the human-machine interface, aiming at encoding complex interactions and tasks into an understandable set of operations with, e.g. visual interface elements, by making use of the user’s general purpose or background knowledge. Metaphors encoded in user interfaces can mediate *meaning* to users of how to interact with user interface artifacts of computer systems, offering interactions which can have counterparts in the real world, e.g. desktop or surfing metaphor. In visual retrieval systems (Section 3.4) the potential of enhanced interaction metaphors has for long time been disregarded and interaction solely restricted to “entering keywords” and “sighting results” (Hearst, 2009). The imagining of other forms of human computer interaction for exploring (large) information spaces has not only been discussed in research since the appearance of Spielberg’s “Minority Report”.

In the Chapter 6 the focus of the research was on enabling semantic interaction using a conversational metaphor in the context of a dialogue-driven knowledge-based question answering system. In this chapter, the research focus is on enhancing visual retrieval interfaces of folksonomy-based

retrieval systems for exploring (large) information spaces by exploiting semantics of user-provided tags and employing enhanced visual interaction and navigation metaphors.

Motivation and Contribution

The research described herein was motivated by the increased deployment of semantic information across the web on the one hand, and the widespread adoption of new interaction modalities such as touch input, Wii remote Controller, Microsoft Kinect, etc. on the other hand, offering new means for supporting human computer interaction for exploring information spaces. Besides meta-data in form of keyword tags, which is introduced by users that interact and create content on the social web, the growth and increasing impact of mostly expert-created linked data on the semantic web as well, confronts research with the challenge to provide enhanced visual interfaces that exploit semantic information and provide rich, interactive capabilities for empowering semantic search and serendipitous discoveries (see discussion in Section 3.4.4 and the tasks described in Section 4.5). Hence, the research can be seen in the context of Marchionini's debate (Marchionini, 2006) on new ways of search and exploration in information retrieval systems.

The focus and the contributions of this chapter are related to visual retrieval interfaces for folksonomy systems, although the presented methods and concepts could be applied to other types of retrieval systems in a similar manner. The general hypothesis is that adding semantic interaction capabilities to 2D/3D visual retrieval interfaces by means of semantic interaction metaphors will enhance the users browsing experience and improve user satisfaction, particularly for use cases of undetermined browsing and exploration of information spaces.

SI Framework for Folksonomy Systems

A general workflow for building visual retrieval interfaces for exploring folksonomy systems based on the Semantic Interaction (SI) framework introduced in Section 4.3, encompasses two basic parts, as illustrated in Figure 7.1.

The first part (left), deals with the semantic analysis of the stored folksonomy data and encompasses the steps of pre-processing and normalization, analysis of semantic relations in the tag lists and semantic aggregation for creating the target SI artifacts (related tags, tag cloud, etc.) to be visualized. The second part illustrates the use of one or more SI metaphor(s),

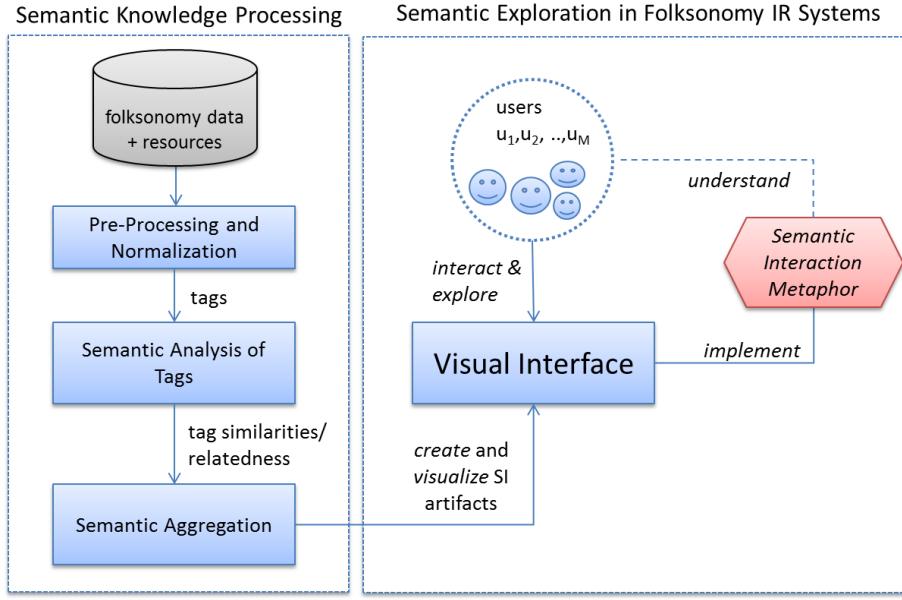


Figure 7.1: Semantic Interaction in visual retrieval interfaces.

which have to be implemented in the visual retrieval interface. A respective semantic interaction metaphor (SIM) materializes in the user interface via the visualized SI artifacts and the corresponding actions and interactions for allowing the navigation and exploration of the information space, e.g. applying "zooming" metaphor for focusing on more specific topic regions or clusters.

In the case study described in Section 7.1, the framework was applied for implementing a semantic tag cloud interface that can be utilized for navigating and exploring topic regions performing interactions for zooming to distinct semantic levels visualizing related tags from general to specific, etc. Similarly, the approach described in Section 7.2 shows semantically arranged documents and related tags in 3D visualizations. Finally, in Section 7.3 a method for navigating (large) semantic information spaces in 3D via touch or mouse input is presented in a case study.

7.1 Case Study: Semantic Tag Cloud

In recent years folksonomy systems have utilized tag clouds as a visual retrieval interface for exploring user-generated data in a restricted form, mainly arranged in alphabetical order and visualizing the most popular or very general tags. In the tag cloud interface, tags from related areas of interest are widely invisible to the users. From the search and retrieval interaction perspective, it is difficult if not impossible for users to find search terms quickly starting with an initial tag, as relatedness is not reflected in the tag cloud visualization. Although some interfaces, e.g. delicious, use a list of related tags, they are restricted to the top 10-20 general tags that have been used frequently with the given search tag. Furthermore, they do not support building complex queries due to missing interaction structures that take into account semantic levels and a topical organization of the information space.

This Chapter investigates means to resolve the described limitations of existing tag-cloud based visual retrieval interfaces by introducing a semantic dimension to interaction with data in folksonomy systems. Therefore, semantic levels of tags are analyzed and a hierarchical multi-topic semantic interaction metaphor is employed in a semantic tag cloud interface (Aras et al., 2010). Important tasks are concerned with the integration of such a semantic tag cloud structure into an interactive visual user interface in 2D and the multi-topic hierarchical semantic exploration of content in a folksonomy system.

The starting point is the semantic analysis of tags in a folksonomy via the co-occurrence method, which allows the identification of semantically related tags. General (conceptually) related tags, as well as tags which describe specific aspects of a respective resource are extracted from the folksonomy hypergraph in order to form multi-level topic clouds. The topical organization at distinct semantic levels is achieved by applying a hierarchical clustering method. In other words, clustered, related tags represent generally related terms and topics at higher cluster levels, while describing more specific aspects at deeper levels. A suitable semantic navigation metaphor for ascending and descending different semantic levels is employed in order to allow for searching and exploring the folksonomy information space.

A comparison in a user study shows that the multi-level semantic tag cloud approach is superior to the baseline state of the art user interface approach in many aspects.

7.1.1 State of the Art

The tag cloud as a visual retrieval interface for folksonomy systems has many advantages, e.g. low cognitive overhead for selecting search tags, etc. as discussed previously in Section 3.1.2.

In general, the tag cloud is created on the basis of the user-assigned tag lists for the resources that have been stored in the folksonomy system. Tags as social annotations represent numerous aspects of the tagged resources, such as rich language variations, topical changes, contextual information, etc. and are thus of high collective value.

Unfortunately, folksonomic tags also have problems (Section 2.3.1) stemming from the ambiguities and complexities in natural language and its everyday use. Common normalization methods are widely based on the Levenshtein distance metric (Gusfield, 1997, pg. 216) for resolving issues with different word forms e.g. singular/plural or spelling correction (e.g. <http://afterthedeadline.com/>), stemming or exploiting external sources such as Google, WikiPedia, thesaurus or word nets.

Looking at retrieval interfaces for folksonomy systems (Section 3.4.2), tag-based search as well as widespread forms of a tag-cloud for browsing resources provide no means for a determined and structured exploration. Semantic information is rarely exploited by the respective retrieval interfaces. Furthermore, the interfaces provide less support for the users to find relevant search tags and are less interactive. The users even don't know which tags are stored in the system they could use, because only the most frequently used limited set of popular tags is shown to the users.

Methods such as semantic proximity or relatedness of search terms can be utilized to improve retrieval efficiency as well as the user experience for searching related information. In the context of folksonomy systems, conceptual relations can be represented via related tags, which can be extracted from the folksonomy analyzing tag distributions and tag co-occurrence relations (Section 2.4.3). Analyzing the context of related tags allows us to find and recommend specific tags to the users, that could be used to build more precise queries. Specia and Motta (2007) report, that calculating the normalized co-occurrence for each pair of tags applying the cosine similarity metric is a viable method, as will be verified later for the used dataset of this case study. In order to reduce information overload, clustering of textual, semi-structured and structured data can be utilized to aggregate diverse information structures efficiently, as briefly described in Section 2.5.2.

Although, various work was conducted to represent data in 2D/3D

space, semantically enabled visual retrieval interfaces for folksonomy systems, which provide improved navigation and interaction capabilities and make use of semantic information, are rarely found. In general, tags are taken “as they are entered”, with no or little semantic normalization by, for example, recognizing sense, etc.

Moreover, one suggestion by Hearst and Pedersen (1996) was, that (visual) user query refinement at different semantic levels of the information space, e.g. via keyword-based search facilities, should be integrated seamlessly with facilities for browsing or navigation in order to be effective. Moreover, they suggest the use of clustering - and in a wider sense - semantic aggregation integrated with keyword search.

Refining specific user queries by means of visual interaction could help to exploit semantics at the borders of topical or semantic regions – which resulted from an appropriate clustering process.

7.1.2 Approach

A visual retrieval interface concept based on semantic information in its core requires unsupervised methods for detecting and extracting semantic relations from the tag space. Semantic relatedness of tags, which is the basis for realizing a semantic arrangement within a tag cloud and creating a hierarchical structure of groups of tags can be obtained by analyzing a sufficient and representative set of folksonomy data.

Semantic Similarity and Clustering

The tag space in a folksonomy system can be very large, considering the number of users in systems such as delicious or flickr which have millions of members. As a consequence, manual definition of cluster centers around which tags are placed or calculating the optimal cluster size manually, can hardly be accomplished. In an unsupervised approach, ideally, the used clustering algorithm should preferably be able to split the tag set into small and reasonable clusters, where each topic is accessible from the top overview of the interface. The most general tags of each cluster have to be displayed first. For a multi-level multi topic representation in a semantic tag cloud, clusters must be divided into sub-clusters.

Related Tags and Hierarchical Exploration

A consistent exploration of related items requires the exploitation of distinct semantic levels, such as overview level with general concepts and deeper levels with specific terms. Technically, similar or related tags are physically located close to each other in the user interface – in contrast to the alphabetic or random arrangement of classic tag clouds. This allows a user to explore neighboring tags, after selecting an initial tag, in order to discover other potentially interesting tags. Furthermore, related tags can be refined by adding additional tags manually in order to satisfy specific information needs. As one single tag cloud is not sufficient to represent complex categories and their sub-categories, an extensive structure of multiple topic clouds that can be explored hierarchically is proposed. The basic idea is to provide a small representative overview tag cloud (food, cooking, health, etc.) that allows us to retrieve more focused tag clouds with a higher semantic density, i.e. very specific related terms (vegan, Mexican, beef, etc.) on demand. In this way, more enhanced queries can be composed.

Query Composition and Semantic Interaction

Integrating the proposed main concepts (tag cloud, related tag and hierarchical organization of tags) in one single interaction structure, allows for simultaneously composing queries from the tag cloud, consulting results and refining the query at the same time. Hereby, finding appropriate search terms for creating efficient queries and gaining a more complete impression of the tag space is possible. Furthermore, such a semantic arrangement and visualization of tags and clusters allows the composition of complex queries by exploiting the different hierarchical levels, i.e. queries can be built from tags at different conceptual levels of the tag hierarchy from general to more specific.

The workflow for creating semantic tag clouds comprises the following steps for pre-processing of the tag lists (Task 1), calculating semantic relatedness (Task 2) for pairs of tags based on co-occurrence analysis and applying agglomerative clustering for obtaining the hierarchical topic clusters (Task 4) that can be visualized (Task 4) in subsequent steps.

Task 1: Data Acquisition and Pre-processing

For a user study, a sample set of bookmarking data from the delicious folksonomy system was extracted during a week in 2008 via RSS and JSON

feeds⁸⁶ using the delicious Open API. All in all, 870.500 annotation triples on 119.817 distinct URLs with 42.373 distinct tags were retrieved.

Furthermore, a suitable corpus for calculating the similarities based on co-occurrence analysis had to be created by filtering the annotation data set according to the following parameters; *number of users*, *used number of distinct tags*, *maximum number of (popular) tags* – as these depend on the particular folksonomy system.

In a pre-processing step, rarely used tags and non-representative annotations were removed for gathering characteristic structures and relations in sufficient quality. As was elucidated in Section 2.3, in a collaborative tagging system potentially representative annotations can be identified by counting the frequency of how often a particular tag was used for tagging a certain resource. In general, the assumption is that rarely used tags are meaningful to individual users while frequently used tags are the ones commonly agreed (Golder and Huberman, 2006, Halpin et al., 2007) on as appropriately describing a resource. For this reason, bookmarks that were tagged by less than 20 users and with appropriate number of frequently used tags were removed from the corpus, as no useful co-occurrence information can be obtained for these. In broad folksonomy systems in particular, where annotations of different users on resources can be aggregated, the threshold for “rarely used” tags has to be set higher than in small folksonomy systems, where the quality of annotations cannot be determined by tag distribution.

Therefore, for the maximum number of popular tags, 10 was set as the maximum, while only such annotations remained that had been used by more than five users. Furthermore, bookmarks and corresponding annotations that were used more than five times in the remaining annotation set were preserved. The final data set comprised of 4707 tags used in 446812 annotations on 57830 distinct URLs.

Task 2: Tag Similarity Analysis

Tag lists attached to individual resources in folksonomy systems have no explicit semantics and can consist of multilingual tags. Despite this, implicit semantics can be derived consistently by analyzing the context and use of the tags. In general, semantic relations can be extracted from the consolidated tag sets by applying either statistics based (latent) semantic analysis by utilizing, e.g. the co-occurrence analysis for pairs of tags (see Section 2.4.3) or semantic analysis based on external knowledge sources

⁸⁶<https://delicious.com/developers>

such as WikiPedia or WordNet. In order to deal with amounts of data and associated tags from ten thousand (if not more) members of a folksonomy system, unsupervised methods to uncover semantic relations are needed. A successfully applied method for determining the semantic relatedness of tags is given by calculating the normalized co-occurrence for each pair of tags applying the cosine similarity metric (Definition 2.7) introduced in Section 2.4:

Similarity Metric Evaluation

In order to determine an appropriate similarity metric for most precisely identifying related tags in the extracted dataset, different similarity metrics (see Section 2.4.3) have been compared. In this study, the absolute co-occurrence metric, the Jaccard coefficient and the cosine similarity metric applied to resources (URLs) and tag vectors have been calculated.

In the evaluation a manual inspection and rating of the semantic relatedness was necessary to verify the results due to missing a suitable external representative knowledge-base for this task. The idea of using a Thesaurus or consulting WikiPedia was discarded, as a high percentage of folksonomy tags cannot be found in external knowledge sources - as reported by Laniado et al. (2007) and Guy and Tonkin (2006).

The following set of 15 tags sampled from the extracted delicious dataset served to determine the most appropriate similarity metric applying manual inspection: *animal, asia, cycling, children, clothing, design, finance, medicine, movies, outdoor, photography, recipes, sports, travel, programming*. The used tags originated from different topics and had different frequencies of use in the entire annotation set. In order to be useful for rating similarity or relatedness, such tags were selected where common sense can be applied easily to judge the results.

For each of the chosen tags, their twenty-five most related tags for the used similarity metric were assessed from a subjective point of view of how related they actually were by applying a three-point scale. A simple rating scheme based on semantic or lexical similarity has been applied as follows: if tags were similar or strongly related, 2 points were assigned; 1 point if they were semantically related to each other and zero points if the indicated relationship was rather random or very general. Example 7.2 shows the example for the tag “recipes”.

Best results were obtained by applying the normalized cosine similarity metric to tag co-occurrence vectors, as is shown in Table 7.1. The example

| Abs. Cooccurrence | Jaccard | Cosine (URL Vectors) | Cosine (tag vectors) |
|----------------------------|----------------------------|----------------------------|----------------------------|
| food cooking <u>recipe</u> | <u>recipe</u> cooking food | <u>recipe</u> cooking food | <u>recipe</u> cooking food |
| baking blog dessert | baking dessert | gourmet foodblog | baking breakfast |
| reference vegetarian | vegetarian foodblog | cookbooks drink | dessert appetizer |
| howto blogs foodblog | chocolate vegan bread | recipes drinks alcohol | vegetarian casserole |
| health chocolate | nutrition health diet cake | vegetarian kitchen | cheese bread pie pasta |
| nutrition vegan bread | cookies vegetables soup | vegan baking useful | sauce dinner soup |
| diet cake diy tips | breakfast desserts | meals dessert bread | foodblog salad dough |
| cookies vegetables | pumpkin foodblogs | cook healthy reference | mexican beef tofu |
| soup fun breakfast | cheese blogs chicken | veg chocolate | crockpot desserts |
| | dinner | cupcakes blog | beans |
| Rating: 18 | Rating: 24 | Rating: 24 | Rating: 26 |

Figure 7.2: Related tags for the tag *recipes* with different similarity metrics. Underlined tags are very similar, bold tags related and all other not appropriate (Aras et al., 2010).

shows high coverage of related tags, and no very general and not related tags such as blog, reference, etc., which are found by the absolute co-occurrence and the Jaccard metric.

This seems reasonable as the normalized cosine similarity metric also considers the context (Specia and Motta, 2007) of directly co-occurring tags, allowing the identification of relations between tags which do not co-occur in a document. The difference in contrast to the other metrics are especially outstanding for such tags which have a low frequency in the sample data set, e.g. animal occurring only 39 times.

| | Abs. co-occurrence | Jaccard | Cosing (URLs) | Cosing (Tags) |
|--------|--------------------|---------|---------------|---------------|
| rating | 213 | 309 | 301 | 361 |
| mean | 14.2 | 20.6 | 20.1 | 24.1 |

Table 7.1: Average ratings for different similarity metrics.

Task 3: Semantic Aggregation

In the context of this study, semantic aggregation can be described as a task of building organizations of conceptual levels and hierarchies, e.g. partitioning the tag space into reasonable semantic or topical regions, e.g. sport, food, etc.

Depending on the granularity or the aggregation level of the obtained

related tags, different forms of semantic interaction can be afforded. Fast and direct overviews over the tag space can be provided by utilizing aggregation levels with lower semantic densities, i.e. tags that are related in a general sense (audio, music), whilst for query refinement more specific levels can be consulted.

The most important aspect here is to enable users to modify and refine a query providing a rich set of associated (related) tags for a topic field, i.e. users are able to see which tags often co-occur in order to refine or change their queries accordingly.

Technically, the tag space must be clustered into appropriate topical clusters applying (preferably) unsupervised clustering techniques by processing the similarity values calculated above. The selection of the most fitting clustering method depends on the task specifics and the general objectives of the browsing and exploration scenario.

An important goal is to obtain a balanced distribution of thematic fields and their hierarchical consolidation from lower (more specific) to higher (general/conceptual) semantic levels. The benefits of this approach are that:

- users are supported in focusing the context of a certain search topic
- the (aggregated) semantics can easily be mapped one-to-one to a visual interaction structure the user understands

Hierarchical Clustering for Topic Clouds

In the previous task the similarity values for all pairs of tags were calculated applying the normalized cosine similarity metric on tag co-occurrence vectors. In order to partition the tag space into topic clusters an appropriate hierarchical clustering algorithm has to be applied. Gemmell et al. (2008) proposed an efficient agglomerative hierarchical clustering algorithm (Algorithm 4) that is utilized in a similar way for grouping highly related tags.

In the beginning each tag represents a single cluster. In an iterative process all clusters that are similar are joined until there is only one cluster left. The similarity between two clusters is calculated by applying the *centroid-link* method for calculating the average similarity between the tags in the two regarded clusters (Manning et al., 2008), i.e. the similarity between their centroids. In other words, general concepts are inferred from the degree centrality of the tagging graph, which was also defined in link analysis in Section 2.1.1.

Algorithm 4 Agglomerative Hierarchical Clustering (Liu, 2007d)

```

1: start with n clusters  $\{c_1, c_2, \dots, c_n\}$ , where each cluster is a tag
2: calculate all pair-wise similarities:  $sim(c_i, c_j)$ 
   applying cosine metric (Definition 2.7).
3: while ( $nrOfClusters > 1$ ) do
4:   find and merge pair of most similar clusters to a new cluster c
5:   add new merged cluster c and set as parent of merged clusters
6:   for all ( $c, c_j$ ) do
7:     calculate centroid-link cluster similarity:
        $sim(c, c_j) = \mu(c) \cdot \mu(c_j)$ ,
       with  $\mu$  as the centroid vector of a cluster.
8:   end for
9: end while

```

The centroid vector or center of gravity for M points in a cluster c is given as:

$$\mu = \frac{1}{M} \cdot \sum_{i=1}^M x_i$$

, where x_i is the vector of the i-th member of a cluster.

The clustering process can be represented as a binary tree of clusters and sub-clusters. The obtained tree structure can be cut according to a (minimum) similarity threshold or (maximum) number of clusters. In Figure 7.3 an example of a hierarchical structure is shown. The dashed line splits the tree into four reasonable top-level clusters which again can be subdivided into several sub-clusters.

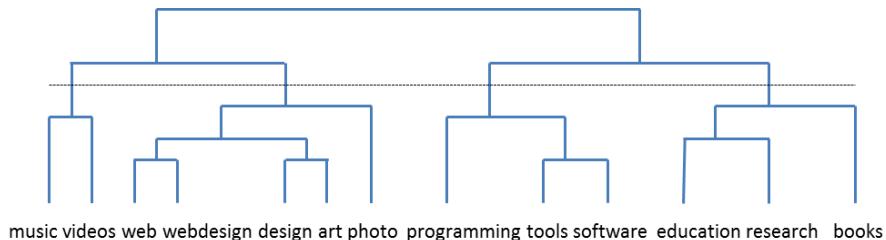


Figure 7.3: Semantic Cloud Topic Clustering Example.

While the first level clusters contain the more general (high-level) topics, each represented by their most popular related tags in the top level (overview) tag cloud of the interface, their sub-clusters form the lower levels that can be consolidated hierarchically. In order to gather the most re-

sonable semantic clusters for evaluating the interface concept, the cutting thresholds have been determined manually. In future work, an appropriate method for setting the similarity thresholds automatically needs to be investigated.

The clustering algorithm generates the hierarchical data structure that is used by the user interface and the graph visualization.

Task 4: Visualization

The obtained internal semantic representations as created by the clustering process from the preceding step must be mapped to a visual representation based on

- a tag-based visual interaction structure, and
- a semantic interaction and navigation metaphor.

In the previous steps, methods for calculating similarity values for pairs of tags and semantic arrangements of (sub-)sets of tags were described. The general parameters for tag-based visualization comprise; tag popularity - which is reflected through font size, a certain form of visualization of the tag interrelations based on spatial proximity, e.g. in form of a graph or tree structure. Clearly, more closely located tags imply a higher level of semantic relatedness. The same is true for the topic regions.

Furthermore, an appropriate navigation metaphor for the chosen visualization has to be provided, e.g. dive into the center for exploring different hierarchical levels, i.e. levels with different semantic densities of a semantic or topical region.

A basic reference example for a tag cloud structure would comprise; a compact overview representation of the tag space showing the most general tags, clustering of related tags with high co-occurrence values, appropriate selection and filtering steps for identifying the most representative related tags and a metaphor to link other tags, e.g. radial or hierarchical arrangements, etc.

7.1.3 Semantic Cloud User Interface

The implemented prototype user interface (Semantic Cloud) shown in Figure 7.4 consists of three main parts for data visualization and interaction. The semantic tag cloud as the main interface element on the left and serves to browse the hierarchical semantic clusters. The results section on the



Figure 7.4: UI of Semantic Cloud - general overview cloud.

right shows the results for the current query. The query is composed by selecting one or several tags in the tag cloud interface or entering additional tags manually into the input field. A third section on the bottom part of the user interface contains classic buttons for navigation, such as adjusting the interface, reset, back, etc.

The entry point for semantic navigation via the tag cloud is the overview tag cloud at the top level, which visualizes a balanced representation of the most popular tags clustered according to tag relatedness.

For better distinguishing topics, tag clouds of different topics, i.e. topic clusters are divided spatially and by color in distinct visual semantic regions. The visualization of the semantic tag cloud, i.e. the semantic arrangement of tags is based on the following concepts:

- Varied font size with respect to the popularity of a tag
 - Graph-visualization with force-directed layout⁸⁷ based on the described similarity metric.

Sub-clusters can be viewed by clicking on a magnifier icon which is placed in the middle of each semantic region/cluster – if they exist. Different

⁸⁷A visualization method allowing to display highly similar elements close to each other efficiently and aesthetically pleasing, for example, making edges to be more or less of equal length and minimizing the number of crossing edges, etc.

hierarchical levels can be navigated by starting from the most general tag and browsing into lower levels (Figure 7.5) with a higher semantic density and more specific tags in order to focus a special thematic field, e.g. from food to vegan, cooking, etc. Queries can be composed by either selecting a tag within the respective semantic region of interest and refining via manual tag input.



Figure 7.5: UI of Semantic Cloud - hierarchical exploration.

Selected (clicked on) tags are highlighted in the tag cloud and further appear in the query list on the right left corner of the tag cloud section. Chosen tags can be removed either by clicking on tags within the tag cloud or the “x” button in the list of selected tags. In addition, relocating a specific query tag can be accomplished by using the magnifying glass icon on the right hand side of each tag. This allows users to focus on semantic regions related to manually entered tags without the need to manually browsing the tag hierarchy.

From the selected tags a user query is created implicitly by combining them with the Boolean AND operator. Basically, query selection/composition directly influences the results generation and allows the dynamic removal or replacement of tags leading to immediate feedback for user interactions. Hence, results can be consulted immediately allowing the adjustment or refinement of their queries appropriately. Consequently, changing the focus of a search – hence the semantic region to be browsed – can be achieved by replacing tags through other related tags. A more detailed view of the applicable retrieval techniques, e.g. Boolean and bag-of-words model, etc. can be studied in Baeza Yates and Neto (1999).

Some Technical Details of the UI

The RaVis visualizations library⁸⁸ was used to visualize the hierarchical graph structure in the form of several hierarchically arranged topic clouds with force-directed layout. In this structure tags are represented as nodes and edges connecting up to three related tags. The edges are represented by the proximity of the related tags to the center based on the used semantic similarity metric. All tags of a sub-cluster are additionally connected to one virtual node placed in the cluster center in order to realize a stronger association and compact layout for the tags within a topic region. The virtual nodes are furthermore utilized by clickable icons for requesting detailed views, i.e. more specific sub-clusters of a topic. In addition, clusters are colored using distinct colors. Adjusting font size according to tag distribution is calculated relative to each sub-cluster by applying a logarithmic formula described by Dekoh⁸⁹:

7.1.4 User Study

The prototype interface was evaluated in a user study to assess its benefits compared to traditional user interfaces of folksonomy systems, represented by the delicious social bookmarking user interface as the baseline. 9 participants (2 female and 7 male) between 22 and 33 years old rated the described prototype of the semantic cloud user interface (based on the data set described in Section 7.1.2) and the delicious interface.

For the evaluation, empirical as well as qualitative methods were used in order to verify the results and gain additional evidence for the user's preferences when using both interface approaches. Figure 7.6 illustrates the prior knowledge of the users.

The subjects all had a computer science background being confident in using a computer and a web browser. In general, though all subjects have been familiar with tag clouds, none of them used them to browse data in folksonomy systems for finding relevant content.

1. Tasks-based Evaluation

A task-based user evaluation was used to test the delicious user interface versus the semantic cloud interface simulating an “undetermined browsing” scenario. Looking at the background knowledge of the participants a setup

⁸⁸RaVis: <https://code.google.com/p/birdeye/wiki/RaVis>

⁸⁹<http://blogs.dekoh.com/dev/2007/10/29/choosing-a-good-font-size-variation-algorithm-for-your-tag-cloud>

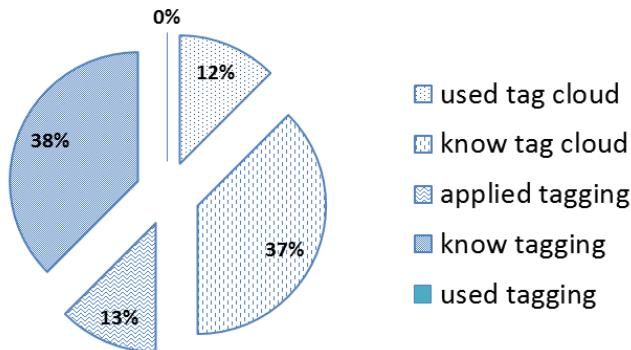


Figure 7.6: Background knowledge of the 9 subjects.

for a “within-subjects testing” was decided, in order to create a basic understanding of existing browsing interfaces, before letting users compare and judge both interface approaches. The reason is, that for a small number of users, differences are more critical (Nielsen, 1993), hence, with the chosen method it can be ensured that knowledge gained in testing one system is transferred to the following test. Choosing a “between subjects” testing was therefore not suitable as it is strongly biased by the participants and their skills and preferences. The following three representative tasks⁹⁰ first had to be solved by using the classic delicious interface and afterwards the semantic cloud:

- Task 1: Look for any website you would find interesting
- Task 2: Look for a website about any interesting “cooking recipe”
- Task 3: Search any website dealing with “music” you find interesting

In order to gain qualitative insights into the user’s ratings of both systems and the usability of both systems, the users were asked to “think aloud” when performing the tasks. In this way, possible interface order effects were also analyzed. Besides that, users were also asked to explain their reasons while rating the interfaces.

2. Questionnaire and Significance Testing

After the tests, each participant had to assess both interfaces by answering four questions (see Appendix C.1) in order to measure the following cor-

⁹⁰These particular tasks have been chosen, as the respective resources and tags were comparably present in both systems.

responding four usability criteria: efficiency, support, intuitive usage, and overall vote on a five point Likert scale. The results of the ratings for each question were analyzed statistically for significance applying the student's t-Test in order to verify the following general hypothesis:

- H1: *The Semantic Cloud user interface concept is a significant enhancement compared to the standard user interface structures of folksonomy systems.*

Consequently, the null hypothesis H0 predicted that the mean rating for both interfaces was equal and differences only due to chance.

7.1.5 Results

For analyzing the answers of the final questionnaire, the mean (μ) and standard deviation (σ) for each question and system was calculated. The overall average scores indicated enhanced support and user experience of the Semantic Cloud interface ($\mu=4.16$, $\sigma=0.825$) compared to Delicious ($\mu=2.94$, $\sigma=0.94$). In order to verify the significance of these results a paired student's t-test was applied. The results are shown in Table 7.2.

| criteria | efficiency (Q1) | support (Q2) | intuitive (Q3) | overall (Q4) |
|----------------|-------------------------|-------------------------|-------------------------|------------------------|
| Delicious | $\mu=3.44, \sigma=1.01$ | $\mu=3.11, \sigma=0.97$ | $\mu=2.89, \sigma=0.78$ | $\mu=2.33, \sigma=1.0$ |
| Semantic Cloud | $\mu=4.11, \sigma=0.33$ | $\mu=4.56, \sigma=0.53$ | $\mu=4.0, \sigma=1.12$ | $\mu=4.0, \sigma=1.32$ |
| p | 0.081 | 0.001 | 0.021 | 0.000 |
| $t_{0.05;8}$ | -2.00 | -4.91 | -2.86 | -5.77 |
| decision | not significant | significant | significant | significant |

Table 7.2: Mean rating, standard deviation and t-Test results.

In the results, the null hypothesis H0 was rejected for a probability p⁹¹ lower than 0.05, which was the case for question Q2, Q3 and Q4. This means that the hypothesis H1 – stating significant enhancement of the Semantic Cloud interface compared to delicious – is true for these cases. Only in case of question Q1 was the null hypothesis H0 not rejected, thus, the differences are not significant.

More expressive explanations of why the systems were rated in a particular way could be inferred from the comments of participants. Basically, both

⁹¹Consider, that p is the probability of t being equal or greater than the observed value $t_{0.05;8}$: $p = p(t \geq t_{0.05;8})$. Hence, p is the probability that H0 is valid or not. A t value is significant (H0 is rejected) in case of p being lower than the alpha level.

interfaces were assessed easy to understand and no major problems occurred during testing. However, the thinking aloud protocol revealed limitations of classic interfaces as expected. Users criticized the limited number of related tags which forced them to enter several tags manually in order to refine their queries. Besides that, it turned out that in most cases the delicious tag cloud in combination with following related tags does not help users to navigate towards a rough search goal. Users rather tend to fall back into classic tag or keyword-like search. A few users stated that they would prefer to use Google rather than a folksonomy-based search system, i.e. they were biased by Google or keyword search. Nevertheless, the experiments and user comments clearly depict the benefits of a semantically clustered tag space and hierarchical browsing of content in a folksonomy system, which are listed as follows:

- Combining tags from different clusters and sections and finding other interesting tags was easy and intuitive
- Semantic arrangement of tags and hierarchical exploration, i.e. breakdown of topic was useful and logical
- The interface was more supportive while providing more tags and respective related tags to select
- Single interaction structure of a semantic tag cloud generated from related tags helpful to edit and refine queries at any time

Users also stated that the semantic interface was more visually attractive and transparent due to spatial semantic arrangement and the usage of colors for different topic fields.

7.1.6 Conclusion

Interfaces of folksonomy retrieval systems rarely exploit semantics from user tags, although tags reflect multiple aspects and contain implicit semantic relations which could be exploited for improving search, information exploration and interaction with search results.

In the described case study, a hierarchical semantic representation of the tag space, obtained via semantic analysis and clustering, was employed in a visual retrieval interface. The semantic tag space was visualized in form of a multi-topic semantic tag cloud, which can be explored hierarchically at distinct semantic levels of density - from general to specific.

A user study showed that users tend to prefer semantic representation and interaction over widely implemented traditional interfaces based on simple tag clouds showing popular tags or other arrangements of tags. The semantic tag cloud interface showed significant results for support, intuitiveness and for overall interaction. Though not statistically significant, the semantic user interface also obtained higher average scores for interaction efficiency.

These results and the user feedback indicate that the user's browsing experience can be enhanced by introducing a semantic dimension to interaction in a folksonomy retrieval interface. The use of semantic interaction metaphors while designing such interfaces can be seen as a core contribution, enabling users to inter-connect and combine query interactions at different semantic levels of the visual user interface. As an example, selecting tags from general overview tag clouds can be coupled with interactions for selecting more specific tags from more specific (lower) semantic levels of co-occurring (related) tags. Consequently, it is not sufficient to map the underlying (clustered) semantic information structures to visual user artifacts, e.g. tag cloud, one to one without thinking about the right semantic interaction metaphor of how to exploit and interact with these kind of structures efficiently. Hence, increasing interaction quality can be regarded as being just as important as retrieval efficiency, particularly in scenarios for browsing and exploring information spaces.

The user study also revealed that in future research cross-topic exploration needs to be enhanced, which would allow the exploration of several thematic fields at the same time, e.g. travel and photography, without the need to explore two semantic clouds in sequential order or by the manual input of tags. As a result, the use of non-exclusive clustering is reasonable in order to cluster particular tags into several clusters. Therewith homonymous tags could be displayed in several thematic fields and used in all their context, also allowing the handling of also very general tags e.g. blog, photo. Non-exclusive clustering would also cover fuzzy cluster borders where tags belonging to different topics could be placed.

Furthermore, users that participated in the evaluation suggested several ideas for improvement, ranging from small extensions, e.g. additional information on results, towards larger challenges such as including a more extensive set of tags "behind the scenes" by analyzing the semantic context of the displayed tags more deeply, e.g. over synonymous relations.

7.2 Case Study: Semantic Browsing in 3D

3D visualizations of collections of information such as search results have been explored for a while. In the case of visual retrieval interfaces for the Social Web, the approaches are widely based on tag-cloud or graph-based visualization for searching and exploring user generated contents.

In the first case study a semantic dimension to search and exploration of data in folksonomy systems was introduced by means of a semantic tag-cloud approach in 2D. The second case study differs in that semantic exploration and interaction is based on emergent semantics extracted from user tags that are exploited in 3D visualization and for navigation of the retrieved results in 3D space.

This section will investigate how emergent semantics – which are extracted applying an equivalent processing pipeline as in the first case study – can be exploited for enabling semantic interaction and navigation in 3D for exploring tagged information spaces? Which (widely used) state of the art visualization and navigation metaphors are applicable therefore? And finally, how does such a 3D semantic browsing approach compare to traditional 2D folksonomy retrieval interfaces?

Similar to the semantic cloud approach described in the previous case study in Section 7.1, related tags are extracted from the folksonomy hypergraph and exploited for the semantic exploration of the corresponding user-provided contents. In order to interact with retrieved results more efficiently, related tags as well as previews of the associated web documents are integrated seamlessly, aiming at providing semantic arrangements as well as a visual impression of the results pages with more details on demand in 3D. As in 3D visualizations, additional degrees of freedom can be exploited, different visualization (wall, carousel, corridor) and semantic interaction metaphors are employed. Interaction and navigation in 3D space based on direct manipulation is supported by means of actions such as zooming, rotating, etc. Semantic relatedness of semantically clustered resources is reflected as proximity in space along distinct spatial dimensions, e.g. z-axis.

The research presented herein shows the benefits of using emergent semantics for exploring folksonomic data combined with 3D interaction Döring et al. (2012), but also reveals usability problems in 3D retrieval and semantic exploration in a user study. In subsequent sections it will be shown that 3D semantic browsing and navigation also necessitates a natural and intuitive semantic interaction and navigation metaphor, which could be realized

employing visual interaction by means of first person shooter perspectives and different navigation metaphors, e.g. vehicle, known from 3D computer games. Again the delicious user interface serves as a state of the art 2D baseline for the evaluation.

7.2.1 State of the Art

Tag clouds and graph-based visualizations of web retrieval interfaces widely make use of two spatial dimensions when used as an interaction structure in visual retrieval interfaces, e.g. for query composition. The returned results of the browsing or retrieval interactions are presented in the form of a list of hyperlinks to the contents plus extracted snippets. Hence, the users are able to interact with the result list along the vertical axis using only one dimension. Besides list and 2D visualizations, some interfaces for exploring data in 3D (data browsers) have been presented in Section 3.4.3. In contrast to the these user interfaces, the work presented herein enhances and combines some of the techniques and interaction metaphors thus far researched for the semantic exploration of folksonomy data in 3D browsing interfaces, which was not yet implemented or researched systematically. Relevant questions deal with semantically clustered 3D views of the information space or retrieved subsets. Popular browser-based interfaces such as Cooliris or SpaceTime 3D (described in Section 3.4.3) focus on the visualization and provide no means for semantic interaction with the results. Besides this, the research focuses on the questions of whether 3D views of the tagged information space can be exploited more efficiently with new interaction metaphors and input modalities such as touch, which will be researched in the third case study in Section 7.3.

7.2.2 Approach

The prerequisite is a sequence of tasks performed in order to prepare the respective tagged datasets retrieved from the folksonomy system. Task 1 and 2 deal with the pre-processing, semantic analysis of tags for extracting related tags through co-occurrence analysis, followed by semantic aggregation (Task 3) through clustering of related tags and associated resources (bookmarked web pages). These tasks serve as a basis for the 3D visualization of the semantically clustered results and the user interaction and navigation in 3D space (Task 4).

Task 1: Data Acquisition and Pre-processing

Delicious is a widely used broad folksonomy system allowing the gathering of appropriate data sets for tag-based analysis. For the prototype implementation and the user study, 26 popular tags with 2400 associated bookmarked URLs having 2947 further associated tags have been extracted. Besides that, a preview image was generated for each bookmark in the extracted corpus utilizing a web-based thumbnail service⁹². As tags can appear in different variations, e.g. singular-plural, they are normalized applying the Levenshtein distance. Alternatively, other previously described methods for tag normalization and filtering could be applied to improve the overall retrieval results.

Task 2: Semantic Analysis and Aggregation

In the following, the process of calculating the similarity between pairs of bookmarked documents and pairs of tags is described. While similarity values for pairs of bookmark vectors are needed for clustering and visualizing the result documents, semantic relatedness of tags is computed for clustering and visualizing the tags of the respective (related) documents.

In order to obtain the similarity values for the stored document corpus, the cosine coefficient applied on tag or document vectors – that was also used in the previous case study – is calculated prior to the clustering process via the following procedure:

1. Create bookmark vectors of size n (which corresponds to the number of distinct tags) for each bookmark b_i with their absolute tag frequency counts f_{ij} for the tags t_1, t_2, \dots, t_n as shown in matrix A:

| A | t_1 | t_2 | ... | t_n |
|-------|----------|----------|-----|----------|
| b_1 | f_{11} | f_{12} | ... | f_{1n} |
| b_2 | f_{21} | f_{22} | ... | f_{2n} |
| ... | ... | ... | ... | ... |
| b_m | f_{m1} | f_{m2} | ... | f_{mn} |

2. Calculate two co-occurrence matrices by applying the cosine similarity measure to pairs of tags (t_i, t_j) and bookmarks (b_s, b_t) applying the formula introduced in Definition 2.7. The weights u_{ij} for a pair of tags as shown in matrix B is calculated from the matrix A by applying the

⁹²Thumbnail-service: <http://snapcasa.com>

cosine similarity metric to the corresponding column vectors of the regarded two tags.

| B | t_1 | t_2 | ... | t_n |
|-------|----------|----------|-----|----------|
| t_1 | u_{11} | u_{12} | ... | u_{1n} |
| t_2 | u_{21} | u_{22} | ... | u_{2n} |
| ... | ... | ... | ... | ... |
| t_n | u_{n1} | u_{n1} | ... | u_{nn} |

The similarities v_{st} for the bookmark pairs shown in matrix C is calculated the same way by applying the cosine similarity to the row vectors.

| C | b_1 | b_2 | ... | b_m |
|-------|----------|----------|-----|----------|
| b_1 | v_{11} | v_{12} | ... | v_{1m} |
| b_2 | v_{21} | v_{22} | ... | v_{2m} |
| ... | ... | ... | ... | ... |
| b_m | v_{m1} | v_{m1} | ... | v_{mm} |

3. Apply the star clustering algorithm to the both co-occurrence matrices.

Task 3: Semantic Aggregation (Clustering)

The Star Algorithm (pseudo-code outlined in Algorithm 5) is applied to the calculated co-occurrence matrices of the tags and the bookmarks.

Algorithm 5 Pseudocode of Star Clustering (Kowalski, 1997)

```

1: for  $i = 1$  to N do
2:   if ( $e_i$  NOT IN cluster) then
3:     put element into a new cluster
4:   end if
5:   for  $j = i + 1$  to N do
6:     if ( $e_j$  SIMILAR TO first element  $e_i$  in cluster) then
7:       put  $e_j$  into current cluster
8:     end if
9:   end for
10: end for

```

The clustering algorithm used was chosen as a compromise, allowing overlapping clusters, while still preserving similarity within the individual clusters. Furthermore, the underlying cluster representation had to be tailored to a radial tag cloud structure for visualizing the three tags that were the most related from each cluster in order to fit the given screen size. In

this way, all relevant web pages can be made visible in the first hierarchy.

Task 4: Query Processing

Prior to visualizing and interacting with search results, a given user query needs to be executed in order to return the most relevant documents from the underlying document corpus. Therefore, the user query vector comprising a list of search terms is compared to the tag vectors of the stored documents. Bao et al. (2007) describe several ways to compare the similarity between a search query and a tag list assigned to a bookmarked document in order to calculate a ranking for a document collection. The following simple formula can be used to calculate the similarity of a query $q=\{q_1, \dots, q_k\}$ comprising k terms and a bookmarked web page p with Annotations $A(p)=\{<a_1, f_1>, \dots, <a_n, f_n>\}$ with terms a_i and their frequency counts f_i .

$$\text{sim}(q, p) = \frac{t}{b} \cdot df$$

The calculation is based on shared terms $q \cap A(p)$ – taking into account the popularity of a bookmark by considering its document frequency. In the formula above, t is the sum of the tag frequency counts ($t=\sum_{i=1}^s f_i$) of the tags that co-occur in query q and in bookmark p , while b is the total frequency count of a bookmark ($b=\sum_{i=1}^n f_i$) and df the document frequency, i.e. how often p was bookmarked in total by users of the social bookmarking system.

Task 5: Visual Interaction and Semantic Exploration of Results in 3D

In contrast to the previously described semantic tag cloud approach, besides visualizing and interacting with a semantic representation of the clustered tag space, e.g. for query composition, a (2D) representation of web documents (document space) of search results is projected onto the 3D space. Therefore, tags as well as the documents are clustered on the basis of the co-occurrence analysis as described before.

In the developed 3D user interface (Figure 7.7), the search results are first arranged spatially using the radial tag cloud metaphor represented as an overview. Due to the available standard screen size and processing power and in order to allow fluid visual interaction, the result set that was to be clustered and visualized in 3D was restricted to the top 100 most relevant bookmarks for a given user query based on the similarity calculation described in Task 3.

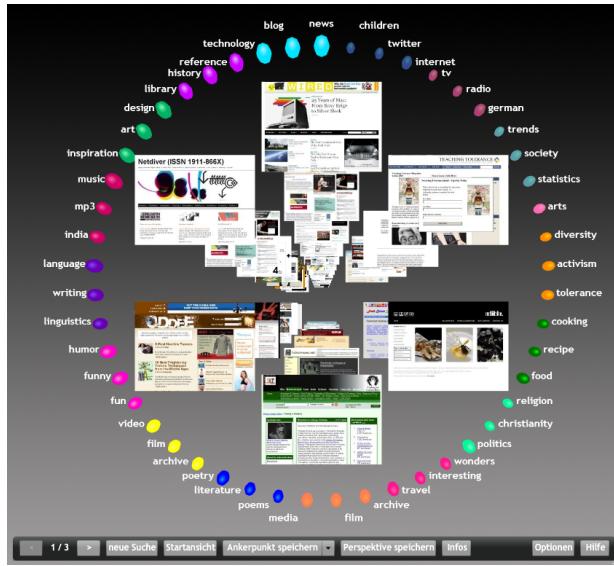


Figure 7.7: The radial semantic tag cloud with the associated document clusters in the inner circle.

Overview Visualization:

The overview visualization serves to get a first impression of the search results by featuring a special tag cloud layout based on tag relatedness, frequencies and clustering. Associated document clusters are placed relative to tags into the inner part of a radially arranged tag cloud structure. After selecting tags that suit their information needs (query composition) users are able to focus on particular related and relevant web pages by navigating to corresponding page clusters in 3D space. For the representation, besides semantic relevance derived from the user tags, statistical relevance is exploited as well.

The navigation is based on a first person shooter perspective and allows one to “dive” into the cluster space manipulating spatial parameters (translation, scaling and rotation) of the 3D space. Clustering based on tag semantics allows the browsing of semantically related thematic fields.

The following sub-sections provide a more detailed explanation of the 3D-visualization and interaction.

I. Forming the Semantic Tag Cloud

After the preceding steps of data preparation for obtaining the clustered representation of tags and resources (Task 1-3), the most frequently occurring three tags from each cluster are arranged in a radial form of a tag

cloud for the overview visualization. Three tags have been chosen to be representative enough to describe the clusters as the available space on the circle is restricted. The related tags from the search results are placed at the edge of the circle and the preview images of the corresponding web pages in the inner circle. The circle metaphor allows the assignment of tags to web pages, because the sites are arranged in the vicinity of the tag cluster, which describes it best. Each tag is placed near a corresponding tag sphere with suitable size and color depending on the cluster (color) and relevance (size). The color groups similar tags that belong together, i.e. tags from the same cluster have spheres with the same color. The size of the sphere shows the relevance of the tag in the entire search result, i.e. the larger the diameter of the sphere, the more often the tag is found in the search result. The diameter of the spheres arranged in a circle is calculated as follows (Seifert et al., 2008):

$$d_i = \frac{d_{max} \cdot \log(x_i - x_{min})}{\log(x_{max} - x_{min})},$$

where d_{max} is the maximum diameter of sphere, x_i the tag frequency, x_{max} the frequency of the tag that occurs most in the search results and x_{min} the frequency of the tag that occurs least. The logarithm is used to normalize the sphere size for different tag frequencies.

Before clustering, tags are sorted in descending order with respect to their frequency. The most frequent tags are placed on top of the circle, while the others with descending frequencies are arranged radially counter-clockwise. This makes sense as evaluations on tag cloud layouts analyzing eye-tracking data showed (Lohmann et al., 2009) that the users tend to focus on the top left area of the screen, independent of the layout type used.

II. 3D Visualization of Web Resources

Tag clusters provide a fast and associative thematic overview utilizing the most occurring related tags, enabling users to modify or refine their query more precisely. Furthermore, users are able to identify similar tags and corresponding relevant page clusters and directly navigate to a group of related bookmarks and web pages with similar topics.

The three-dimensional space has the advantage that more information can be represented by the additional dimension. Moreover, enhanced interaction possibilities arise as users are able to manipulate more degrees of freedom. The x and y axes are used to position page clusters that contain

“semantically” related or similar web pages, while the z axis is utilized to reflect the relevance ranking of the regarded web pages along the depth-axis.

The corresponding web page cluster thumbnails are placed in the inner circle in form of a stack enfolded along the depth axis. As this space is restricted to only a few clusters, parallel screens are utilized to place additional clusters the users can interact with.

In the middle of the circle no cluster is arranged so that the user can fly through the circle “into” the space to take a closer look at the individual websites in the cluster by navigating left- or right, up and down and changing perspectives in 3D.

A website cluster is placed near the tag or tag-cluster that most accurately represents the corresponding document cluster in conformance with the star clustering method, where the first element (“the star”) in a cluster is the one which is most representative.

III. Semantic Interaction and Navigation

The semantic 3D browser allows us the browsing and exploration of the tag space via the described radial semantic tag cloud and focus on related documents by navigating in 3D space. The semantic tag cloud can also be utilized to compose a user query.



Figure 7.8: Changing Perspective in the radial Semantic Tag Cloud View.

The interaction can be performed via mouse and keyboard, while it is also viable to use multi-touch input or a Wii-mode controller to navigate in 3D space. The clustering-view allows free navigation from the first person shooter perspective, i.e. moving and modifying view perspective by manipulating the available DOF (Figure 7.8).

For more detailed interactions, the clusters can be viewed utilizing three different spatial arrangements; wall, carousel and corridor (Figure 7.9). The corresponding original web pages can be viewed in the integrated browser

utilizing tabbed-browsing.

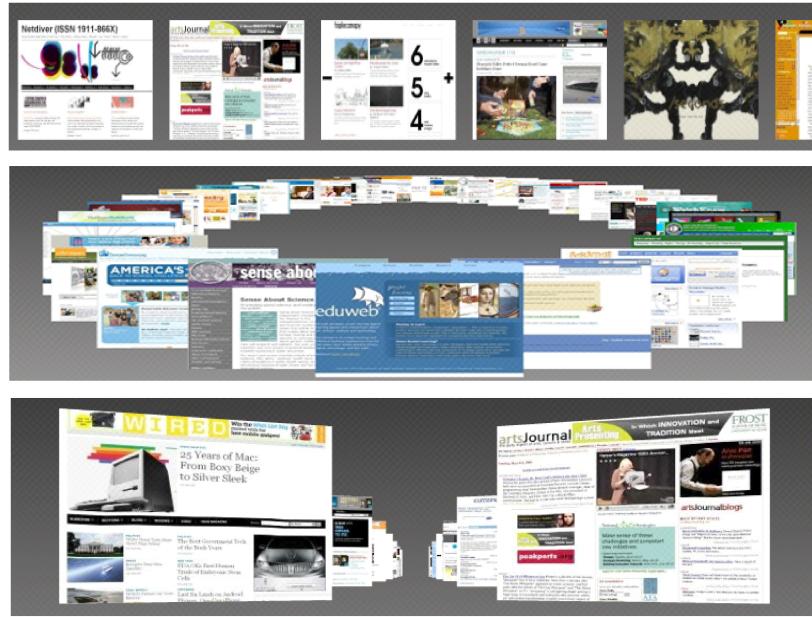


Figure 7.9: Spatial Metaphors for 3D-visualization of document clusters.

The page clusters can be sorted according to tag frequency, relevance ranking, date and the number of users that annotated a particular bookmark. The tag frequency reflects the relative frequency of the tags per document, which considers all search tags. The relevance ranking additionally takes into account the number of times a bookmark was stored by the users, i.e. takes into account the popularity of a bookmark.

Besides the pre-calculated visualization of the tag clusters, it is possible to select one or more tags from the tag cloud for creating and visualizing a cluster of semantically related web pages dynamically. In the case of selecting several tags, tag co-occurrence is used instead of the relative tag frequency.

Spatial Metaphors for Visualizing Document Clusters:

The user can explore the search results in two different ways; via the tags or via the website clusters. In the main clustering view the user can move into the virtual 3D space and manipulate the camera position and perspective in the “first-person shooter perspective”.

The user has the opportunity to either delete non-relevant tags or irrelevant web pages from the search result. As already mentioned, the user can take a closer look at the website clusters by moving “into” the space

through the middle of the circle. Moreover, each cluster can be consulted in detail in three additional spatial arrangements - in the wall-, in the carousel- or in the corridor view.

In all views an individual web page can be moved and placed in a different place. Thus, the user has the possibility of putting aside a web page for subsequent actions and keep it for later use.

In addition, each website can be accessed and viewed in the built-in HTML web browser application. Herewith, several websites can be viewed on demand using tabs.

Thus, the user is able to switch between the tabs and view all previously accessed web pages or close single tabs. If the browser is closed, the user returns to the 3D view from which he/she has started the browser.

Input Modalities for Navigation:

The control concept is based on interactions with mouse and keyboard. The mouse itself is not sufficient to cover all interaction possibilities of a three-dimensional space. For this reason the keyboard is used optionally if needed. It will be shown later that complex interactions in 3D space also necessitate understandable and intuitive interaction and navigation techniques.

7.2.3 User Study

In order to evaluate the 3D semantic browsing approach for folksonomy data, a user study was conducted. Again, the delicious social bookmarking retrieval interface served as a baseline. The main focus was to analyze the impact of semantic relevance for the user's browsing experience, investigate the interaction behavior with different visual and navigation metaphors in 3D, and how the 3D approach for semantic exploration of data in a folksonomy system compares to traditional (non-semantic) 2D interfaces represented by the delicious system. Semantic relevance was reflected in the semantic space browser by means of semantically arranged and clustered related tags and associated documents in the 3D information space.

Other interesting questions were whether users with 3D gaming experience are more satisfied than ordinary web users, and how users with a good social web knowledge responded compared to users that only had no or less experience with social networking applications and folksonomy systems.

The main parameters tested were those of intuitiveness, ease of use, efficiency, satisfaction with the browsing experience based on standardized questionnaires for capturing subjective impressions.

Evaluation Set-Up

A total of 29 persons invited via e-mail participated in the web experiments for interacting either with the delicious system or the semantic 3D browser application, ranging in age from 21 to 33 years (19 male, 10 female). 12 persons were classified as experts, and 17 as novices depending on their background knowledge in the case that they achieved more than fifty percent of the points from the first part of the questionnaire dealing with background knowledge in related disciplines such as 3D navigation, social tagging systems, etc. Additional analysis comprised the users with a lot of 3D and social networking experience compared to the corresponding non-experienced users.

| |
|--|
| Task 1: |
| a) Search with tag “visualization” |
| b) Look at results |
| c) Find page titled “flickrvision” |
| Task 2: |
| a) Use “explore tags” and type in “education” |
| b) Find pages you’re interested |
| c) Use the related tags to restrict the results |
| Task 3: |
| <ul style="list-style-type: none"> • Use the “tag cloud” to find page(s) you’re interested in |

Table 7.3: Tasks for Delicious.

The general evaluation procedure comprised the following steps:

1. Answer pre-questionnaire (person, background knowledge, etc.)
2. Watch (screencast) and read introduction to delicious, then solve the 3 given tasks (Table 7.3) using delicious
3. Watch (screencast) and read introduction to semantic space browser, then solve the 3 given tasks (Table 7.4) using the semantic space

browser

| |
|--|
| Task 1: |
| a) Search with tag “visualization” |
| b) Look at the “radial semantic tag cloud” to get an idea about available topics |
| c) Move mouse to particular pages to see which topics they belong to |
| d) Study the result pages more detailed by moving closer to a cluster and looking at individual pages. Use the 3D interaction and navigation capabilities of the browser |
| e) Return to the start screen and use the 3D views to find a page named “GPS visualizer” |
| Task 2: |
| a) Search with tag “education” and find interesting pages |
| b) Restrict the results using the related tags from the “radial semantic tag cloud” |
| Task 3: |
| <ul style="list-style-type: none"> • Find a website you’re interested in by using the 3D browser’s interaction and navigation capabilities |

Table 7.4: Tasks for the Semantic 3D Browser.

4. Answer questions for comparing both systems

The evaluation was assessed by assigning each individual participant to one of two groups. Group A first tested delicious (DEL), then the semantic space browser (SSB) application. Group B first used the semantic browser, then the delicious system. The test users were randomly assigned to one of the groups, while it was ensured that experts and novice users were distributed equally between the groups. Both groups together formed the third overall counterbalanced group C. The answers of both groups were recorded and stored together with starting time, end time and information about canceled tests.

SUS and QUIS

The SUS (Brooke, 1996) and QUIS (Harper and Norman, 1993) questionnaires utilized in this user study were based on a 5-point Likert scale and have been applied in various usability experiments for testing overall system usability (SUS) and specific user interactions (QUIS). The QUIS-section of the questionnaire contained well-suited questions (see example in Table 7.5) for testing the aforementioned parameters for the categories of browsing, visualization and 2D/3D interaction.

For SUS, scores between 0-4 were assigned for each question by the users. For positive formulated questions the score is reduced by one, while for negative questions the final score is 5 minus the assigned score. In SUS the final score is then calculated by multiplying the sum of the individual scores by 2.5, resulting in a score between 0 and 100. For QUIS, scores between 0 to 4 could be assigned for each question, while a higher score means a better vote. In order to aggregate scores for a category, the individual votes are summed up and averaged by the total number of questions in a category.

| Question | Parameter | 0 | 1 | 2 | 3 | 4 |
|--------------------------------|-----------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Navigation along z-axis | intuitiveness | <input type="radio"/> |
| | ease of use | <input type="radio"/> |
| | efficiency | <input type="radio"/> |
| General Navigation in 3D space | ease of use | <input type="radio"/> |
| Visibility of Thumbnails | quality | <input type="radio"/> |
| Changing 3D Views | efficiency | <input type="radio"/> |
| Additional Views | supportiveness | <input type="radio"/> |
| | efficiency | <input type="radio"/> |
| | intuitiveness | <input type="radio"/> |
| Effort to find information | time | <input type="radio"/> |
| | number of steps | <input type="radio"/> |

Table 7.5: Excerpt of the 3D section in the QUIS questionnaire form using a 5-point Likert scale.

The interaction category was evaluated based on the *time it took to the find information* within the assigned search task and the *number of steps* (interactions) to achieve this goal. For the 3D-browsing and interactions and for the different semantic UI artifacts, additional questions for measuring *intuitiveness*, *ease of use* and (*navigation*) *efficiency*, etc. were answered by users. In the last part of the questionnaire a direct comparison of both systems was assessed, focusing on navigation efficiency and search effort.

7.2.4 Results

The overall group of 28 participants (1 outlier removed) attested a SUS score of 66.88 for delicious and 68.94 for the semantic 3D browser.

Tullis and Albert (2008) report, that a score under 60% can be regarded as relatively poor and a result over 80% a good one. Consequently, both systems are between both best ranges indicating further necessary improvements.

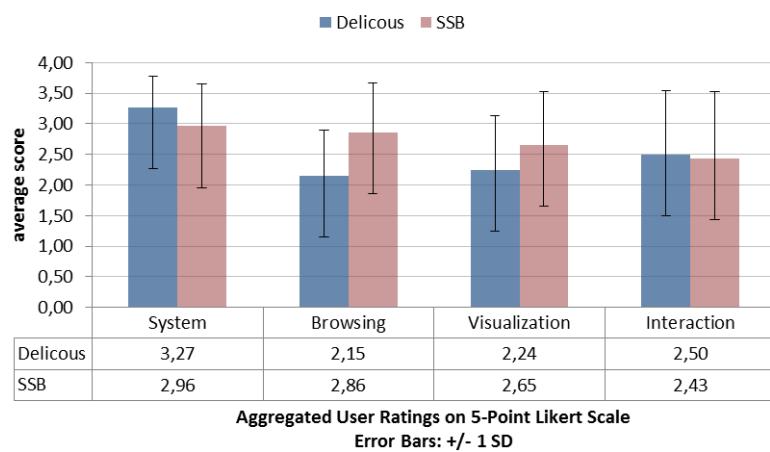


Figure 7.10: Aggregated mean scores for the four tested categories.

Furthermore, both systems were compared with respect to the categories: system, browsing experience, visualization and interaction by calculating mean (μ) and standard deviation (σ) for each category and system. Figure 7.10 shows the aggregated average scores for the tested categories normalized by the number of relevant questions in each category.

The results attest a general plus for delicious in the systems and interaction category, while the semantic 3D browser gets higher average scores for the browsing experience and the visualization used. The results of the system category are not surprising as delicious is a folksonomy system with a more mature UI than the newly developed 3D browser prototype.

In order to assess whether the ratings for the two interface approaches differed significantly with respect to the research focus of enhancing semantic exploration of folksonomic data, a paired two-tailed t-Test with an alpha level of 0.05 was first applied for the categories of browsing experience, visualization and interaction.

In addition, the impact of 3D on the user ratings was analyzed conducting additional tests. In case of a significant result, a subsequent one-tailed

t-Test served to confirm or reject the preferences for the one or the other interface. Thus, the alternative hypothesis H1 for that case was that “the Semantic Space Browser provides enhancements with respect to the regarded category”, while the null hypothesis says that the difference in the means of a regarded category either way was not significant, but due to chance.

Table 7.6 gives an overview about the results (mean, std. deviation and p value) for the counterbalanced joint test group for the categories of browsing experience, visualization and the interaction related questions with and without including aspects related to semantic interaction and 3D for the Semantic Space Browser. The p results of the corresponding one-tailed t-test are shown in the same row in round brackets. The values allow (confirm or reject) the null hypothesis to be judged. In all cases where it is rejected, the alternative hypothesis stating that significant enhancements are provided by the semantic space browser user interface with respect to a regarded category or test parameter compared to the baseline interface is regarded as valid.

| - | Browsing | Visualization | Interaction | SI in 3D |
|-----|-------------------------|-------------------------|-------------------------|-------------------------|
| DEL | $\mu=2.15, \sigma=0.75$ | $\mu=2.24, \sigma=0.88$ | $\mu=2.50, \sigma=1.06$ | $\mu=2.50, \sigma=1.06$ |
| SSB | $\mu=2.86, \sigma=0.81$ | $\mu=2.60, \sigma=0.86$ | $\mu=2.43, \sigma=1.09$ | $\mu=2.84, \sigma=0.68$ |
| p | 0.000174 (<.0001) | 0.046168 (0.023084) | 0.758941 (0.3794705) | 0.100624 (0.05912) |
| - | -4.35 | -2.09 | +0.31 | -1.7 |
| - | significant | significant | not significant | not significant |

Table 7.6: Mean rating, standard deviation and t-Test results.

The results show that the difference in the means for the categories browsing and visualization are statistically significant, while the results of the interaction category are not significant and need further analysis. As the “SI in 3D” category has a relative low p value – although the significance level of 5% was exceeded – it is worthwhile looking to the individual questions related to the 3D visualization metaphors and its semantic user interface artifacts for semantic interaction in more detail.

Semantic Interaction in 3D

Figure 7.11 shows the mean results focusing on the questions related to 3D aspects and semantic user interface artifacts of the semantic space browser. The delicious user interface artifacts comprise related tags, a tag cloud

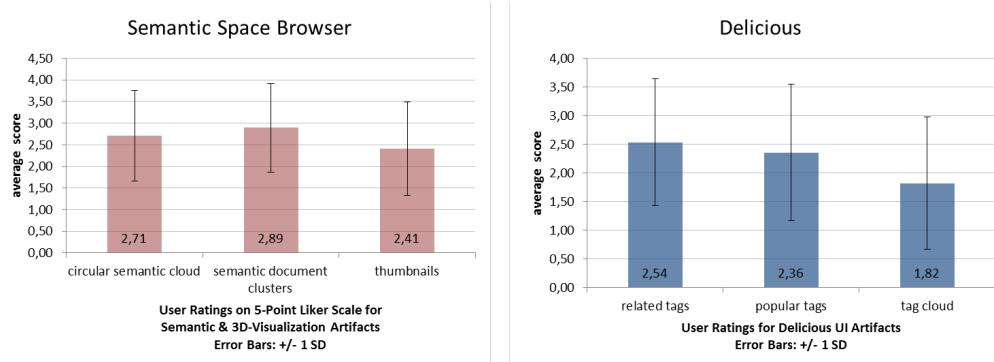


Figure 7.11: Aggregated mean scores for 3D Visualization and Semantics.

visualization and the most popular tags. The semantic 3D browser interface consists of the radial semantic tag cloud, the semantic document clusters and the thumbnail visualization of the bookmarked websites.

One clear observation is that the semantic space browser achieves higher mean scores for all of the aforementioned aspects. In particular the semantic tag cloud and associated semantic clusters and their visualization in 3D are regarded as being helpful for accomplishing the assigned search tasks.

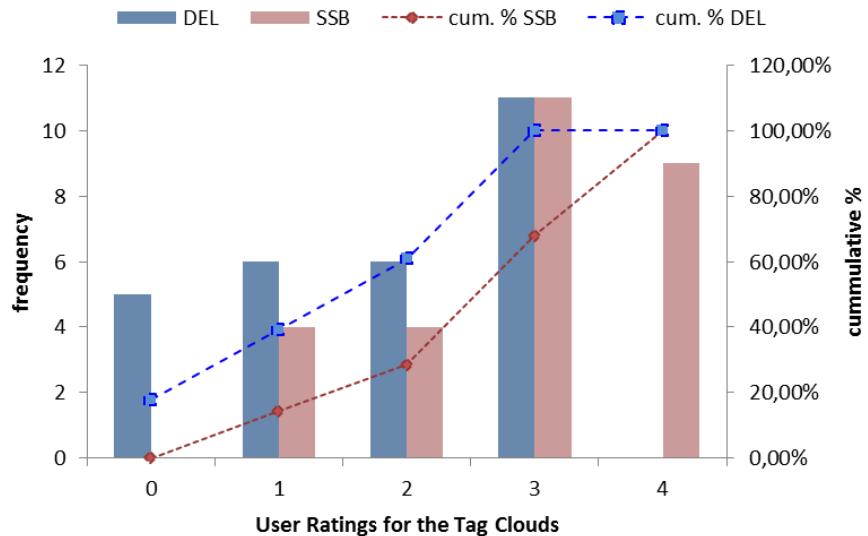


Figure 7.12: Distribution of user ratings comparing both tag clouds.

Moreover, the distribution of user ratings (Figure 7.12) comparing both tag clouds show that the users gave no zero value to the semantic tag cloud, while they voted 9 times with the highest score of 4. In contrast, the delicious tag cloud never obtained the highest while scoring no points 5 times.

Looking at the results of delicious, particularly the tag cloud, achieves relatively poor results. The difference in the mean ratings are significant in favor of the semantic 3D browser, as it is confirmed by a one-tailed paired t-Test ($t_{0.05;27}=-3.94$, $p=0.0002595$) with an alpha level of 0.05.

Figure 7.13 shows the 95% confidence intervals (CI) for the means of the interaction relevant questions for the semantic 3D browser with and without aspects related to 3D interaction and semantic interaction compared with delicious. While the basic interaction category shows no relevant differences for both systems, adding 3D and SI features depict clearly significant differences. Qualitative answers from the user comments and interviews, which will be reported later, support these findings.

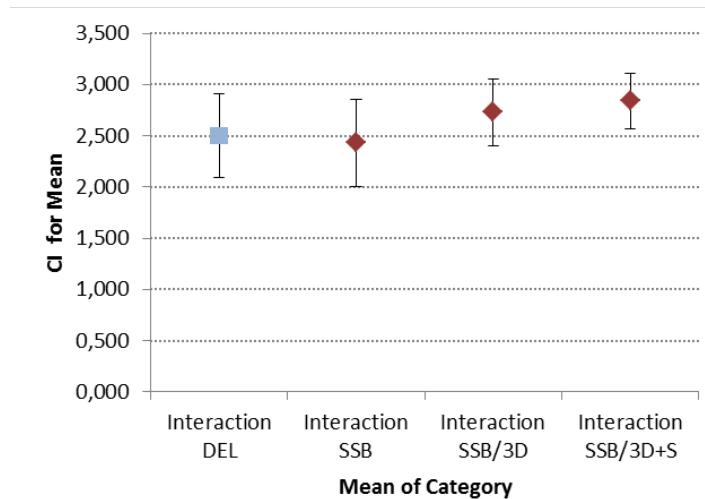


Figure 7.13: CI for the different interaction categories compared.

Navigation in 3D

A more detailed look at the 3D-specific questions show, that the 3D navigation along the z-axis to reach relevant thumbnails was rated above average for the parameters intuitiveness ($\mu=2.82$), ease of use ($\mu=3.04$), efficiency ($\mu=2.72$) and overall navigation in 3D space ($\mu=2.97$).

Changing the cluster views was regarded as easy ($\mu=3.25$). The different cluster views (wall, carousel, corridor) obtained good ratings for the tested parameters support ($\mu=3.12$), efficiency ($\mu=2.79$) and intuitiveness ($\mu=3.04$).

Comparison Questions: DEL vs. SSB

Besides analyzing individual categories, the users were asked a set of questions for comparing both interface approaches directly.

The analysis of *navigation efficiency* – following links in delicious search results and using the scrollbar for selecting them from the lists – compared to the 3D navigation for finding relevant search results, attested a mean score of 2.37 ($\sigma=0.69$) for delicious and a score of 2.56 ($\sigma=0.80$) for the semantic 3D browser. The search effort was rated 2.48 ($\sigma=1.01$) for delicious and 2.52 ($\sigma=0.94$) for the 3D semantic browser.

A paired t-Test ($t_{0.05;27}=-4.09$, $p=0.0002$) with an alpha level of 0.05 showed significant results for the overall comparison questions related to search efficiency, search effort and entertainment. Looking at the parameter of “fun”, the 3D visualization and semantic interaction achieved a mean value $\mu=3.37$ ($\sigma=0.79$). The fun aspect was rated with a mean of 1.37 ($\sigma=0.79$) very low for delicious.

Regarding the 3D interaction, the main difficulty that was unveiled during the evaluation was the unfamiliar free navigation in the 3D space for controlling the available 6DOF. Although it can't be clarified completely with the obtained results, unfamiliarity with the free navigation in 3D and the effects of the more complicated visualizations might have had negative effects on search effort and navigation, which needs further analysis focusing on the interaction parameters more deeply. While expert 3D users had good control over 3D-based navigation, a more suitable interaction metaphor and constrained navigation was desired by users.

Comments also revealed that the complexity of the used cluster visualizations (wall, carousel, corridor) had a certain impact on interaction efficiency and user experience.

User Feedback

Users commented that the preview of web pages and the semantic clustering overviews were helpful for orientation and a more efficient search. They also proposed different clustering methods, e.g. a hierarchical one.

Particularly when being asked about the radial semantic tag cloud and the similarity-based semantic arrangement of the search results based on thumbnail visualization, users gave a positive feedback. Despite this, comment was also made that the quality of the thumbnails needs improvement.

Critical comments were related to navigating in 3D space with the implemented free navigation, leading to situations where users lost orientation

while using the wall view or apply incorrect rotations, for example. Some users suggested using variants of the views with more rows (wall) and removing the redundancy (corridor and initial semantic cluster view seem similar). One user questioned the general applicability of 3D as many people are not able to deal with 3D projected on a 2D screen. Experiences in 3D games do not confirm this point of view, while the navigation support and the input method have crucial impact on usability of the navigation metaphor (see Section 7.3).

The users also stated that they liked the first person shooter perspective and the semantic cluster visualizations. The overall positive feedback from gamers and 3D experienced users and additional user answers related to the used input method (using keyboard and mouse in combination inefficient) confirmed this. It must also be stated, that a certain learning effect can't be neglected looking at the results of the users that had experience with 3D from games, etc.

Additional critical comments on delicious can be summarized as follows; delicious was attested to contain too much textual interaction, while missing feasible support in the provided tag cloud due to absent related tags, no possibility to combine their own tags with tags from the tag cloud and generally less possibilities for user refinement based on tag semantics. The provided related tags were generally regarded as being helpful and likewise, the popular tags but not sufficiently so.

7.2.5 Conclusion

This case study investigated an approach for exploring data in a folksonomy retrieval system exploiting related tags utilized in 3D visualizations of retrieved results. The results showed that semantic interaction in 3D visualizations of search results is a viable approach, which was confirmed with experiments for assessing the user's browsing experience, 3D interaction and visualization preferences. The experiments also revealed some usability problems in 3D exploration of the results, in particular for users unexperienced with navigation in 3D space.

Nevertheless, a more detailed investigation of 3D-relevant interaction parameters, showed the potential for improving interaction for semantic exploration in large 3D information spaces when users get support for navigation. Furthermore, it was shown that just using semantic information is not enough on its own when not combined with a suitable (semantic) interaction metaphor for navigation. Consequently, interaction techniques

that are supportive and intuitive have to be focused more deeply.

The next case study deals with navigation in 3D information space with distinct input methods (mouse vs. touch) and navigation (free vs. constrained) techniques.

7.3 Case Study: Navigating Large Information Spaces

In the previously presented case studies, semantic interaction metaphors have been employed for exploring information spaces in the context of visual retrieval interfaces for folksonomy systems. Besides the conceptual design of such semantic user interfaces, their impact and benefits for improving user experience in tag-based search and undetermined browsing scenarios have been investigated. One important finding was that visual retrieval interfaces for exploration must seamlessly integrate a well suited spatial metaphor for visualizing semantic information and associated data with an efficient and easy-to-use navigation technique for exploring the information space, e.g. a retrieved result set. In order to resolve the usability problems in 3D exploration of the information spaces reported before – in particular for users unexperienced with navigation in 3D space – the following third case study (Aras et al., 2011) investigates a 3D navigation metaphor based on continuous gestures and steering control based on mouse and (multi-)touch input. In order to enable a more natural and efficient interaction for large 3D semantic information spaces, a flexible navigation technique for various visual representations that scales well from low distance interaction to interaction across large distances using continuous gestures for mouse and touch input with visual feedback on direction and speed was researched.

7.3.1 State of the Art

Navigation in 3D spaces was defined by Bowman et al. (2004) as *travel* and *wayfinding*, while travel considers the actions and movements to modify position and perspective, wayfinding rather is concerned with the user's cognitive way of thinking and making decisions and plans etc. In general, visualization in 3D deals with navigation and interaction in contextual information spaces, while allowing the navigation to more detailed information affording different forms of gestures.

Research on 3D travel techniques for multi-touch devices has thus far focused on direct manipulation Hancock et al. (2009), Martinet et al. (2010), Reisman et al. (2009). Yet these do not scale well to large information spaces, since covering large distances requires the user to constantly move fingers/hand/arm back and forth which is physically straining and imprecise.

Particularly for the use cases of undetermined browsing and semantic

exploration described in the previous case studies, such a scalable yet precise and easy-to-use navigation technique is essential in order to explore semantically-related information and contents efficiently. In reference to the case study presented in Section 7.2, semantic relatedness or similarity of the visualized tags and associated content was projected in 3D space along the depth-axis, which can be navigated by the users to explore and find the information they are looking for, starting with one or several related tags and associated contents.

7.3.2 Approach

The steering technique *ElasticSteer* for 3D information browsers on multi-touch and mouse input devices is based on a vehicle metaphor and multiplies two degrees of freedom (DOF) input with modifier multi-finger gestures or mouse buttons. It provides easy feedback on steering direction and speed with a rubber band metaphor. Two variants were designed and implemented: free, unconstrained 6DOF steering and path-constrained 2DOF steering geared toward specific spatial metaphors used in 3D information browsing. An evaluation of the technique showed that constrained 2DOF navigation is more efficient than unconstrained navigation regarding task completion time, number of gestures used and view resets. However, the latter allows the user to develop personal navigation strategies and often feels more immersive. Furthermore, it was also shown that touch input is almost as fast and efficient to use as the mouse while providing the better user experience.

In 3D information spaces the user often has to cover large distances for navigation and search tasks. Thus, a control technique that scales well from small precise navigation to covering large distances is needed. While one can minimize this problem using distinct gestures for switching scales, the following strives for a more integrated approach using continuous interaction and a steering metaphor.

ElasticSteer is based on a rubber band metaphor. This means the speed and direction is based on the relative direction and distance of the starting point of the gesture compared to the current point. This has three advantages: it scales well to large distances without requiring much movement, while still allowing precise movements. Second, it visualizes speed and direction. Third, it provides a physically inspired interaction style that is easy to understand. In order to reduce motor problems and keep mouse compatibility, the control of 6 DOF was separated into three 2 DOF input

techniques. Multi-finger gestures or mouse buttons switch between different DOFs: one finger (LMB) controls camera yaw and camera depth. Two fingers (MMB) allow translation within the view plane. Three fingers (RMB) control roll and pitch. For multi-finger gestures the geometric center is used for calculating speed and direction. In order to determine the impact on navigation efficiency versus user experience and immersion, the steering technique was augmented with path-constraints. These reduce the control DOF to two, but need to be tailored to the employed spatial visualization metaphor.

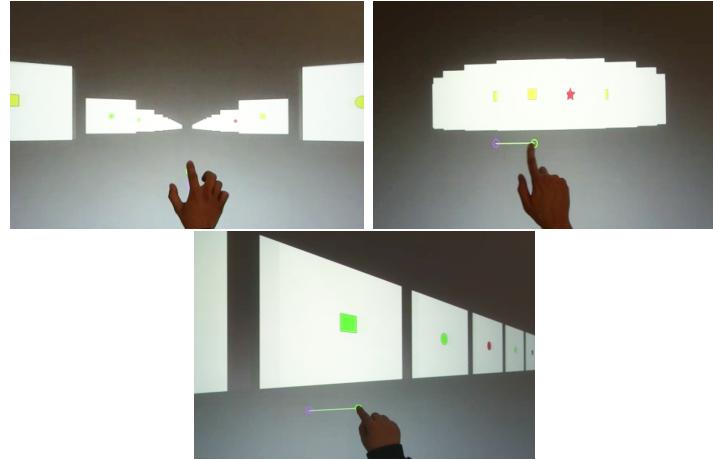


Figure 7.14: Navigating via *ElasticSteer* on an interactive surface.

The path constraints were developed for three state-of-the-art visualization metaphors: corridor, carousel, and wall, as introduced in the case study in Section 7.2. In the *corridor*, objects are arranged in two rows. Supported navigation constrains movement to two axes. Horizontal input movement is mapped to movement to the side while vertical movement is mapped to the depth axis. The *carousel* arranges objects in a circle. Supported navigation constrains movement along a radial path. The radius of the camera path, i.e. the distance from the information objects, can be adjusted by the user. Thus camera movement is constrained to two dimensions; horizontal input moving the camera on the circle, and vertical movement changing the distance to the carousel. The *wall* arranges objects on a 2D plane. In supported navigation, movements take place in two dimensions within a rectangular area. Horizontal movement is mapped to moving sideways, while vertical movement is mapped to moving towards/away from the wall.

7.3.3 User Study

The ElasticSteer prototype was evaluated by conducting a task-based user study with the 3D semantic browser application as described before. The objective was to compare the input methods mouse vs. touch and free vs. constrained navigation. The setup was a 3D space (Figure 7.14) with 45 2D rectangles with colored geometric shapes depicted on them. The task was a naive search task “find the page with the green star”.

The three independent variables to be examined were input device (mouse, multi-touch), control mapping (free/unconstrained, constrained) and spatial metaphor (corridor, carousel, wall). For each task completion time (CT), the number of gestures used (NG) and the number of times the camera position was manually reset (NR) were measured. Each subject had to perform a total of 36 tasks in four steps for each of the three views.

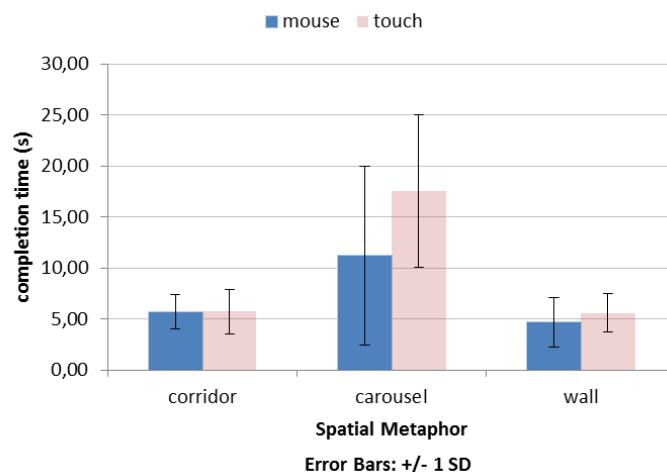


Figure 7.15: Mean values for tested parameter CT: mouse vs. touch.

The subjects were divided into different groups (MU, MC, TU, TC)⁹³ depending on the tested input method (Mouse, Touch) and navigation metaphor (Unconstrained, Constrained). Subsets of these were then formed into independent groups regarding the independent variable. Each participant filled out a questionnaire after the evaluation. Besides logging the user actions, a video was recorded. 16 subjects between 26-39 in age (9 female and 7 male), all right-handed, 7 classified as experts and 9 as novices participated in a similar way to the evaluation of the semantic 3D browser approach. The three spatial metaphors were evaluated separately. As navigation paths may differ depending on the view that is used, the task com-

⁹³M: Mouse, T: Touch, C: Constrained, U: Unconstrained

pletion time was normalized based on the distance to the initial camera position.

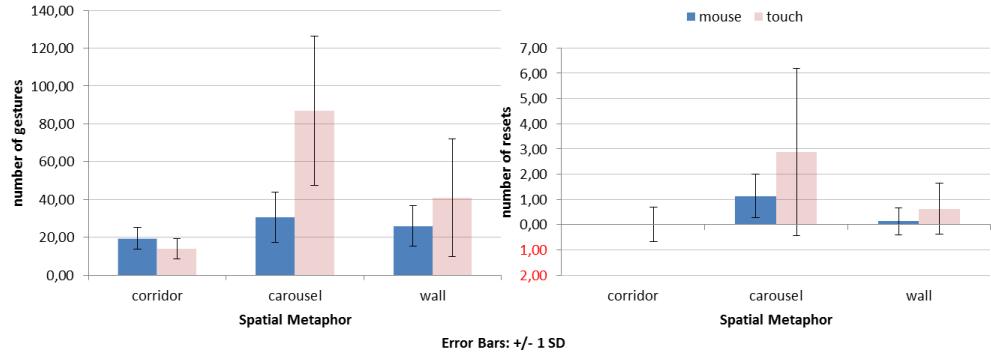


Figure 7.16: Mean values for tested parameters NG, NR: mouse vs. touch.

The results of the quantitative analysis (Figures 7.15 and 7.16) show that the input methods mouse and touch were quite comparable for corridor and wall. Significant differences (Table 7.7 showing paired t-Test with an alpha level of 0.05) can be observed for the number of gestures used in the carousel, which were far less for mouse input and the number of resets in wall view. Task completion time was lower on average for the mouse users, which can be explained by the more familiarity with mouse input.

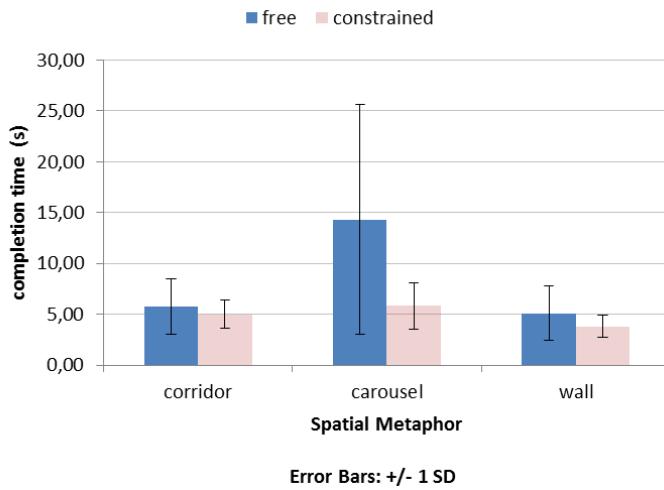


Figure 7.17: Mean values for tested parameter CT: free vs. constrained navigation.

The results further show that the constrained navigation was more efficient (Figure 7.17 and 7.18) for all three tested parameters and each of the views.

Significant results (Table 7.8 showing paired t-Test with an alpha level of 0.05) were obtained for all parameters and views, except the number of resets using the carousel view.

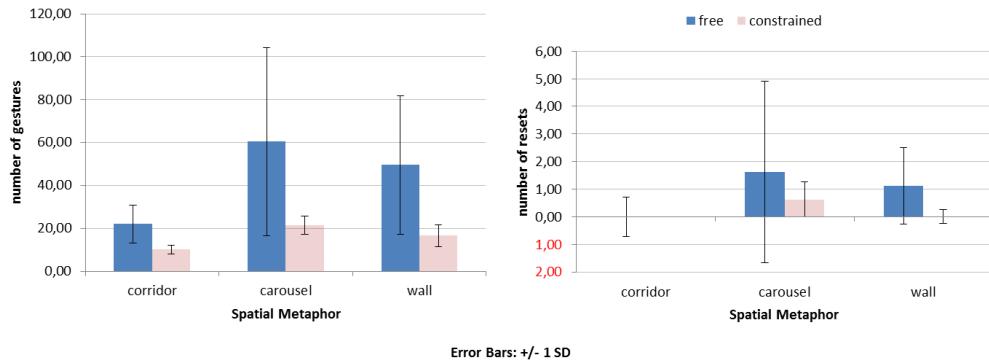


Figure 7.18: Mean values for tested parameters NG, NR: free vs. constrained navigation.

Giving subjective feedback on the input methods, participants stated that the time to complete a task was comparable, but touch gave them more freedom of movement, sense of fun and a better feedback over speed and direction of the navigation. They particularly liked the control and sensory feedback using touch. A few users had the impression that in contrast to mouse input, there is a more direct mapping of gestures to the virtual space.

| | p | $t_{0.05;15}$ | decision |
|-----------------|--------|---------------|-----------------|
| <i>Corridor</i> | | | |
| CT | 0.0720 | 1.93 | significant |
| NG | 0.1320 | 1.59 | not significant |
| NR | 0.1640 | 1.46 | not significant |
| <i>Carousel</i> | | | |
| CT | 0.32 | 1.03 | not significant |
| NG | 0.014 | 2.77 | significant |
| NR | 0.386 | 0.894 | not significant |
| <i>Wall</i> | | | |
| CT | 0.287 | 1.1 | not significant |
| NG | 0.276 | 1.13 | not significant |
| NR | 0.069 | 1.96 | significant |

Table 7.7: ElasticSteer User Study (Mouse vs. Touch).

They felt more immersed in the virtual space navigation and the browsing scenario. Concerning the tested navigation alternatives, constrained navigation was regarded the faster, easier and most preferred method. Nev-

ertheless, free navigation attested the advantages of freedom of movement and more possibilities to develop individual strategies.

| | p | $t_{0.05;15}$ | decision |
|-----------------|--------|---------------|-----------------|
| <i>Corridor</i> | | | |
| CT | 0.0010 | 4.26 | significant |
| NG | 0.0010 | 4.30 | significant |
| NR | 0.0960 | 1.78 | significant |
| <i>Carousel</i> | | | |
| CT | 0 | 4.93 | significant |
| NG | 0.001 | 4.05 | significant |
| NR | 0.108 | 1.71 | not significant |
| <i>Wall</i> | | | |
| CT | 0.003 | 3.51 | significant |
| NG | 0.016 | 2.71 | significant |
| NR | 0.077 | 1.9 | significant |

Table 7.8: ElasticSteer user study (free vs. constrained navigation).

7.3.4 Conclusion

The results of the second case study applying different visualizations and semantic interaction metaphors for 3D attested increased user satisfaction of the browsing experience. Furthermore, the interaction was regarded more efficient and search effort low compared to traditional user interfaces, which allow for no, or restricted semantic access to the underlying information spaces. Although the users had more fun with the 3D browser, they had difficulties with the free navigation in 3D space – in particular, users with no or less 3D gaming experience. Motivated by these comments, a constrained navigation metaphor for mouse and touch interaction was investigated by the described ElasticSteer approach.

Navigation in 3D has been applied in many application scenarios such as 3D games or in 3D applications that are based on direct manipulation. In the focused use case of exploring large semantic information spaces in 3D an appropriate natural and understandable input technique is essential. Such a navigation technique has to fuse 3D spatial metaphors for visualizing a collection of objects in 3D space with a suitable navigation technique in order to be efficient. The 3D navigation technique ElasticSteer implements steering control with visual feedback on direction and speed via a rubber band metaphor. It was shown that it can be successfully used for browsing

(large) semantic information spaces. In the applied semantic 3D browser the semantic proximity/relatedness has been mapped to the depth axis. Employing a semantic vehicle and a rubber band metaphor in the navigation technique allows good scaling from short to long distances along this axis. Since 3D information spaces often follow certain spatial metaphors, the control DOF has been reduced to two via path-constraints geared towards either the wall, carousel or corridor visual metaphor. A study showed that it works almost equally well for mouse and multi-touch input devices with a clear benefit for touch regarding user satisfaction and immersion.

7.4 Conclusion

In the presented 3 case studies it was shown, that semantic interaction metaphors materialized in visual retrieval interfaces applying the general concept of the semantic interaction framework outlined in Section 4.3 help to improve user experience for exploring 2D/3D information spaces.

With the help of two prototypical applications based on implicit semantics extracted from a folksonomy, it was elaborated that semantic interaction structures based on semantic arrangements should be integral part of a visual interaction structure in 2D/3D. Furthermore, the evaluation showed that suitable navigation metaphors are essential in order to allow for navigating the information space efficiently.

In the semantic cloud case study a hierarchical semantic exploration of the tag space was realized employing a multi-topic semantic cloud, where relatedness of tags was mapped to proximity in 2D space. Moreover, clustered related tags form a semantic topic region to be navigated from a bird's eye view with general topics and more specific arrangements employing a zooming or diving metaphor utilized in visual artifacts for exploration, here, a magnifier. One challenge which was difficult to resolve applying the implemented clustering approach was the semantic exploration of semantically related topic regions at the same time. Fuzzy borders applying a non-exclusive clustering method was suggested as a possible workaround.

Applying a similar workflow, a semantic exploration of information spaces in 3D was investigated based on several visualizations. As in the first case study, implicit semantic tags extracted from the folksonomy have been utilized for building semantic arrangements in 3D space, in order to be navigated employing travel or wayfinding interaction. Although users welcomed the used semantic arrangements and 3D visualization of search results, one finding was that simple and transparent visualizations should be preferred,

while intuitive navigation requires the consideration of immersion, e.g. by using semantically arranged websites visualized by using thumbnails.

The navigation metaphor for exploring large information spaces in 3D, was based on a vehicle metaphor employing direct manipulation for navigation interaction. The results of the evaluation of this method clearly hinted at constrained navigation metaphors for restricting the degrees of freedom in 3D reasonably in order to allow it to be used by non-experienced users with 3D.

Chapter 8

Conclusions

The main objective of this thesis was to investigate new forms of semantic interaction with web data exploiting semantic information and distinct interaction metaphors in web-based retrieval systems.

First, basic data and knowledge models for the web were introduced in Chapter 2. The focus in Chapter 3 was to present and discuss state of the art retrieval models and methods for information extraction and existing 2D/3D retrieval user interface concepts. The result of the discussion was that existing user interface concepts for search and exploration are limited due to several aspects. Fundamental challenges exist with respect to semantic relevance of the returned results, information bias observed with the state of the art search engines and their restricted (semantic) interaction capabilities for searching and exploring web content efficiently.

Thus, in order to investigate and introduce a semantic dimension for searching and exploration of web content aiming to improve the user's experience significantly was a primer objective of the research. Therefore, essential tasks for semantic interaction such as semantic layering, semantic mediation and semantic human-computer interaction have been identified and elaborated for two general use case scenarios in web retrieval in Chapter 4.

A semantic interaction framework was introduced in Section 4.3 aiming to describe and provide the basic building blocks and a conceptual design for implementing exemplary use cases. The described tasks were employed in two use case scenarios: web-based question answering in a knowledge-based dialogue system and semantic exploration of information spaces in 2D/3D.

8.1 Aim and Contribution

Automatic semantic access to structured web data and informative web page fragments is challenging due to several reasons. Firstly, missing semantic information and secondly, the lack of contextual markers in order to disambiguate informative parts from non-relevant, i.e. extracting relevant information from web pages.

Therefore, one aim of the thesis was to investigate alternative methods for semantic annotation of existing data on the syntactic web based on the wisdom of the crowds principle. Focusing on structured data and informative web page fragments collaborative tagging and human computation have been applied for tasks of semantic labeling in Chapter 5.

As complex dialogue-based interactions have high requirements for the precision and complexity of the extracted information structures on the web, it was also investigated how to employ an expert-created ontology for semantic annotation, transformation and extraction of information structures from the web. Thus, the aim of the research described in Chapter 6 focused on researching how natural language questions of users can be answered employing a semantic question answering pipeline based on semantic wrappers and be deployed in the knowledge-based real world dialogue system SmartWeb.

The studies described in Chapter 7 aimed to investigate the semantic exploration of information spaces in 2D/3D employing suitable visualizations and semantic interaction metaphors for enhancing the (semantic) interaction capabilities of web-based visual retrieval systems, e.g. for exploring resources in folksonomy systems.

The following sections describe and discuss the main contributions of this thesis with respect to the major tasks for semantic interaction, as defined in the conceptual framework in Chapter 4.

8.1.1 Collaborative Semantic Layering

Social tagging has thus far been applied to single entities such as images, bookmarks, etc. In the tag2wrap study in Section 5.1, the collaborative tagging approach was applied for the task of annotating structured data in web pages. Therefore, a visual annotation metaphor for promoting the conceptualization of tag structures was employed and evaluated.

The aim was to support the creation of conceptual structures for describing structured web data or informative web page fragments, e.g. to be

exploited in information extraction tasks.

With the applied concept it was shown, that collaborative tagging of structured data generates appropriate annotation structures. The experiments further indicated the impact of the visual selection metaphor on tagging behavior for forming conceptual structures, making it necessary to evaluate several other selection metaphors in future work.

The employed consolidation method was successfully applied for selecting the most suitable concept and most synonymous tags could be disambiguated well. Despite this, it was found that the employed method was not optimal in some cases due to the used selection criterion based on occurrence probabilities from WordNet. Particularly for real-world environments this approach might be limited, as most controlled vocabularies or word nets are incomplete, e.g. missing concepts or synonyms.

Furthermore, the user evaluation indicated at the influence of the chosen user group on the quality of the annotations due to their different background and domain knowledge, e.g. soccer fans might be more appropriate for tagging soccer web pages. Last but not least, user motivation and incentives have high impact on user contribution. In particular, disambiguating descriptive tags requires high user contribution, which could be examined and improved with game-centric approaches.

A positive effect on user contribution in a collaborative tagging environment might be the promise of wrapper applications, e.g. semantic search on tagged data, triggering of structured content, news feeds, etc. which could be provided to the contributors automatically after tagging.

As user motivation and missing incentives are serious challenges for collaborative tagging tasks when applied to structured data in direct forms, e.g. via a social tagging user interface, the human computation paradigm was employed for embedding annotation tasks in indirect forms seamlessly, e.g. as part of a gaming action in a computer game in Section 5.2.

In a first study, a mock-up of a tagging game based on binary verification was described. The FastTag proof of concept showed that it is possible to integrate semantic tagging tasks into HC games, while serious challenges exist due to game design which have a high impact on user motivation and user contribution. Despite this, the method was able to exploit the user's background knowledge and human skills such as classification, contextual disambiguation, etc. In particular, the task of annotating structured data or informative web page fragments was investigated. The method allowed the assignment of relevant tags to web page fragments, which could be linked to annotate complex information structures, similar to the tag2wrap approach.

Although similar tasks were resolved in tagging games for single entities such as images, etc. the enrichment of complex information structures in web-pages was not investigated. The first experiences with a human computation tagging game showed the impact of game design for successful adoption of this task. Besides that, user motivation, incentives, rewards, fun, etc. were identified as crucial factors for a better gaming experience, e.g. adopted in OntoGalaxy (Krause et al., 2010) approach successfully.

In Webpardy, the HC paradigm was applied to the task of question answering based on the idea of the popular Jeopardy quiz. The aim was to collect (question, answer) pairs in order to associate a set of relevant high quality questions with a resource. The evaluation comprised the quality and type of the obtained result pairs.

The proof of concept can be regarded as a first step, while issues related to the quality of entered questions, e.g. cheating, entering incorrect sentences, etc. were identified during the experiments. Besides quality management for the obtained questions, the game design including factors such as attractiveness, fun, etc. for obtaining high user contribution can be regarded as essential components of an HC game.

Although the results were generally encouraging, a long term real world adoption of the game is required in order to obtain higher user contribution and for investigating additional aspects of interactivity, success feedback, etc. As a first try, a reversed game mode (double Webpardy) was applied to disambiguate good questions from non-relevant or spam in addition to a reference automatic verification method for entered questions.

8.1.2 Semantic QA in a Dialogue System

Conversational metaphors in human-computer interaction aim to imitate human behavior in interacting with a computer system. In case of the SmartWeb dialogue system introduced in Chapter 6, a major requirement was to access and interpret online information available in web pages just in time for the purpose of question answering. Answering the user's natural language question required the extraction of the answers directly from one or more web documents with high precision and match a given semantic user query based on meaning, i.e. semantic information from ontologies.

Semantic Wrapper Agents in SmartWeb:

The aim of Chapter 6 was to show how natural language questions of users can be answered by employing semantic wrappers as part of a question

answering pipeline in the SmartWeb dialogue system. Therefore, a general concept and a workflow for labeling, extraction and semantic querying of complex information structures was first investigated. Therefore, a semi-automatic visual semantic wrapper generation approach was developed and evaluated for the dialogue system, focusing on querying factual information extracted from the current state of a web page.

An important contribution of the presented approach was the use of linguistic information from the SmartWeb integrated ontology for creating complex annotations (e.g. facts and events related to a football match) through visual interaction. From the labeled sample information structures, semantic wrappers are generated employing a recursive wrapper generation algorithm, while the representation of the underlying ontology remains invisible to the user. Herewith, the complexity and effort for wrapper creation was reduced significantly, as fewer interactions are required to generate wrappers for complex target information structures.

In addition, the integration of semantic wrappers into the semantic question answering pipeline allowed access to online information on web pages just at query time, which is essential for dialogue systems for answering user questions in 1-2 turns. Herewith, semantic access to web pages where the content is changing frequently can be realized, e.g. in pages publishing football events changes can occur any second.

Employing wrappers in dialogue systems requires high precision extraction, in order to be able to return valid ontological instances to be interpreted by the knowledge processing components of the dialogue systems, as automatic approaches produce too much noise, as discussed in Section 3.2.3. Although the scalability of this approach is challenging and requires additional work, focused domains with available domain ontologies can be addressed with reasonable effort. Here, in the worst case, re-creation of several wrapper for the accessed web sources is needed.

Besides that, deploying semantic wrappers into an agent system allowed semantic access to several heterogeneous web sources in order to access and integrate additional aspects of information by means of a “pool of specialized wrappers”. Furthermore, a reduction of the instances to be extracted was achieved employing a form of semantic indexing on previously extracted semantic structures by maintaining a semantic similarity matrix in broker agents that store a score of similarity for each previously executed semantic query. The semantic similarity score was also utilized to select appropriate wrappers for extracting the answer instance candidates from the individual web sources.

Moreover, a successful deployment required a seamless integration of a semantic question answering pipeline based on the components for semantic wrapping, semantic transformation and scoring of question-answer pairs based on the system-wide ontology (SWintO).

The scoring of the relevance of the returned answer instances is an important task for answering natural language questions of users. In this work, an approach based on calculating the semantic similarity of ontological instances considering the semantic content of the extracted information structures was employed. The evaluation showed that the scoring method is able to match ontological question-answer pairs with high precision in the case of factual knowledge, where the structure of the answer instances can be deduced from the ontological model.

8.1.3 Semantic HCI for Exploring Information Spaces

The main contribution of the research described in Chapter 7 was to research the benefit of introducing a semantic dimension into interaction for visual retrieval interfaces in 2D/3D. Therefore, several visual metaphors for the semantic arrangement of web content based on folksonomy tags and enhanced semantic interaction and navigation metaphors for the exploration of web content in 2D/3D space were investigated. Different from previous work on interaction with web content, the research focused on semantic interaction from an HCI perspective.

It was shown that semantic arrangements based on implicit semantics can help to enhance user experience for searching and browsing beyond undetermined search tasks. The results indicate that the design of the semantic arrangements, as well as the navigation technique, have a great influence on user acceptance if they are too complicated and less transparent.

The following sections describe the individual contributions of the presented case studies, focusing on semantic exploration of information spaces in folksonomy-based retrieval systems employing interaction metaphors in 2D/3D.

Semantic Cloud:

The Semantic Cloud study investigated the task for semantic exploration of the folksonomy tag space and related resources employing a hierarchically organized tag cloud metaphor as a visual retrieval interface. Therefore, a hierarchically organized multi-topic semantic cloud based on semantic relat-

edness of tags was employed. The tag space was semantically partitioned based on levels of semantic density – from general frequent co-occurring tags to co-occurring tags from the long tail that may provide specific aspects for a regarded resource. The interface design was compared to a well-established state of the art retrieval interface for folksonomy data in a task-based evaluation.

It was shown that users prefer semantic partitioning of the information space, while distinct levels of specificity must be accessible via appropriate navigation metaphors that are understandable and transparent. In undetermined browsing scenarios where users start with a few general tags in mind, moving from general to regions with specific related tags was attested reasonable, if a suitable navigation metaphor was an integrative part of the interaction structure. In other words, users are able to deduce where to go, which levels to explore and how to navigate to related topics in the hierarchically organized multi-topic semantic tag cloud.

The evaluation of the user interface concept revealed that cross topic exploration must be enhanced in order to allow users to explore several topics at the same time, e.g. travel and photography, avoiding the exploration of two semantic clouds in sequential order or by manual input of tags.

Tag-cloud-based search interfaces represent forms of semantic retrieval user interfaces with low cognitive and physical workload for the users, which was investigated intensively in simple implementations. Moreover, the practical impact of usability compared to state of the art visualizations was investigated through important aspects such as ease of use, interactivity, visualization of search results based on their semantics, support to express information need, etc.

Semantic Space Browser:

The second case study investigated the task of semantic exploration of results of folksonomy retrieval systems, hence, exploring the folksonomy tag space and related resources, by exploiting relatedness of tags in 3D visualizations and navigation. For the navigation in 3D space, the first person shooter perspective based on the direct manipulation metaphor was employed. The semantic navigation technique was realized by mapping semantic relatedness of tags for the regarded resources to the depth axis in 3D space. The proof of concept visual user interface was used to evaluate user experience and several 3D and other interaction parameters.

The evaluation showed that employing 3D visualizations and seman-

tic interaction metaphors were considered helpful for accomplishing the regarded tasks for searching and browsing. Looking more closely at 3D navigation related aspects, they were rated above average, e.g. 3D semantic navigation along the z-axis to reach a target web page.

Despite this, the results clearly depict, that simple designs for semantic interaction should be preferred over complex 3D visualizations of retrieval results. Furthermore, usability problems in navigation 3D information space, e.g. navigating to the target web pages in the search tasks, have been reported by users unexperienced with 3D.

A major finding was the high impact of the navigation metaphor for the semantic exploration. While 3D visualizations of the information space was regarded as being fun and more attractive compared to the baseline interface, the benefits of the additional dimensions can only affect user experience positively, if a dedicated navigation metaphor for semantic exploration, e.g. based on direct manipulation and touch-input, was integrated seamlessly into the 3D interaction and navigation, e.g. via travel and wayfinding interaction and navigation metaphor known from other applications, e.g. map interaction. Investigating 3D-relevant interaction parameters furthermore showed the potential for improving interaction for semantic exploration in large 3D information spaces when users get support for navigation, which was explored in the third case study.

ElasticSteer:

The semantic 3D browser case study clearly showed that input modalities and navigation techniques are essential in order to allow for an efficient and easy-to-use semantic retrieval interface. Semantic interaction in 3D space necessitates to seamlessly fuse 3D spatial metaphors for representing a collection of information objects in 3D space with suitable interaction and navigation techniques. In the tested 3D semantic browser application, semantic proximity was mapped to the depth axis in order to benefit from a navigation technique that allows good scaling from short to long distances along this axis, as the users of a semantic 3D browser must look-up and select single related documents as well as explore other related topic regions arranged in 3D space.

The 3D navigation technique ElasticSteer used steering control with visual feedback on direction and speed with a vehicle/rubber band metaphor. The case study showed, that constrained navigation is superior over free navigation independent of the used input technique – mouse and multi-touch

input – with a clear benefit for touch input regarding user satisfaction and immersion, which also confirmed the results of the user questionnaires from the second study on semantic exploration of folksonomy-based information spaces.

8.2 Outlook and Future Work

User-provided tags represent the user's personal and collective associations for a resource, e.g. image, web page, etc. How to extract and exploit relevant semantics from user tags for returning personalized semantic views on the result sets is an interesting question. Not to forget, that transforming web data to a semantic representation with high quality, necessitates appropriate methods for quality management (Krause, 2014), which can be challenging depending on the type of information and the investigated use case – be it in social web applications or human computation “games with a purpose”.

Besides an automatic verification method more HC-based methods for quality management should be investigated. It may also be reasonable to focus on transferring the gaming experience from the real Jeopardy game, e.g. adopting the missing multi-user experience.

Furthermore, extracting and semantically transforming web data remains a challenge in terms of scalability and degree of automation, allowing room for questions of how to interconnect and re-use existing semantic information structures, e.g. in learning algorithms, for automatic generation of formal semantics with well-defined meanings. Integrating ontology matching methods with semantic wrapper generation, for example, would allow manual steps to be dropped and mapping rules for complex ontologies to be generated increasingly automatically.

Nevertheless, the work in the SmartWeb project on answering natural language questions can be seen as a starting point for online semantic access to web data at query time, which could be adopted in semantic search engines encoding extracted information structures e.g. via RDFa. Moreover, in contrast to web API's which remain data islands and rather allow data access via different proprietary interfaces, data and information interoperability can be achieved by using standard ontologies and information schemes.

Linking distributed semantic data is among the main goals of the linked open data initiative and related projects. It can be assumed that more tools for creating web contents supporting RDFa will be available, while it may not be applied for all domains. Currently, the linked data web contains a

wide range of structured linked data from the domains such as bio-medicine, pharmacy, geography, etc.

In future work, challenges due to scalability of the presented approach could be researched employing collaborative or crowd-based approaches as presented in Chapter 5. Furthermore, the support of questions beyond factual knowledge requires the semantic transformation of the extracted information structures to be enhanced.

Integration of additional scoring concepts like the OntoScore approach would allow the inference of related answers with varying semantic patterns. Besides that, exploiting natural language processing (NLP) techniques for extracting and analyzing semantic instances from unstructured text are further challenges. An initial attempt was explored in Porzel et al. (2006) by employing a constructional analyzer for parsing *breaking news* from live tickers for the sports domain. Possible other application scenarios that are imaginable are triggering of web content (see also Section 4.2.2, legal issues) in form of wrapper applications for e.g. web pages with frequently changing content such as product sales, special offers, news tickers, etc.

Furthermore, the growing impact of the linked data web requires dedicated semantic search interfaces for non-expert users. Searching and exploring the linked data web intuitively via semantic interaction metaphors that are easy to understand, transparent and that provide valuable feedback for discovering new semantically related information is a major challenge for next-generation semantic user interfaces for search.

Although the initial approaches such as the sig.ma search interface exist, allowing the entry of keyword-like queries into the Google-like input form, exploring content this way is rather the way that experts do things, hence, optimizing the user's search and browsing experience is an important task for the linked-data web as well.

The experiences from the case studies in Chapter 7 indicate that semantic information representation and visual interaction metaphors should be integrated seamlessly with easy-to-use navigation techniques. Applying this principle to linked data, where pertinent links and semantic relations among information entities are seamlessly hidden behind easy-to-understand and interpretable associative semantics, could be regarded as an interesting user interface concept, implementable in next-generation semantic user interfaces for search and exploration, returning results and answer candidates based on semantic relevance to the users. Besides that, less work exists on employing social networking principles for exploring large semantic knowledge graphs such as those provided by the linked open data web.

Bibliography

- Allen, J., Miller, Bradford, E. Ringger, and T. Sikorski (1996). A robust system for natural spoken dialogue. In *acl-96s*. [cited at p. 188]
- Allsop, J. (2007). *Microformats: Empowering Your Markup for Web 2.0*. friends of ED. [cited at p. 104]
- Aras, H., W. Cai, and J. Wiersbitzki (2009). Tag2wrap - a social tagging environment for emergent semantic structuresin web documents. In *IADIS WWW/Internet Conference 2009, Rome, Italy, November 19-22*. IADIS. [cited at p. 12, 107]
- Aras, H., V. Chandrasekhara, S. Krueger, R. Malaka, and R. Porzel (2006). Intelligent integration of external data and services into smartkom. In W. Wahlster (Ed.), *SmartKom: Foundations of Multimodal Dialogue Systems*, Cognitive Technologies, pp. 363–378. Springer Berlin Heidelberg. [cited at p. 151]
- Aras, H., M. Krause, A. Haller, and R. Malaka (2010). Webpardy: harvesting qa by hc. In *Proceedings of the ACM SIGKDD Workshop on Human Computation, HCOMP '10*, New York, NY, USA, pp. 49–52. ACM. [cited at p. 12, 127]
- Aras, H., S. Siegel, and R. Malaka (2010, February). Semantic Cloud: An Enhanced Browsing Interface for Exploring Resources in Folksonomy Systems. In *Workshop on Visual Interfaces to the Social and Semantic Web (VISSW2010), IUI2010*. [cited at p. 12, 222, 228]
- Aras, H., B. Walther-Franks, M. Herrlich, P. Rodacker, and R. Malaka (2011). ElasticSteer Navigating Large 3D Information Spaces via Touch or Mouse. In L. Dickmann, G. Volkmann, R. Malaka, S. Boll, A. Krüger, and P. Olivier (Eds.), *Smart Graphics*, Volume 6815 of *Lecture Notes in Computer Science*, Chapter 14, pp. 138–141. Berlin, Heidelberg: Springer Berlin / Heidelberg. [cited at p. 12, 259]

- Arjona, J. L., R. Corchuelo, A. R. Cortés, and M. Toro (2002). A practical agent-based method to extract semantic information from the web. In *CAiSE '02: Proceedings of the 14th International Conference on Advanced Information Systems Engineering*, London, UK, pp. 697–700. Springer-Verlag. [cited at p. 55, 151]
- Aula, A. (2005). *Studying user strategies and characteristics for developing web search interfaces*. Ph. D. thesis, University of Tampere, Finland. [cited at p. 52]
- Babski, C., S. Carion, P. Keller, and C. Guignard (2002). knowscape, a 3d multi-user experimental web browser. In *ACM SIGGRAPH 2002 conference abstracts and applications*, SIGGRAPH '02, New York, NY, USA, pp. 315–315. ACM. [cited at p. 72]
- Baeza Yates, R. A. and B. R. Neto (1999). *Modern Information Retrieval*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. [cited at p. 36, 50, 68, 128, 233]
- Bao, S., G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su (2007). Optimizing web search using social annotations. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, New York, NY, USA, pp. 501–510. ACM. [cited at p. 243]
- Bates, M. J. (1989). The Design of Browsing and Berrypicking Techniques for the Online Search Interface. *Online Review* 13(5), 407–424. [cited at p. 51, 74, 82]
- Baumgartner, R., S. Flesca, and G. Gottlob (2001). Visual web information extraction with lixto. In *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*, San Francisco, CA, USA, pp. 119–128. Morgan Kaufmann Publishers Inc. [cited at p. 57, 155]
- Begelman, G. (2006). Automated tag clustering: Improving search and exploration in the tag space. In *In Proc. of the Collaborative Web Tagging Workshop at WWW06*. [cited at p. 40, 70]
- Belkin, N. (1980). Anomalous states of knowledge as a basis for information retrieval. *Canadian Journal of Information Science* 5, 133–143. [cited at p. 81]
- Belkin, N. J., P. G. Marchetti, and C. Cool (1993, June). Braque: design of an interface to support user interaction in information retrieval. *Inf. Process. Manage.* 29, 325–344. [cited at p. 52]
- Berners-Lee, T., J. Hendler, and O. Lassila (2001). The semantic web. *Scientific American*. [cited at p. 5, 150]

- Bizer, C., T. Heath, and T. Berners-Lee (2009). Linked data - the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)* 5(3), 1–22. [cited at p. 31, 94]
- Bollegrala, D., Y. Matsuo, and M. Ishizuka (2007). Measuring semantic similarity between words using web search engines. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, New York, NY, USA, pp. 757–766. ACM. [cited at p. 40]
- Bosca, A., D. Bonino, M. Comerio, S. Grega, and F. Corno (2007). A reusable 3d visualization component for the semantic web. In *Proceedings of the twelfth international conference on 3D web technology*, Web3D '07, New York, NY, USA, pp. 89–96. ACM. [cited at p. 71]
- Bowman, D. A., E. Kruijff, J. J. LaViola, and I. Poupyrev (2004). *3D User Interfaces: Theory and Practice*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc. [cited at p. 259]
- Brin, S. and L. Page (1998). The anatomy of a large-scale hypertextual web search engine. In *WWW7: Proceedings of the seventh international conference on World Wide Web*, Amsterdam, The Netherlands, The Netherlands, pp. 107–117. Elsevier Science Publishers B. V. [cited at p. 7, 16, 94]
- Broder, A. (2002, September). A taxonomy of web search. *SIGIR Forum* 36, 3–10. [cited at p. 51]
- Broekstra, J. and A. Kampman (2004, August). Serql: An rdf query and transformation language. [cited at p. 214]
- Brooke, J. (1996). SUS: A quick and dirty usability scale. In P. W. Jordan, B. Weerdmeester, A. Thomas, and I. L. Mclelland (Eds.), *Usability evaluation in industry*. London: Taylor and Francis. [cited at p. 117, 251]
- Bruls, M., K. Huizing, and J. van Wijk (2000). Squarified Treemaps. In *Proc. of Joint Eurographics and IEEE TCVG Symp. on Visualization (TCVG 2000)*, pp. 33–42. IEEE Press. [cited at p. 66]
- Buitelaar, P., P. Cimiano, P. Haase, and M. Sintek (2009). Towards Linguistically Grounded Ontologies. In *The Semantic Web: Research and Applications*, pp. 111–125. [cited at p. 30]
- Buitelaar, P., P. Cimiano, S. Racioppa, and M. Siegel (2006, May). Ontology-based Information Extraction with SOBA. In *Proceedings of LREC*, Genoa, Italy. [cited at p. 123]

- Buitelaar, P., T. Declerck, A. Frank, S. Racioppa, M. Kiesel, M. Sintek, R. Engel, M. Romanelli, D. Sonntag, B. Loos, V. Micelli, R. Porzel, and P. Cimiano (2006). LingInfo: Design and Applications of a Model for the Integration of Linguistic Information in Ontologies. In *Proceedings of the 5th international conference on Language Resources and Evaluation (LREC)*. [cited at p. 30, 159, 322]
- Burg, B. (2002). Agents in the world of active web-services. In M. Tanabe, P. van den Besselaar, and T. Ishida (Eds.), *Digital Cities II: Computational and Sociological Approaches*, Volume 2362 of *Lecture Notes in Computer Science*, pp. 119–124. Springer Berlin-Heidelberg. [cited at p. 152]
- C.-N. Hsu, M.-T. D. (1998). Wrapping semistructured web pages with finite-state transducers. In *Proceedings of the Conference on Automatic Learning and Discovery (CONALD-98)*, Pittsburg, PA. Carnegie Mellon University. [cited at p. 57]
- Cai, D., S. Yu, J.-R. Wen, and W.-Y. Ma (2003). Vips: a vision-based page segmentation algorithm. Technical report, Microsoft Research. [cited at p. 108, 124, 159]
- Carpinetto, C. and G. Romano (1996, November). Information retrieval through hybrid navigation of lattice representations. *Int. J. Hum.-Comput. Stud.* 45(5), 553–578. [cited at p. 73]
- Carroll, J. M. and J. C. Thomas (1980). Metaphor and the cognitive representation of computer systems. Technical Report RC8302, IBM T. J. Watson Research Center. [cited at p. 81]
- Carroll, J. M., M.-R. L. . K. W. A. (1988). Interface metaphors and user interface design. In M. Helander (Ed.), *Handbook of Human-Computer Interaction* (Second ed.), pp. 67–85. Elsevier Science Publishers. [cited at p. 87]
- Chang, C. and S. Kuo (2004). Olera: A semi-supervised approach for web data extraction with visual support. *IEEE Intelligent Systems (SCI, EI 19)*. [cited at p. 57]
- Chang, C.-H. and S.-C. Lui (2001). Iepad: information extraction based on pattern discovery. In *WWW'01*, pp. 681–688. [cited at p. 58, 60]
- Chen, H., B. R. Schatz, A. L. Houston, R. R. Sewell, T. D. Ng, and C. Lin (1997). Internet browsing and searching (poster): user evaluations of category map and concept space techniques. In *Proceedings of the second ACM international conference on Digital libraries*, DL '97, New York, NY, USA, pp. 257–. ACM. [cited at p. 68]

- Cilibrasi, R. L. and P. M. B. Vitanyi (2007, March). The google similarity distance. *IEEE Trans. on Knowl. and Data Eng.* 19(3), 370–383. [cited at p. 38]
- Cimiano, P., A. Eberhart, P. Hitzler, D. Oberle, S. Staab, and R. Studer (2004). The SmartWeb Foundational Ontology. Technical report, (AIFB), University of Karlsruhe, Karlsruhe, Germany. SmartWeb Project. [cited at p. 149]
- Ciravegna, F., A. Dingli, D. Petrelli, and Y. Wilks (2002). User-system cooperation in document annotation based on information extraction. In *EKAW '02: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, London, UK, pp. 122–137. Springer-Verlag. [cited at p. 76]
- Cockburn, A. and B. McKenzie (2001). 3d or not 3d?: evaluating the effect of the third dimension in a document management system. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '01, New York, NY, USA, pp. 434–441. ACM. [cited at p. 70]
- Cockburn, A. and B. McKenzie (2002). Evaluating the effectiveness of spatial memory in 2d and 3d physical and virtual environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, CHI '02, New York, NY, USA, pp. 203–210. ACM. [cited at p. 70]
- Cortes, C. and V. Vapnik (1995, September). Support-Vector Networks. *Machine Learning* 20(3), 273–297. [cited at p. 42]
- Cramer, I., J. L. Leidner, and D. Klakow (2006). Building an Evaluation Corpus for German Question Answering by Harvesting Wikipedia. In *Proceedings of The Fifth International Conference on Language Resources and Evaluation (LREC)*, pp. 1514–1519. ELRA. [cited at p. 123]
- Crescenzi, V. and G. Mecca (1998). Grammars have exceptions. *Inf. Syst.* 23(9), 539–565. [cited at p. 54]
- Crescenzi, V., G. Mecca, and P. Merialdo (2001). Roadrunner: Towards automatic data extraction from large web sites. In *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*, San Francisco, CA, USA, pp. 109–118. Morgan Kaufmann Publishers Inc. [cited at p. 59, 60]
- Croft, W. and D. A. Cruse (2004, February). *Cognitive Linguistics (Cambridge Textbooks in Linguistics)*. Cambridge University Press. [cited at p. 107]
- Cruz, I. F., H. Xiao, and F. Hsu (2004). An ontology-based framework for xml semantic integration. Coimbra, Portugal. IDEAS. [cited at p. 188]

- Damerau, F. J. (1964, March). A technique for computer detection and correction of spelling errors. *Commun. ACM* 7(3), 171–176. [cited at p. 38]
- Deerwester, S., S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman (1990). Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE* 41(6), 391–407. [cited at p. 50]
- Dengel, A. (Ed.) (2011, 10). *Semantische Technologien: Grundlagen. Konzepte. Anwendungen.* (1. Aufl. ed.). Spektrum Akademischer Verlag. [cited at p. 28, 321]
- Dengel, A. (Ed.) (2012). *Semantische Technologien: Grundlagen. Konzepte. Anwendungen.* Heidelberg: Spektrum Akademischer Verlag. [cited at p. 82]
- Dill, S., N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, K. S. Mccurley, S. Rajagopalan, A. Tomkins, J. A. Tomlin, and J. Y. Zien (2003). A case for automated large scale semantic annotations. *Journal of Web Semantics* 1, 115–132. [cited at p. 76]
- Ding, Y., H. Litz, R. Malaka, and D. Pfisterer (2003). On Programming Information Agent Systems - An Integrated Hotel Reservation Service as Case Study. In *Proceedings of the first German Conference on Multiagent System Technologies (MATES03)*. [cited at p. 152]
- Ding, Y., R. Malaka, C. Kray, and M. Schillo (2001). Raja: a resource-adaptive java agent infrastructure. In *Proceedings of the fifth international conference on Autonomous agents*, AGENTS '01, New York, NY, USA, pp. 332–339. ACM. [cited at p. 152]
- Doorenbos, R. B., O. Etzioni, and D. S. Weld (1997). A scalable comparison-shopping agent for the world-wide web. In *Proceedings of the first international conference on Autonomous agents*, AGENTS '97, New York, NY, USA, pp. 39–48. ACM. [cited at p. 6]
- Döring, T., H. Aras, B. Walther-Franks, M. Herrlich, P. Rodacker, A. Penner, and R. Malaka (2012). Using Gestural Interaction on Mobile Phones for Navigating 3D Information Spaces on Interactive Walls. In *The 3rd Dimension of CHI (3DCHI) Workshop at CHI 2012*. ACM. [cited at p. 12, 239]
- Ehrig, M. (2006, October). *Ontology Alignment: Bridging the Semantic Gap (Semantic Web and Beyond)*. Springer. [cited at p. 29]
- Ehrig, M. and S. Staab (2004, AUG). Qom - quick ontology mapping. Technical report, Institut AIFB, Universität Karlsruhe. [cited at p. 196]

- Engel, R. (2002). SPIN: Language Understanding for Spoken Dialogue Systems Using a Production System Approach. In *Proc. of 7th International Conference on Spoken Language Processing (ICSLP-2002)*. [cited at p. 145]
- Engel, R. (2006). Spin: A semantic parser for spoken dialog systems. In *Proceedings of Fifth Slovenian and First International Language Technologies Conference (IS-LTC) 2006, Ljubljana, Slovenia*. [cited at p. 146, 188]
- Etzioni, O., M. Cafarella, D. Downey, A. maria Popescu, T. Shaked, S. Soderl, D. S. Weld, and E. Yates (2005). Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence* 165, 91–134. [cited at p. 76]
- Ferguson, G., J. F. Allen, B. Miller, and E. Ringger (1996). The desgin and implementation of the trains-96 system. Technical Report 96-5, University of Rochester, New York. [cited at p. 188]
- Ferrucci, D., E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, N. Schlaefer, and C. Welty (2010). Building Watson: An Overview of the DeepQA Project. *AI Magazine* 31(3). [cited at p. 123, 127]
- Fineman, B. (2004). Computers as people - human interaction metaphors in hci. Master's thesis, Carnegie-Mellon University. [cited at p. 80, 81, 321]
- Frakes, W. (1992). *Stemming Algorithms*, Chapter 8. Prentice-Hall. [cited at p. 38]
- Frohlich, D. (1997). Direct Manipulation and Other Lessons. In M. Helander, T. Landauer, and P. Prabhu (Eds.), *Handbook of Human-Computer Interaction*, Chapter 8. North Holland: Elsevier Science Publishers B.V.,. [cited at p. 7]
- Fu, W.-T., T. G. Kannampallil, and R. Kang (2010). Facilitating exploratory search by model-based navigational cues. In *Proceedings of the 15th international conference on Intelligent user interfaces*, IUI '10, New York, NY, USA, pp. 199–208. ACM. [cited at p. 97]
- Fujimura, K., S. Fujimura, T. Matsubayashi, T. Yamada, and H. Okuda (2008). Topography: visualization for large-scale tag clouds. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, New York, NY, USA, pp. 1087–1088. ACM. [cited at p. 69]
- Furnas, G. W., C. fake, L. von Ahn, J. Schachter, S. Golder, K. Fox, M. Davis, C. Marlow, and M. Naaman (2006). Why do tagging systems work? In *CHI '06 extended abstracts on Human factors in computing systems*, CHI '06, New York, NY, USA, pp. 36–39. ACM. [cited at p. 33]

- Gabrilovich, E. and S. Markovitch (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *In Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 1606–1611. [cited at p. 38]
- Gangemi, A., N. Guarino, C. Masolo, A. Oltramari, and L. Schneider (2002). Sweetening ontologies with dolce. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, EKAW '02, London, UK, pp. 166–181. Springer-Verlag. [cited at p. 149]
- Gangemi, A. and P. Mika (2003). Understanding the semantic web through descriptions and situations. In *CoopIS/DOA/ODBASE*, pp. 689–706. [cited at p. 149]
- Gehrke, J., O. Herzog, H. Langer, R. Malaka, R. Porzel, and T. Warden (2010). An agent-based approach to autonomous logistic processes. *Künstliche Intelligenz* 24(2), 137–141. [cited at p. 151]
- Gemmell, J., A. Shepitsen, B. Mobasher, and R. Burke (2008). Personalizing navigation in folksonomies using hierarchical tag clustering. In *Proceedings of the 10th international conference on Data Warehousing and Knowledge Discovery*, DaWaK '08, Berlin, Heidelberg, pp. 196–205. Springer-Verlag. [cited at p. 69, 229]
- Golder, S. and B. A. Huberman (2005, Aug). The Structure of Collaborative Tagging Systems. [cited at p. 33, 107]
- Golder, S. A. and B. A. Huberman (2006, April). Usage patterns of collaborative tagging systems. *Journal of Information Science* 32(2), 198–208. [cited at p. 36, 226]
- Grahl, M., A. Hotho, and G. Stumme (2007, September). Conceptual clustering of social bookmarking sites. In *7th International Conference on Knowledge Management (I-KNOW '07)*, Graz, Austria, pp. 356–364. Know-Center. [cited at p. 69]
- Grossman, D. A. and O. Frieder (2004). *Information Retrieval: Algorithms and Heuristics* (Zweite ed.). The Kluwer International Series of Information Retrieval. Springer. [cited at p. 36, 37]
- Gruber, T. (1993, June). A translation approach to portable ontology specifications. *Knowledge Acquisition* 5(2), 199–220. [cited at p. 28]
- Grumbach, S. and G. Mecca (1999). In search of the lost schema. In *ICDT '99: Proceedings of the 7th International Conference on Database Theory*, London, UK, pp. 314–331. Springer-Verlag. [cited at p. 53]

- Gupta, M., R. Li, Z. Yin, and J. Han (2010, November). Survey on social tagging techniques. *SIGKDD Explor. Newsl.* 12(1), 58–72. [cited at p. 84]
- Gurevych, I., R. Malaka, R. Porzel, and H.-P. Zorn (2003). Semantic coherence scoring using an ontology. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Morristown, NJ, USA, pp. 9–16. Association for Computational Linguistics. [cited at p. 95, 196, 198, 208]
- Gurevych, I. and H. Niederlich (2005). Accessing germanet data and computing semantic relatedness. In *Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*, ACLdemo '05, Stroudsburg, PA, USA, pp. 5–8. Association for Computational Linguistics. [cited at p. 38]
- Gurevych, I., R. Porzel, E. Slinko, N. Pfleger, J. Alexandersson, and S. Merten (2003). Less is more: Using a single knowledge representation in dialogue systems. In *Proceedings of the HLT-NAACL'03 Workshop on Text Meaning*, Edmonton, Canada, pp. 14–21. [cited at p. 188]
- Gusfield, D. (1997). *Algorithms on Strings, Trees and Sequences*. CUP, Cambridge, UK. [cited at p. 56, 58, 223]
- Guy, M. and E. Tonkin (2006). Folksonomies. Tidying up Tags? *D-Lib Magazine* 12(1). [cited at p. 227]
- Halpin, H., V. Robu, and H. Shepherd (2007). The complex dynamics of collaborative tagging. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, New York, NY, USA, pp. 211–220. ACM. [cited at p. 113, 226]
- Hammouda, K. M. and M. S. Kamel (2004, November). Document Similarity Using a Phrase Indexing Graph Model. *Knowl. Inf. Syst.* 6(6), 710–727. [cited at p. 37]
- Hancock, M., T. T. Cate, and S. Carpendale (2009). Sticky tools: Full 6dof force-based interaction for multi-touch tables. In *Proc. ITS*. [cited at p. 259]
- Handschrift, S. and S. Staab (2003). Cream: Creating metadata for the semantic web. Volume 42, pp. 579–598. New York, NY, USA: Elsevier North-Holland, Inc. [cited at p. 62]
- Harper, B. D. and K. L. Norman (1993). Improving user satisfaction: The questionnaire for user interaction satisfaction version 5.5. In *Proceedings of the 1st Annual Mid-Atlantic Human Factors Conference*, pp. 224–228. [cited at p. 251]

- Hassan-Montero, Y. and V. Herrero-Solana (2006). Improving tag-clouds as visual information retrieval interfaces. In *InScit2006: International Conference on Multidisciplinary Information Sciences and Technologies*. [cited at p. 69, 70]
- Hearst, M. A. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora. In *In Proceedings of the 14th International Conference on Computational Linguistics*, pp. 539–545. [cited at p. 128, 132]
- Hearst, M. A. (2009). *Search User Interfaces* (1 ed.). Cambridge University Press. [cited at p. 51, 52, 219]
- Hearst, M. A. and J. O. Pedersen (1996). Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '96, New York, NY, USA, pp. 76–84. ACM. [cited at p. 68, 224]
- Heath, T. (2008). *Information-seeking on the Web with Trusted Social Networks from Theory to Systems*. Ph. D. thesis, Knowledge Media Institute - Open University, Milton Keynes, United Kingdom. [cited at p. 99]
- Heath, T. and C. Bizer (2011). *Linked Data: Evolving the Web into a Global Data Space*, Volume 1. Morgan & Claypool. [cited at p. 94]
- Heath, T. and E. Motta (2008). Revyu: Linking reviews and ratings into the web of data. *Web Semant.* 6(4), 266–273. [cited at p. 63, 65]
- Heflin, J. and J. Hendler (2001). A portrait of the semantic web in action. *IEEE Intelligent Systems* 16(2), 54–59. [cited at p. 62]
- Held, C. and U. Cress (2008). Social tagging aus kognitionspsychologischer sicht. In B. Gaiser, T. Hampel, and S. Panke (Eds.), *Good Tags-Bad Tags: Social Tagging in der Wissensorganisation*, pp. 37–49. Mnster: Waxmann. [cited at p. 7, 98]
- Henzinger, M. (2005). Hyperlink analysis on the world wide web. In *HYPertext '05: Proceedings of the sixteenth ACM conference on Hypertext and hypermedia*, New York, NY, USA, pp. 1–3. ACM. [cited at p. 15]
- Higgins, D. and J. Burstein (2007). Sentence similarity measures for essay coherence. In *Proceedings of the 7th International Workshop on Computational Semantics (IWCS)*. [cited at p. 131]
- Hitzler, P., R. Sebastian, and M. Krötzsch (2009). *Foundations of Semantic Web Technologies*. London: Chapman & Hall/CRC. [cited at p. 157]
- Hoare, C. and H. Sorensen (2005, November). Information foraging with a proximity-based browsing tool. *Artif. Intell. Rev.* 24, 233–252. [cited at p. 66]

- Hogue, A. (2004). Tree pattern inference and matching for wrapper induction on the world wide web. Master's thesis, Massachusetts Institute of Technology. [cited at p. 60]
- Hogue, A. and D. Karger (2005). Thresher: automating the unwrapping of semantic content from the world wide web. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, New York, NY, USA, pp. 86–95. ACM. [cited at p. 57, 155]
- Honkela, T., S. Kaski, K. Lagus, and T. Kohonen (1997). Websom - self-organizing maps of document collections. In *Neurocomputing*, pp. 101–117. [cited at p. 40]
- Horner, M. (2008). The giraffe semantic web browser. In *Proceedings of the 12th international conference on Entertainment and media in the ubiquitous era, MindTrek '08*, New York, NY, USA, pp. 184–188. ACM. [cited at p. 72]
- Horrocks, I. (2008, December). Ontologies and the semantic web. *Commun. ACM* 51, 58–67. [cited at p. 153]
- Huck, G., P. Fankhauser, K. Aberer, and E. J. Neuhold (1998). Jedi: Extracting and synthesizing information from the web. In *COOPIS '98: Proceedings of the 3rd IFCIS International Conference on Cooperative Information Systems*, Washington, DC, USA, pp. 32–43. IEEE Computer Society. [cited at p. 54, 60, 168]
- Jain, S. and D. C. Parkes (2009). The role of game theory in human computation systems. In *HCOMP '09: Proceedings of the ACM SIGKDD Workshop on Human Computation*, New York, NY, USA, pp. 58–61. ACM. [cited at p. 121]
- Jennings, N. R. (2001, April). An agent-based approach for building complex software systems. *Commun. ACM* 44, 35–41. [cited at p. 151]
- Jennings, N. R. and M. J. Wooldridge (Eds.) (1998). *Agent technology: foundations, applications, and markets*. Secaucus, NJ, USA: Springer-Verlag New York, Inc. [cited at p. 151]
- Jiang, J. J. and D. W. Conrath (1997, September). Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In *International Conference Research on Computational Linguistics (ROCLING X)*, pp. 9008+. [cited at p. 37]
- Joachims, T., L. Granka, B. Pan, H. Hembrooke, and G. Gay (2005). Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, New York, NY, USA, pp. 154–161. ACM. [cited at p. 72]

- Kahan, J. and M.-R. Koivunen (2001). Annotea: an open rdf infrastructure for shared web annotations. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, New York, NY, USA, pp. 623–632. ACM. [cited at p. 62]
- Käki, M. (2005). Findex: search result categories help users when document ranking fails. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '05, New York, NY, USA, pp. 131–140. ACM. [cited at p. 68]
- Kalfoglou, Y. and M. Schorlemmer (2003). Ontology mapping: the state of the art. *The Knowledge Engineering Review* 18(1), 1–31. [cited at p. 196]
- Kirn, S., O. Herzog, P. Lockemann, and O. Spaniol (2006). *Multiagent engineering: theory and applications in enterprises*. International handbooks on information systems. Springer. [cited at p. 151]
- Klein, M., D. Fensel, F. van Harmelen, and I. Horrocks (2000). The relation between ontologies and schema-languages: translating OIL-specifications in XML-schema. In *Proc. of the Workshop on Application of Ontologies and Problem Solving Methods*, Berlin, Germany. [cited at p. 196]
- Knoblock, C. A., S. Minton, J. L. Ambite, N. Ashish, P. J. Modi, I. Muslea, A. G. Philpot, and S. Tejada (1998). Modeling web sources for information integration. In *AAAI '98/IAAI '98: Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, Menlo Park, CA, USA, pp. 211–218. American Association for Artificial Intelligence. [cited at p. 13]
- Konchady, M. (2008). *Building Search Applications: Lucene, LingPipe, and Gate*. Mustru Publishing. [cited at p. 74]
- Kossmann, D. (2000). The state of the art in distributed query processing. *ACM Computing Surveys* 32, 2000. [cited at p. 94]
- Kowalski, G. (1997). *Information Retrieval Systems: Theory and Implementation* (1st ed.). Norwell, MA, USA: Kluwer Academic Publishers. [cited at p. 43, 242]
- Krause, M. (2014). *Homo Ludens in the Loop*. Ph. D. thesis, University of Bremen. [cited at p. 277]
- Krause, M. and H. Aras (2009). Playful tagging: folksonomy generation using online games. In *WWW '09 Proceedings of the 18th international conference on World wide web*, Madrid, pp. 1207–1208. ACM Press. [cited at p. 12, 125]

- Krause, M., A. Takhtamysheva, M. Wittstock, and R. Malaka (2010). Frontiers of a paradigm: exploring human computation with digital games. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '10, New York, NY, USA, pp. 22–25. ACM. [cited at p. 121, 272]
- Krotzsch, M., P. Hitzler, D. Vrandecic, and M. Sintek (2006). How to reason with owl in a logic programming system. In *Proceedings of the Second International Conference on Rules and Rule Markup Languages for the Semantic Web*, Washington, DC, USA, pp. 17–28. IEEE Computer Society. [cited at p. 146]
- Kushmerick, N. (1997). *Wrapper Induction for Information Extraction*. Ph. D. thesis. [cited at p. 55]
- Kushmerick, N. (2000). Wrapper induction: efficiency and expressiveness. *Artif. Intell.* 118(1-2), 15–68. [cited at p. 56]
- Kushmerick, N., D. S. Weld, and R. B. Doorenbos (1997). Wrapper induction for information extraction. In *Intl. Joint Conference on Artificial Intelligence (IJCAI)*, pp. 729–737. [cited at p. 55]
- Laender, A. H. F., B. A. Ribeiro-Neto, A. S. da Silva, and J. S. Teixeira (2002). A brief survey of web data extraction tools. *SIGMOD Rec.* 31(2), 84–93. [cited at p. 54]
- Lamping, J., R. Rao, and P. Pirolli (1995). A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '95, New York, NY, USA, pp. 401–408. ACM Press/Addison-Wesley Publishing Co. [cited at p. 66]
- Landauer, T., D. Egan, J. Remde, M. Lesk, C. Lochbaum, and D. Ketchum (1993). Enhancing the usability of text through computer delivery and formative evaluation: The SuperBook project. In C. McKnight, A. Dillon, and J. Richardson (Eds.), *Hypertext: A Psychological Perspective*, Ellis Horwood Series in Interactive Information Systems, Chapter 5. Ellis Horwood Limited. [cited at p. 72]
- Laniado, D., D. Eynard, and M. Colombetti (2007). A semantic tool to support navigation in a folksonomy. In *Proceedings of the eighteenth conference on Hypertext and hypermedia*, HT '07, New York, NY, USA, pp. 153–154. ACM. [cited at p. 227]
- Lerman, K., S. N. Minton, and C. A. Knoblock (2003). Wrapper maintenance: A machine learning approach. *Journal of Artificial Intelligence Research* 18, 2003. [cited at p. 54, 321]

- Lewandowski, D., H. Wahlig, and G. Meyer-Bautz (2006). The freshness of web search engine databases. *J. Inf. Sci.* 32(2), 131–148. [cited at p. 75, 94]
- Libkin, L. (2005). Logics for unranked trees: An overview. In L. Caires, G. Italiano, L. Monteiro, C. Palamidessi, and M. Yung (Eds.), *Automata, Languages and Programming*, Volume 3580 of *Lecture Notes in Computer Science*, pp. 35–50. Springer Berlin / Heidelberg. [cited at p. 19]
- Lidwell, W., K. Holden, and J. Butler (2003, October). *Universal Principles of Design*. Rockport Publishers. [cited at p. 52]
- Lin, D. (1998). An Information-Theoretic Definition of Similarity. In *In Proceedings of the 15th International Conference on Machine Learning*, pp. 296–304. [cited at p. 37]
- Lin, X. and D. Soergel (1991). A Self Organizing Semantic Map for Information Retrieval. In *Proc. 14th International SIGIR Conference*, pp. 262–269. Chicago. [cited at p. 44]
- Liu, B. (2007a). *Link Analysis*, pp. 217–271. Springer, Berlin-Heidelberg. [cited at p. 22, 23]
- Liu, B. (2007b). *Rule Induction*, pp. 75–81. Springer, Berlin-Heidelberg. [cited at p. 16, 57]
- Liu, B. (2007c). *Structured Data Extraction: Wrapper Generation*, pp. 323–379. Springer, Berlin-Heidelberg. [cited at p. 60]
- Liu, B. (2007d). *Web Data Mining - Exploring Hyperlinks, Contents, and Usage Data* (1st ed.). Berlin-Heidelberg, Germany: Springer, Berlin-Heidelberg. [cited at p. 24, 43, 44, 129, 212, 230]
- Liu, L., W. Han, D. Buttler, C. Pu, and W. Tang (1999). An xml-based wrapper generator for web information extraction. In *In Proc.of ACM-SIGMOD 99*, pp. 540–543. [cited at p. 57]
- Lohmann, S., J. Ziegler, and L. Tetzlaff (2009). Comparison of tag cloud layouts: Task-related performance and visual exploration. In *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part I*, INTERACT '09, Berlin, Heidelberg, pp. 392–404. Springer-Verlag. [cited at p. 69, 245]
- Lux, M., M. Granitzer, and R. Kern (2007). Aspects of broad folksonomies. In *Proceedings of the 18th International Conference on Database and Expert Systems Applications*, DEXA '07, Washington, DC, USA, pp. 283–287. IEEE Computer Society. [cited at p. 38, 128]

- Mai, J.-E. (2006). Contextual analysis for the design of controlled vocabularies. *Bulletin of the American Society for Information Science and Technology* 33(1), 17–19. [cited at p. 84]
- Manning, C. D., P. Raghavan, and H. Schütze (2008, July). *Introduction to Information Retrieval* (1 ed.). Cambridge University Press. [cited at p. 43, 74, 229]
- Manning, C. D. and H. Schütze (1999). *Foundations of statistical natural language processing*. Cambridge, MA, USA: MIT Press. [cited at p. 39]
- Marchetti, A., M. Tesconi, F. Ronzano, M. Rosella, and S. Minutoli (2007, May). Semkey: A semantic collaborative tagging system. [cited at p. 62]
- Marchionini, G. (1989, January). Information-seeking strategies of novices using a full-text electronic encyclopedia. *J. Am. Soc. Inf. Sci.* 40, 54–66. [cited at p. 51]
- Marchionini, G. (2006, April). Exploratory search: from finding to understanding. *Commun. ACM* 49, 41–46. [cited at p. 64, 73, 220]
- Marchionini, G. and R. White (2007). Find What You Need, Understand What You Find. *International Journal of Human-Computer Interaction* 23(3), 205–237. [cited at p. 51, 99]
- Marcus, A. (1994). Managing metaphors for advanced user interfaces. In *Proceedings of the workshop on Advanced visual interfaces*, AVI '94, New York, NY, USA, pp. 12–18. ACM. [cited at p. 80]
- Markov, Z. and D. T. Larose (2007). Wiley-Interscience. [cited at p. 37, 42]
- Marlow, C., M. Naaman, D. Boyd, and M. Davis (2006). HT06, tagging paper, taxonomy, flickr, academic article, to read. In *HYPertext'06: Proceedings of the 7th conference on hypertext and hypermedia*, New York, NY, USA, pp. 31–40. ACM Press. [cited at p. 32, 34, 36]
- Martinet, A., G. Casiez, and L. Grisoni (2010). The design and evaluation of 3d positioning techniques for multi-touch displays. In *Proc. 3DUI*, pp. 115–118. IEEE. [cited at p. 259]
- Mathes, A. (2004, december). Folksonomies - cooperative classification and communication through shared metadata. [cited at p. 97]
- McCool, R. (2006). Rethinking the semantic web, part 2. *IEEE Internet Computing* 10(1), 96–95. [cited at p. 104]

- Mika, P. (2005). Ontologies are us: A unified model of social networks and semantics. In *Proceedings of the 4th International Semantic Web Conference (ISWC'05)*, Galway, Ireland. [cited at p. 34, 69, 140]
- Mika, P. (2007). *Social Networks and the Semantic Web*, Chapter 4, pp. 69. Springer Science + Business Media, LLC. [cited at p. 27, 61, 321]
- Mirizzi, R., A. Ragone, T. Di Noia, and E. Di Sciascio (2010). Semantic wonder cloud: exploratory search in dbpedia. In *Proceedings of the 10th international conference on Current trends in web engineering, ICWE'10*, Berlin, Heidelberg, pp. 138–149. Springer-Verlag. [cited at p. 66, 67]
- Mitchell, J. C. (1969). University Press, Manchester. [cited at p. 32]
- Mitchell, T. M. (1980). The Need for Biases in Learning Generalizations. Technical Report CBM-TR-117, Departament of Computer Science, Rutgers University. [cited at p. 40]
- Mitchell, T. M. (1997). *Machine Learning* (1 ed.). McGraw-Hill Science/Engineering/Math. [cited at p. 40, 41]
- Moens, M.-F. (2006, October). *Information Extraction: Algorithms and Prospects in a Retrieval Context (The Information Retrieval Series)* (1 ed.). Springer. [cited at p. 45, 58, 128]
- Moreno, A. and J. Pavón (2007). *Issues in Multi-Agent Systems: The AgentCities.ES Experience (Whitestein Series in Software Agent Technologies and Autonomic Computing)*. [cited at p. 152]
- Morrison, D. R. (1968). Patricia—practical algorithm to retrieve information coded in alphanumeric. *J. ACM* 15(4), 514–534. [cited at p. 58]
- Muslea, I., S. Minton, and C. Knoblock (1999). A hierarchical approach to wrapper induction. In *AGENTS '99: Proceedings of the third annual conference on Autonomous Agents*, New York, NY, USA, pp. 190–197. ACM. [cited at p. 56, 57]
- Narayanan, S. and S. Harabagiu (2004a, May 2 - May 7). Answering questions using advanced semantics and probabilistic inference. In S. Harabagiu and F. Lacatusu (Eds.), *HLT-NAACL 2004: Workshop on Pragmatics of Question Answering*, Boston, Massachusetts, USA, pp. 10–16. Association for Computational Linguistics. [cited at p. 198]
- Narayanan, S. and S. Harabagiu (2004b). Question answering based on semantic structures. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics. [cited at p. 198]

- Neale, D. C. and J. M. Carroll (1997). The Role of Metaphors in User Interface Design. In M. Helander, T. K. Landauer, and P. Prabhu (Eds.), *Handbook of Human-Computer Interaction* (Second ed.), Chapter 20, pp. 441–462. Elsevier. [cited at p. 87]
- Neven, F. (2002, September). Automata theory for xml researchers. *SIGMOD Rec.* 31(3), 39–46. [cited at p. 19]
- Nielsen, J. (1993). *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. [cited at p. 235]
- Niles, I. and A. Pease (2001). Towards a standard upper ontology. In C. Welty and B. Smith (Eds.), *Workshop on Ontology Management*, Ogunquit, Maine. Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001). [cited at p. 149]
- Norman, D. A. (1988, June). *The Psychology Of Everyday Things*. Basic Books. [cited at p. 51]
- Oberle, D., A. Ankolekar, P. Hitzler, P. Cimiano, M. Sintek, M. Kiesel, B. Mougalouie, S. Baumann, S. Vembu, M. Romanelli, P. Buitelaar, R. Engel, D. Sonntag, N. Reithinger, B. Loos, H.-P. Zorn, V. Micelli, R. Porzel, C. Schmidt, M. Weiten, F. Burkhardt, and J. Zhou (2007, September). Dolce ergo sumo: On foundational and domain models in the smartweb integrated ontology (swinto). *Web Semant.* 5(3), 156–174. [cited at p. 93]
- Oblinger, D. and J. Oblinger (2005). *Is it age or IT: First steps toward understanding the Net Generation*, pp. 1–12. EDUCAUSE. [cited at p. 65]
- Ottoson, E. (2008). *Söka sitt : Om möten mellan människor och föremål*. Ph. D. thesis, Uppsala University, Department of Cultural Anthropology and Ethnology. [cited at p. 65]
- Pascoe, J. (1997). The stick-e note architecture: extending the interface beyond the user. In *IUI '97: Proceedings of the 2nd international conference on Intelligent user interfaces*, New York, NY, USA, pp. 261–264. ACM Press. [cited at p. 105]
- Passant, A. and P. Laublet (2008). Meaning of a tag: A collaborative approach to bridge the gap between tagging and linked data. *Proceedings of the WWW 2008 Workshop Linked Data on the Web (LDOW2008), Beijing, China, Apr.* [cited at p. 63]

- Perez, C. and A. d. Antonio (2004). 3d visualization of text collections: An experimental study to assess the usefulness of 3d. In *Proceedings of the Information Visualisation, Eighth International Conference*, IV '04, Washington, DC, USA, pp. 317–323. IEEE Computer Society. [cited at p. 70]
- Pivk, A., Y. Sure, P. Cimiano, M. Gams, V. Rajkovic, and R. Studer (2006). Transforming arbitrary tables into f-logic frames with tartar. *Data & Knowledge Engineering (DKE)*. accepted for publication. [cited at p. 187]
- Plaisant, C., J. Grosjean, and B. B. Bederson (2002). SpaceTree: supporting exploration in large node link tree, design evolution and empirical evaluation. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pp. 57–64. [cited at p. 66]
- Popov, B., A. Kiryakov, D. Manov, A. Kirilov, and O. M. Goranov (2003). Towards semantic web information extraction. In *In proceedings of ISWC (Sundial Resort*. [cited at p. 76]
- Porzel, R., V. Micelli, H. Aras, and H. P. Zorn (2006). Tying the knot: Ground entities, descriptions and information objects for construction-based information extraction. In *In Proceedings of Ontolex 2006*. [cited at p. 190, 278]
- Porzel, R., N. Pfleger, S. Merten, M. Loeckelt, I. Gurevych, R. Engel, and J. Alexandersson (2003). More on less: Further applications of ontologies in multi-modal dialogue systems. In *Proceedings of the 3rd IJCAI 2003 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, Acapulco, Mexico. [cited at p. 188]
- Porzel, R., H.-P. Zorn, B. Loos, and R. Malaka (2006). Towards a separation of pragmatic knowledge and contextual information. In *Proceedings of the ECAI Workshop on Contexts and Ontologies*, Riva del Garda, Italy. [cited at p. 146]
- Poslad, S., P. Buckle, and R. Hadingham (2000). The fipa-os agent platform: Open source for open standards. In *Proceedings of the 5th International Conference and Exhibition on the Practical Application of Intelligent Agents and MultiAgents* 355, 368. [cited at p. 153]
- Pretzsch, C. (2006). Ontology-based answer selection in dialog systems. Master's thesis, University of Heidelberg, Heidelberg, Germany. [cited at p. 199]
- Price, G. and C. Sherman (2001, July). *The Invisible Web: Uncovering Information Sources Search Engines Can't See* (1 ed.). Information Today, Inc. [cited at p. 62]
- Quinlan, J. R. (1986, March). Induction of decision trees. *Machine Learning* 1(1), 81–106. [cited at p. 42]

- Quinlan, J. R. (1993, January). *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)* (1 ed.). Morgan Kaufmann. [cited at p. 42]
- Quint, V. and I. Vatton (1997). An introduction to amaya. *World Wide Web J.* 2(2), 39–46. [cited at p. 62]
- R. Studer, Richard V. Benjamins, D. F. (1998, March). Knowledge engineering: Principles and methods. *Data and Knowledge Engineering* 25(1-2), 161–197. [cited at p. 28]
- Reisman, J. L., P. L. Davidson, and J. Y. Han (2009). A screen-space formulation for 2d and 3d direct manipulation. In *Proc. UIST*, pp. 69–78. ACM. [cited at p. 259]
- Reithinger, N., S. Bergweiler, R. Engel, G. Herzog, N. Pfleger, M. Romanelli, and D. Sonntag (2005). A look under the hood: design and development of the first smartweb system demonstrator. In *ICMI*, pp. 159–166. [cited at p. 144]
- Resnik, P. (1995). Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *IJCAI*, pp. 448–453. [cited at p. 37]
- Rijsbergen, C. J. V. (1979). *Information Retrieval* (2nd ed.). Newton, MA, USA: Butterworth-Heinemann. [cited at p. 45]
- Risden, K., M. P. Czerwinski, T. Munzner, and D. B. Cook (2000, November). An initial examination of ease of use for 2d and 3d information visualizations of web content. *Int. J. Hum.-Comput. Stud.* 53(5), 695–714. [cited at p. 70]
- Ritter, H. and T. Kohonen (1989). Self-organizing semantic maps. *Biological Cybernetics* 61(4), 241–254. [cited at p. 43]
- Robertson, G. G., S. K. Card, and J. D. Mackinlay (1993, April). Information visualization using 3d interactive animation. *Commun. ACM* 36(4), 57–71. [cited at p. 73]
- Robertson, G. G., J. D. Mackinlay, and S. K. Card (1991). Cone trees: animated 3d visualizations of hierarchical information. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, CHI '91, New York, NY, USA, pp. 189–194. ACM. [cited at p. 70]
- S. Greenaway, M. Thelwall, Y. D. (2009). Tagging youtube - a classification of tagging practice on youtube. In *Proc. 12th International Conference on Scientometrics and Informetrics, 14th-17th July,, Rio De Janeiro, Brazil*, pp. 660–664. ISSI. [cited at p. 120]

- Sahuguet, A. and F. Azavant (1999). Building light-weight wrappers for legacy web data-sources using w4f. In *VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases*, San Francisco, CA, USA, pp. 738–741. Morgan Kaufmann Publishers Inc. [cited at p. 57, 155]
- Salton, G. (1989). *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. [cited at p. 51]
- Schönhage, B., A. van Ballegooij, and A. Ellïëns (2000). 3d gadgets for business process visualization: a case study. In *Proceedings of the fifth symposium on Virtual reality modeling language (Web3D-VRML)*, VRML '00, New York, NY, USA, pp. 131–138. ACM. [cited at p. 70]
- Schrammel, J., M. Leitner, and M. Tscheligi (2009). Semantically structured tag clouds: an empirical evaluation of clustered presentation approaches. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, New York, NY, USA, pp. 2037–2040. ACM. [cited at p. 69]
- Sebastiani, F. (2002, March). Machine learning in automated text categorization. *ACM Comput. Surv.* 34, 1–47. [cited at p. 69]
- Sebrechts, M. M., J. V. Cugini, S. J. Laskowski, J. Vasilakis, and M. S. Miller (1999). Visualization of search results: a comparative evaluation of text, 2d, and 3d interfaces. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, New York, NY, USA, pp. 3–10. ACM. [cited at p. 70]
- Seifert, C., B. Kump, W. Kienreich, G. Granitzer, and M. Granitzer (2008). On the beauty and usability of tag clouds. In *Proceedings of the 2008 12th International Conference Information Visualisation*, IV '08, Washington, DC, USA, pp. 17–25. IEEE Computer Society. [cited at p. 245]
- Shen, D., J. L. Leidner, A. Merkel, and D. Klakow (2006). The alyssa system at trec 2006: A statistically-inspired question answering system. In *The Fifteenth Text REtrieval Conference (TREC 2006)*. [cited at p. 123]
- Sheth, A. P. and C. Ramakrishnan (2007). Relationship web: Blazing semantic trails between web resources. *IEEE Internet Computing* 11(4), 77–81. [cited at p. 7]
- Shneiderman, B. (1987, March). Designing the user interface strategies for effective human-computer interaction. *SIGBIO Newslett.* 9, 6–. [cited at p. 80]
- Shneiderman, B., D. Byrd, and W. B. Croft (1997). Clarifying search: A user-interface framework for text searches. Technical report. [cited at p. 7]

- Shneiderman, B., D. Byrd, and W. B. Croft (1998, April). Sorting out searching: a user-interface framework for text searches. *Commun. ACM* 41, 95–98. [cited at p. 51]
- Simon, K. and G. Lausen (2005). Viper: augmenting automatic information extraction with visual perceptions. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, New York, NY, USA, pp. 381–388. ACM. [cited at p. 56, 59]
- Sinha, R. and R. Mihalcea (2007). Unsupervised graph-basedword sense disambiguation using measures of word semantic similarity. In *Proceedings of the International Conference on Semantic Computing*, ICSC '07, Washington, DC, USA, pp. 363–369. IEEE Computer Society. [cited at p. 40]
- Sinha, S. and S. Narayanan (2005, July). Model-based answer selection. In *AAAI05: Workshop on Inference for Textual Question Answering*. Pittsburgh, Pennsylvania (US). [cited at p. 198]
- Siorpaes, K. and M. Hepp (2007). OntoGame: Towards overcoming the incentive bottleneck in ontology building. *On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops*, 1222–1232. [cited at p. 122]
- Siorpaes, K. and M. Hepp (2008a). Games with a Purpose for the Semantic Web. *IEEE Intelligent Systems* 23(3), 50–60. [cited at p. 122]
- Siorpaes, K. and M. Hepp (2008b). OntoGame: Weaving the Semantic Web by Online Games. *The Semantic Web: Research and Applications*, 751–766. [cited at p. 122]
- Sipser, M. (1996). *Introduction to the Theory of Computation* (1st ed.). International Thomson Publishing. [cited at p. 162, 168]
- Sonntag, D., R. Engel, G. Herzog, A. Pfalzgraf, N. Pfleger, M. Romanelli, and N. Reithinger (2007). Smartweb handheld - multimodal interaction with ontological knowledge bases and semantic web services. In *Artifical Intelligence for Human Computing*, pp. 272–295. [cited at p. 145, 146]
- Spaerck-Jones, K. (1981). *Information Retrieval Experiment*. Newton, MA, USA: Butterworth-Heinemann. [cited at p. 46]
- Specia, L. and E. Motta (2007). Integrating folksonomies with the semantic web. In *Proceedings of the 4th European conference on The Semantic Web: Research and Applications*, ESWC '07, Berlin, Heidelberg, pp. 624–639. Springer-Verlag. [cited at p. 40, 69, 223, 228]

- Stoica, E., M. A. Hearst, and M. Richardson (2007). Automating creation of hierarchical faceted metadata structures. In C. L. Sidner, T. Schultz, M. Stone, and C. Zhai (Eds.), *HLT-NAACL*, pp. 244–251. The Association for Computational Linguistics. [cited at p. 69]
- Strube, M. and S. P. Ponzetto (2006, July). WikiRelate! Computing Semantic Relatedness Using Wikipedia. [cited at p. 40]
- Suzuki, J., Y. Sasaki, and E. Maeda (2002). Svm answer selection for open-domain question answering. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING '02, Stroudsburg, PA, USA, pp. 1–7. Association for Computational Linguistics. [cited at p. 209]
- Takhtamysheva, A., R. Porzel, and M. Krause (2009). Games for games: manipulating contexts in human computation games. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '09, New York, NY, USA, pp. 38–39. ACM. [cited at p. 120]
- Tanasescu, V. and O. Streibel (2007, November). Extreme tagging: Emergent semantics through the tagging of tags. In P. Haase, A. Hotho, L. Chen, E. Ong, and P. C. Mauroux (Eds.), *Proceedings of the International Workshop on Emergent Semantics and Ontology Evolution (ESOE2007) at ISWC/ASWC2007, Busan, South Korea*. [cited at p. 63]
- Teevan, J., S. T. Dumais, and E. Horvitz (2010, April). Potential for personalization. *ACM Trans. Comput.-Hum. Interact.* 17(1), 4:1–4:31. [cited at p. 65]
- Toms, E. G., H. L. O'Brien, R. Kopak, and L. Freund (2005). Searching for relevance in the relevance of search. In *Proceedings of the 5th international conference on Context: conceptions of Library and Information Sciences*, CoLIS'05, Berlin, Heidelberg, pp. 59–78. Springer-Verlag. [cited at p. 99]
- Tran, T., P. Cimiano, S. Rudolph, and R. Studer (2008). Ontology-based interpretation of keywords for semantic search. pp. 523–536. [cited at p. 72]
- Tullis, T. and W. Albert (2008, March). *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics (Interactive Technologies)*. Morgan Kaufmann. [cited at p. 252]
- Tummarello, G., R. Delbru, and E. Oren (2007). Sindice.com: weaving the open linked data. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, ISWC'07/ASWC'07, Berlin, Heidelberg, pp. 552–565. Springer-Verlag. [cited at p. 67]

- von Ahn, L. and L. Dabbish (2004). Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, New York, NY, USA, pp. 319–326. ACM. [cited at p. 84, 120]
- Von Ahn, L., B. Maurer, C. McMillen, D. Abraham, and M. Blum (2008, September). reCAPTCHA: human-based character recognition via Web security measures. *Science* 321(5895). [cited at p. 6, 120]
- Wahlster, W. (2004). Smartweb: Mobile applications of the semantic web. In *Proceedings of Informatik 2004*. [cited at p. 143]
- Wahlster, W. (2006). *SmartKom: Foundations of Multimodal Dialogue Systems (Cognitive Technologies)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc. [cited at p. 188]
- Wang, J. and F. H. Lochovsky (2003). Data extraction and label assignment for web databases. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, New York, NY, USA, pp. 187–196. ACM. [cited at p. 59]
- Weizenbaum, J. (1966, January). Eliza: a computer program for the study of natural language communication between man and machine. *Commun. ACM* 9(1), 36–45. [cited at p. 80]
- Woodruff, A., A. Faulring, R. Rosenholtz, J. Morrisson, and P. Pirolli (2001). Using thumbnails to search the web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '01, New York, NY, USA, pp. 198–205. ACM. [cited at p. 73]
- Wooldridge, M. (1999). Multi-agent systems - a modern aproach to distributed artific intelligence. In G. Weiss (Ed.), *Intelligent Agents*. MIT Press. [cited at p. 151]
- Wu, Z. and M. Palmer (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pp. 133–138. Association for Computational Linguistics. [cited at p. 37]
- Xiang, P. and Y. Shi (2006). Recovering semantic relations from web pages based on visual cues. In *Proceedings of the 11th international conference on Intelligent user interfaces*, IUI '06, New York, NY, USA, pp. 342–344. ACM. [cited at p. 108]
- Xu, J., A. Licuanan, J. May, S. Miller, and R. M. Weischedel (2003). Answer selection and confidence estimation. In *New Directions in Question Answering*, pp. 134–137. [cited at p. 198]

- Yee, K.-P., D. Fisher, R. Dhamija, and M. Hearst (2001). Animated exploration of dynamic graphs with radial layout. In *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, Washington, DC, USA, pp. 43–. IEEE Computer Society. [cited at p. 66]
- Zhai, Y. and B. Liu (2005). Web data extraction based on partial tree alignment. In *Proceedings of the 14th international conference on World Wide Web*, WWW '05, New York, NY, USA, pp. 76–85. ACM. [cited at p. 56]
- Zhai, Y. and B. Liu (2007). Extracting web data using instance-based learning. *World Wide Web* 10(2), 113–132. [cited at p. 57]

Appendix A

Appendix of Chapter 5

A.1 Evaluation of the Tagging Method

| type | description |
|------|---|
| 1 | single elements can be tagged no unstructured text nested hierarchies |
| 2 | single or groups of data can be tagged unstructured text here and there hierarchies are visible |
| 3 | mostly unstructured text single data items can't be tagged barely or no structural hierarchies |

Table A.1: Web page types in the web experiment.

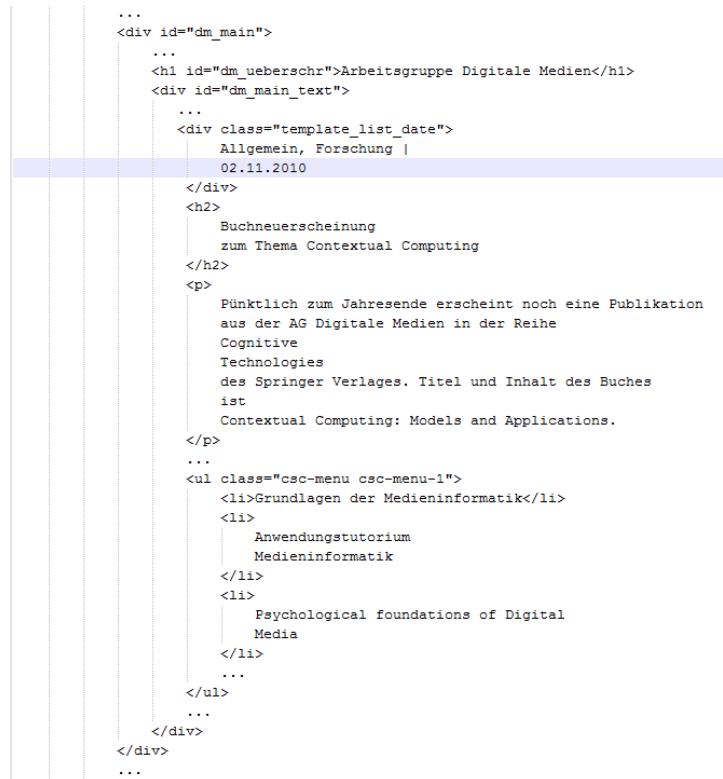
| Nr | Question |
|----|--|
| 1 | Overall satisfaction with tag2wrap? |
| 2 | Overall usage was interesting? |
| 3 | Easy to use/complexity? |
| 4 | User Interface, visual impression? |
| 5 | Feedback and status of interaction? |
| 6 | Intuitive usage of selection and tagging? |
| 7 | Usefulness of multi-selection (bounding box)? |
| 8 | Highlighting textual information parts useful? |
| 9 | Tag recommendation desirable? |

Table A.2: Questionnaire for the web experiment.

Appendix B

Appendix of Chapter 6

B.1 Example HTML Page: Research Group Digital Media News.



The screenshot shows a portion of an HTML page. A horizontal blue bar highlights the date '02.11.2010' in a list item under a heading. The page contains several sections of text and lists:

```
...
<div id="dm_main">
  ...
  <h1 id="dm_ueberschr">Arbeitsgruppe Digitale Medien</h1>
  <div id="dm_main_text">
    ...
    <div class="template_list_date">
      Allgemein, Forschung |
      02.11.2010
    </div>
    <h2>
      Buchneuerscheinung
      zum Thema Contextual Computing
    </h2>
    <p>
      Pünktlich zum Jahresende erscheint noch eine Publikation
      aus der AG Digitale Medien in der Reihe
      Cognitive
      Technologies
      des Springer Verlages. Titel und Inhalt des Buches
      ist
      Contextual Computing: Models and Applications.
    </p>
    ...
    <ul class="csc-menu csc-menu-1">
      <li>Grundlagen der Medieninformatik</li>
      <li>
        Anwendungstutorium
        Medieninformatik
      </li>
      <li>
        Psychological foundations of Digital
        Media
      </li>
      ...
    </ul>
    ...
  </div>
</div>
...
```

Figure B.1: Example - Research group Digital Media news page.

B.2 Sample Structure for the Semantic Class "footballMatch"

```

<footballMatch>
  <matchInfos>
    <round>Gruppenspiele</round>
    <matchname>Costa Rica - Polen</matchname>
    <matchresult>1:2 (1:1)</matchresult>
    <matchnr>34</matchnr>
    <date>20 Juni 2006</date>
    <time>16:00</time>
    <location>Hanover / FIFA World Cup Stadium, Hanover</location>
    <spectators>43000</spectators>
    <goals>
      <scoredGoal>
        <scorer>Ronald GOMEZ (CRC)</scorer>
        <minute>25'</minute>
      </scoredGoal>
      ...
    </goals>
  </matchInfos>
  <matchOfficials>
    <referee>Shamsul MAIDIN (SIN)</referee>
    ...
  </matchOfficials>
  <matchTeam>
    <teamname>Costa Rica</teamname>
    <footballPlayer>
      <playernr>[18]</playernr>
      <playername>Jose PORRAS (GK)</playername>
    </footballPlayer>
    <footballPlayer>
      <playernr>[2]</playernr>
      <playername>Jervis DRUMMOND (-70')</playername>
    </footballPlayer>
    ...
  </matchTeam>
  ...
  <matchTeam>
    ...
  </matchTeam>
</footballMatch>

```

B.3 Visual Wrapper Tool JEFF

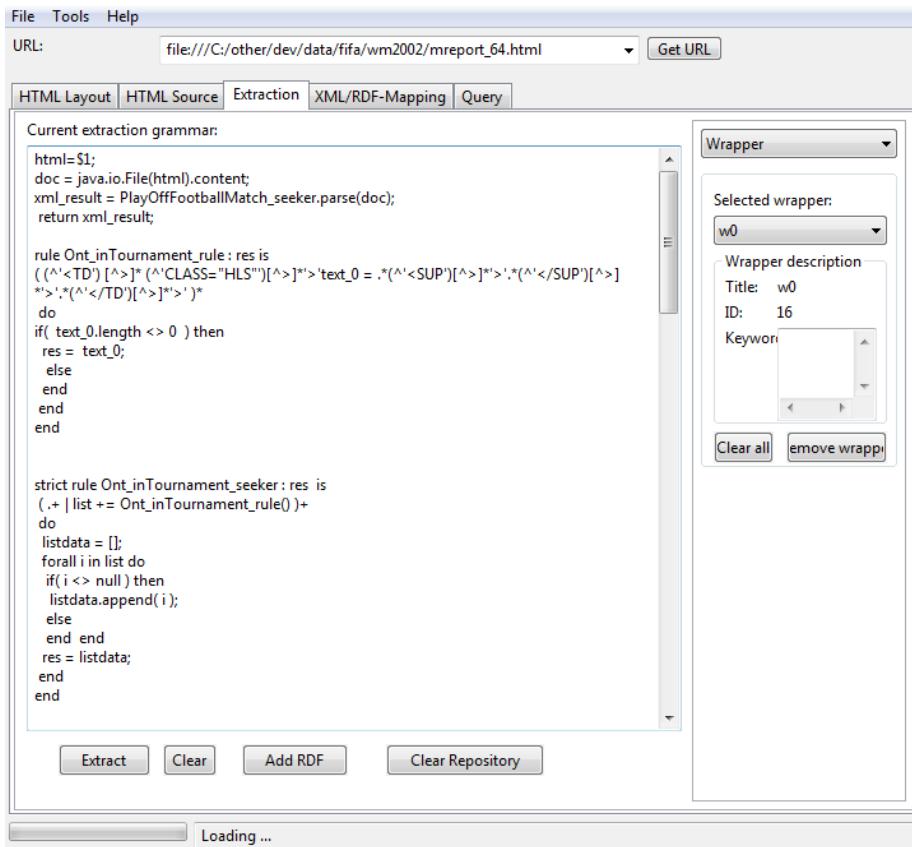


Figure B.2: Jedi Grammar Generation in JEFF.

B.4 Rule Naming Conventions

| Rule type | Convention | Example |
|-----------|------------------------|---------------------|
| a | ont_<predicate>_i | ont_keywords_0 |
| s | ont_<concept> | ont_agnews |
| g+ | ont_<concept>_iterator | ont_agnews_iterator |
| j | ont_<topConcept> | ont_dmag |

Table B.1: Rule naming conventions.

B.5 SeRQL query example

```

SELECT contextProperty
FROM
  {x0} rdf:type {j.6:WorldCup},
  {x0} j.0:HAPPENS-AT {x1},
  {x1} j.0:BEGINS {x2},
  {x2} rdf:type {j.0:time-point},
  {x1} rdf:type {j.0:time-interval},
  {x0} j.6:winner {x3},
  {x3} j.6:origin {x4},
  {x4} j.0:IDENTIFIER {var5},
  {x4} rdf:type {j.3:Nation},
  {x3} rdf:type {j.6:FootballNationalTeam},
  {x2} j.0:YEAR {contextProperty} WHERE var5 LIKE "Brasilien" IGNORE CASE
USING NAMESPACE
  j.1=<http://smartweb.semanticweb.org/ontology/mpeg7#>,
  xsd=<http://www.w3.org/2001/XMLSchema#>,
  rdfs=<http://www.w3.org/2000/01/rdf-schema#>,
  j.2=<http://smartweb.semanticweb.org/ontology/emma#>,
  owl=<http://www.w3.org/2002/07/owl#>,
  j.3=<http://smartweb.semanticweb.org/ontology/smartsumo#>,
  dc=<http://purl.org/dc/elements/1.1/>,
  j.0=<http://smartweb.semanticweb.org/ontology/smardolce#>,
  j.5=<http://smartweb.semanticweb.org/ontology/discourse#>,
  j.4=<http://smartweb.semanticweb.org/ontology/swemma#>,
  j.6=<http://smartweb.semanticweb.org/ontology/sportevent#>,
  daml=<http://www.daml.org/2001/03/daml+oil#>,
  rdf=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>

```

Figure B.3: “When was Brazil soccer world champion?” as SeRQL query.

B.6 Sample s2rdf Rule File (.s2r) for the FIFA corpus.

1.) Mapping

```

list Resource, list footballResource
concept footballMatch s:RoundStageFootballMatch
concept worldCup s:FIFAWorldCup
concept fmatchTeam s:FootballMatchTeam
concept squad s:Squad
concept nationalTeam s:FootballNationalTeam
concept matchTeam s:FootballMatchTeam
concept timePoint sd:time-point
concept player s:FootballPlayer
concept fieldPlayer s:FieldMatchFootballPlayer
concept substitute s:SubstituteMatchFootballPlayer
concept goalkeeper s:Goalkeeper
concept stadium s:Stadium
concept officials s:officials
concept referee s:Referee
concept substitution s:Substitution
concept yellowCard s:ShowingYellowCard
concept redCard s:ShowingRedCard
concept scoreGoal s:ScoreGoal
property location ss:hasAddress
property player s:committedOn
property committedBy s:committedBy
property lineUp s:lineup
property bench s:bench
property country s:heldIn
property matchResult s:matchResult
property type s:officialName
property happensAt sd:HAPPENS-AT
property atMinute sd:HAPPENS-AT
property name s:impersonatedBy
property year sd:YEAR
property day sd:DAY
property month sd:MONTH
property hour sd: HOUR
property minute sd:MINUTE
property second sd:SECOND
condmap name s:RoundStageFootballMatch sd:HAS-DENOMINATION
condmap name s:Stadium sd:HAS-DENOMINATION
condmap name s:Goalkeeper s:hasUpperRole
condmap name s:FieldMatchFootballPlayer s:hasUpperRole
condmap name s:SubstituteMatchFootballPlayer s:hasUpperRole
condmap heldIn s:RoundStageFootballMatch s:heldIn
condmap name s:FootballMatchTeam sd:HAS-DENOMINATION
condmap partOf s:Squad s:partOf
maplist matchEvents s:MatchEvents
list goals, list cards, list substitutions
maplist lineUp s:lineup, maplist bench s:bench

```

2.) Insertion/Expansion

```

insert s:FIFAWorldCup[sd:HAPPENS-AT] sd:time-interval[sd:BEGINS]
insert s:FIFAWorldCup[s:heldIn] ss:Country[sd:HAS-DENOMINATION]
insert ss:Country[sd:HAS-DENOMINATION] ss:country-denomination[sd:NAME]

insert s:RoundStageFootballMatch[sd:HAPPENS-AT] sd:time-interval[sd:BEGINS]

insert s:Stadium[ss:hasAddress] ss:Address[ss:hasCity]
insert ss:Address[ss:hasCity] ss:City[sd:IDENTIFIER]

insert s:Stadium[sd:HAS-DENOMINATION] sd:denomination[sd:NAME]

insert s:ScoreGoal[sd:HAPPENS-AT] sd:time-point-relative[sd:OFFSET]
insert s:ScoreGoal[s:committedBy] s:FieldMatchFootballPlayer[s:hasUpperRole]

insert s:ShowingYellowCard[sd:HAPPENS-AT] sd:time-point-relative[sd:OFFSET]
insert s:ShowingRedCard[sd:HAPPENS-AT] sd:time-point-relative[sd:OFFSET]
insert s:Substitution[sd:HAPPENS-AT] sd:time-point-relative[sd:OFFSET]

insert s:ShowingYellowCard[s:committedOn] s:FieldMatchFootballPlayer[s:hasUpperRole]
insert s:ShowingRedCard[s:committedOn] s:FieldMatchFootballPlayer[s:hasUpperRole]

insert s:Substitution[s:in] s:SubstituteMatchFootballPlayer[s:hasUpperRole]
insert s:Substitution[s:out] s:FieldMatchFootballPlayer[s:hasUpperRole]

insert s:FootballPlayer[s:impersonatedBy] sd:natural-person[sd:HAS-DENOMINATION]
insert sd:natural-person[sd:HAS-DENOMINATION] sd:natural-person-denomination[sd:NAME]

insert s:RoundStageFootballMatch[sd:HAS-DENOMINATION] sd:denomination[sd:NAME]
insert s:FootballMatchTeam[sd:HAS-DENOMINATION] sd:denomination[sd:NAME]

insert s:FootballNationalTeam[s:origin] ss:Country[sd:HAS-DENOMINATION]

insert s:Referee[s:impersonatedBy] sd:natural-person[sd:HAS-DENOMINATION]
insert s:Referee[s:nationality] ss:Nation[sd:HAS-DENOMINATION]
insert ss:Nation[sd:HAS-DENOMINATION] sd:denomination[sd:NAME]

insert s:FieldMatchFootballPlayer[s:hasUpperRole] s:FootballPlayer[s:impersonatedBy]
insert s:SubstituteMatchFootballPlayer[s:hasUpperRole] s:FootballPlayer[s:impersonatedBy]
insert s:Goalkeeper[s:hasUpperRole] s:FootballPlayer[s:impersonatedBy]

3.) Merging

merge smartdolce:time-point smartdolce:MINUTE smartdolce:SECOND smartdolce: HOUR
merge smartdolce:time-point-relative smartdolce:OFFSET
merge sportevent:FootballPlayer sportevent:impersonatedBy

merge sportevent:Goalkeeper sportevent:hasUpperRole
merge sportevent:FieldMatchFootballPlayer sportevent:hasUpperRole
merge sportevent:SubstituteMatchFootballPlayer sportevent:hasUpperRole

merge sportevent:Substitution smartdolce:HAPPENS-AT
merge smartdolce:natural-person-denomination smartdolce:NAME
merge smartdolce:natural-person smartdolce:natural-person-denomination
merge sportevent:RoundStageFootballMatch smartdolce:HAS-DENOMINATION

```

B.7 ProperScore Evaluation

| q/a | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1 | 1.00 | 0.21 | 0.20 | 0.41 | 0.20 | 0.20 | 0.25 | 0.28 | 0.20 | 0.20 | 0.41 | 0.30 | 0.18 | 0.18 |
| 2 | 0.50 | 0.98 | 0.41 | 0.25 | 0.40 | 0.43 | 0.42 | 0.29 | 0.59 | 0.43 | 0.42 | 0.31 | 0.29 | 0.29 |
| 3 | 0.51 | 0.41 | 1.00 | 0.21 | 0.39 | 0.53 | 0.51 | 0.06 | 0.43 | 0.53 | 0.51 | 0.09 | 0.06 | 0.06 |
| 4 | 0.54 | 0.12 | 0.19 | 1.00 | 0.17 | 0.19 | 0.27 | 0.19 | 0.10 | 0.19 | 0.54 | 0.09 | 0.06 | 0.06 |
| 5 | 0.28 | 0.24 | 0.24 | 0.15 | 1.00 | 0.46 | 0.28 | 0.08 | 0.24 | 0.55 | 0.30 | 0.29 | 0.08 | 0.08 |
| 6 | 0.19 | 0.21 | 0.20 | 0.12 | 0.54 | 1.00 | 0.19 | 0.10 | 0.20 | 0.59 | 0.19 | 0.40 | 0.10 | 0.10 |
| 7 | 0.56 | 0.39 | 0.47 | 0.39 | 0.47 | 0.47 | 1.00 | 0.20 | 0.39 | 0.47 | 0.56 | 0.25 | 0.20 | 0.20 |
| 8 | 0.44 | 0.27 | 0.19 | 0.30 | 0.19 | 0.19 | 0.19 | 0.93 | 0.27 | 0.19 | 0.19 | 0.44 | 0.40 | 0.40 |
| 9 | 0.49 | 0.64 | 0.52 | 0.12 | 0.47 | 0.52 | 0.49 | 0.07 | 1.00 | 0.52 | 0.49 | 0.10 | 0.07 | 0.07 |
| 10 | 0.17 | 0.18 | 0.18 | 0.11 | 0.57 | 0.52 | 0.17 | 0.09 | 0.18 | 1.00 | 0.17 | 0.35 | 0.09 | 0.09 |
| 11 | 0.44 | 0.21 | 0.24 | 0.30 | 0.27 | 0.24 | 0.42 | 0.04 | 0.20 | 0.24 | 1.00 | 0.05 | 0.04 | 0.04 |
| 12 | 0.42 | 0.27 | 0.20 | 0.20 | 0.36 | 0.35 | 0.24 | 0.40 | 0.27 | 0.35 | 0.20 | 1.00 | 0.26 | 0.26 |
| 13 | 0.34 | 0.34 | 0.24 | 0.24 | 0.24 | 0.24 | 0.24 | 0.53 | 0.34 | 0.24 | 0.24 | 0.34 | 1.00 | 0.93 |
| 14 | 0.34 | 0.34 | 0.24 | 0.24 | 0.24 | 0.24 | 0.24 | 0.53 | 0.34 | 0.24 | 0.24 | 0.34 | 0.93 | 1.00 |

Figure B.4: Evaluation scores for Dataset 1

| q/a | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1 | 1.00 | 0.76 | 0.76 | 0.75 | 0.71 | 0.71 | 0.71 | 0.74 | 0.75 | 0.73 | 0.70 | 0.65 | 0.65 | 0.65 | 0.66 | 0.66 |
| 2 | 0.76 | 1.00 | 0.76 | 0.76 | 0.71 | 0.71 | 0.71 | 0.75 | 0.75 | 0.73 | 0.70 | 0.65 | 0.65 | 0.65 | 0.66 | 0.66 |
| 3 | 0.76 | 0.76 | 1.00 | 0.75 | 0.71 | 0.71 | 0.70 | 0.73 | 0.75 | 0.73 | 0.69 | 0.65 | 0.65 | 0.64 | 0.65 | 0.66 |
| 4 | 0.70 | 0.70 | 0.70 | 1.00 | 0.68 | 0.67 | 0.67 | 0.71 | 0.70 | 0.67 | 0.64 | 0.60 | 0.60 | 0.59 | 0.60 | 0.60 |
| 5 | 0.75 | 0.75 | 0.75 | 0.76 | 1.00 | 0.77 | 0.77 | 0.75 | 0.75 | 0.72 | 0.74 | 0.70 | 0.70 | 0.70 | 0.71 | 0.71 |
| 6 | 0.75 | 0.75 | 0.75 | 0.76 | 0.77 | 1.00 | 0.78 | 0.76 | 0.75 | 0.73 | 0.74 | 0.71 | 0.71 | 0.70 | 0.71 | 0.71 |
| 7 | 0.73 | 0.73 | 0.73 | 0.74 | 0.76 | 0.76 | 1.00 | 0.74 | 0.73 | 0.71 | 0.74 | 0.70 | 0.70 | 0.70 | 0.71 | 0.71 |
| 8 | 0.71 | 0.71 | 0.71 | 0.73 | 0.72 | 0.72 | 0.72 | 1.00 | 0.71 | 0.69 | 0.69 | 0.64 | 0.64 | 0.64 | 0.65 | 0.65 |
| 9 | 0.74 | 0.74 | 0.74 | 0.74 | 0.70 | 0.70 | 0.70 | 0.73 | 1.00 | 0.74 | 0.69 | 0.66 | 0.66 | 0.67 | 0.65 | 0.65 |
| 10 | 0.72 | 0.72 | 0.72 | 0.72 | 0.68 | 0.68 | 0.68 | 0.71 | 0.74 | 1.00 | 0.69 | 0.67 | 0.66 | 0.66 | 0.65 | 0.65 |
| 11 | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 | 0.74 | 1.00 | 0.72 | 0.72 | 0.71 | 0.71 | 0.71 | 0.71 |
| 12 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.79 | 0.79 | 0.79 | 1.00 | 0.81 | 0.80 | 0.78 | 0.78 | 0.78 |
| 13 | 0.77 | 0.77 | 0.77 | 0.77 | 0.78 | 0.79 | 0.79 | 0.77 | 0.80 | 0.79 | 0.78 | 0.82 | 1.00 | 0.81 | 0.78 | 0.78 |
| 14 | 0.67 | 0.67 | 0.67 | 0.67 | 0.71 | 0.70 | 0.71 | 0.67 | 0.69 | 0.68 | 0.71 | 0.73 | 0.73 | 1.00 | 0.75 | 0.72 |
| 15 | 0.68 | 0.68 | 0.68 | 0.68 | 0.72 | 0.71 | 0.72 | 0.68 | 0.69 | 0.68 | 0.71 | 0.71 | 0.71 | 0.75 | 1.00 | 0.73 |
| 16 | 0.68 | 0.68 | 0.68 | 0.68 | 0.72 | 0.71 | 0.72 | 0.68 | 0.69 | 0.68 | 0.71 | 0.72 | 0.72 | 0.73 | 1.00 | |

Figure B.5: Evaluation scores for Dataset 2

| q/a | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | | | |
|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|--|--|--|
| 1 | 1.00 | 0.55 | 0.40 | 0.20 | 0.20 | 0.20 | 0.21 | 0.41 | 0.40 | 0.20 | 0.20 | 0.20 | 0.30 | 0.20 | 0.20 | 0.26 | 0.56 | 0.55 | 0.55 | 0.56 | 0.55 | 0.25 | 0.20 | 0.41 | 0.55 | | | |
| 2 | 0.55 | 1.00 | 0.44 | 0.28 | 0.28 | 0.28 | 0.43 | 0.43 | 0.28 | 0.28 | 0.24 | 0.28 | 0.29 | 0.20 | 0.55 | 0.55 | 0.55 | 0.55 | 0.55 | 0.31 | 0.28 | 0.31 | 0.43 | | | | | |
| 3 | 0.30 | 0.31 | 1.00 | 0.14 | 0.09 | 0.09 | 0.11 | 0.30 | 0.32 | 0.44 | 0.46 | 0.36 | 0.26 | 0.11 | 0.09 | 0.14 | 0.30 | 0.30 | 0.30 | 0.30 | 0.30 | 0.18 | 0.14 | 0.30 | 0.30 | | | |
| 4 | 0.61 | 0.59 | 0.59 | 1.00 | 0.53 | 0.53 | 0.51 | 0.61 | 0.61 | 0.61 | 0.52 | 0.61 | 0.29 | 0.51 | 0.53 | 0.41 | 0.61 | 0.61 | 0.61 | 0.61 | 0.63 | 0.61 | 0.41 | 0.41 | 0.41 | | | |
| 5 | 0.46 | 0.45 | 0.37 | 0.40 | 1.00 | 0.65 | 0.55 | 0.38 | 0.39 | 0.40 | 0.37 | 0.40 | 0.29 | 0.54 | 0.57 | 0.38 | 0.50 | 0.47 | 0.46 | 0.46 | 0.46 | 0.40 | 0.39 | 0.21 | 0.30 | | | |
| 6 | 0.46 | 0.45 | 0.37 | 0.40 | 0.63 | 1.00 | 0.57 | 0.38 | 0.39 | 0.40 | 0.37 | 0.40 | 0.31 | 0.54 | 0.57 | 0.38 | 0.47 | 0.47 | 0.46 | 0.48 | 0.46 | 0.40 | 0.39 | 0.21 | 0.30 | | | |
| 7 | 0.51 | 0.48 | 0.41 | 0.41 | 0.59 | 0.63 | 1.00 | 0.42 | 0.42 | 0.44 | 0.40 | 0.44 | 0.35 | 0.59 | 0.57 | 0.42 | 0.51 | 0.49 | 0.49 | 0.55 | 0.49 | 0.41 | 0.41 | 0.26 | 0.32 | | | |
| 8 | 0.44 | 0.42 | 0.42 | 0.24 | 0.20 | 0.20 | 0.21 | 1.00 | 0.42 | 0.24 | 0.27 | 0.24 | 0.05 | 0.20 | 0.20 | 0.12 | 0.44 | 0.42 | 0.42 | 0.44 | 0.42 | 0.42 | 0.24 | 0.30 | 0.29 | | | |
| 9 | 0.57 | 0.56 | 0.57 | 0.17 | 0.08 | 0.08 | 0.08 | 0.57 | 1.00 | 0.20 | 0.16 | 0.20 | 0.06 | 0.08 | 0.09 | 0.17 | 0.57 | 0.57 | 0.57 | 0.57 | 0.57 | 0.29 | 0.18 | 0.57 | 0.58 | | | |
| 10 | 0.17 | 0.16 | 0.58 | 0.17 | 0.19 | 0.19 | 0.18 | 0.17 | 0.19 | 1.00 | 0.57 | 0.52 | 0.35 | 0.18 | 0.17 | 0.11 | 0.17 | 0.18 | 0.17 | 0.17 | 0.17 | 0.17 | 0.18 | 0.11 | 0.11 | | | |
| 11 | 0.28 | 0.28 | 0.63 | 0.25 | 0.24 | 0.24 | 0.24 | 0.30 | 0.28 | 0.55 | 1.00 | 0.46 | 0.29 | 0.24 | 0.24 | 0.10 | 0.28 | 0.28 | 0.28 | 0.28 | 0.28 | 0.24 | 0.15 | 0.15 | | | | |
| 12 | 0.19 | 0.18 | 0.55 | 0.19 | 0.22 | 0.22 | 0.21 | 0.19 | 0.21 | 0.59 | 0.54 | 1.00 | 0.40 | 0.20 | 0.19 | 0.12 | 0.19 | 0.20 | 0.19 | 0.19 | 0.19 | 0.20 | 0.12 | 0.13 | | | | |
| 13 | 0.42 | 0.42 | 0.35 | 0.20 | 0.27 | 0.30 | 0.30 | 0.20 | 0.20 | 0.35 | 0.36 | 0.35 | 1.00 | 0.27 | 0.27 | 0.27 | 0.42 | 0.44 | 0.44 | 0.47 | 0.44 | 0.24 | 0.20 | 0.20 | 0.44 | | | |
| 14 | 0.49 | 0.47 | 0.50 | 0.49 | 0.62 | 0.62 | 0.64 | 0.49 | 0.52 | 0.52 | 0.47 | 0.52 | 0.10 | 1.00 | 0.62 | 0.24 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.52 | 0.12 | 0.12 | | | |
| 15 | 0.39 | 0.39 | 0.38 | 0.40 | 0.57 | 0.57 | 0.56 | 0.39 | 0.42 | 0.39 | 0.38 | 0.39 | 0.05 | 0.56 | 1.00 | 0.22 | 0.39 | 0.39 | 0.39 | 0.39 | 0.43 | 0.39 | 0.06 | 0.06 | | | | |
| 16 | 0.52 | 0.47 | 0.47 | 0.49 | 0.49 | 0.52 | 0.52 | 0.49 | 0.49 | 0.35 | 0.49 | 0.49 | 0.49 | 1.00 | 0.52 | 0.49 | 0.49 | 0.52 | 0.49 | 0.49 | 0.49 | 0.49 | 0.52 | 0.49 | | | | |
| 17 | 0.56 | 0.55 | 0.43 | 0.28 | 0.31 | 0.31 | 0.29 | 0.45 | 0.44 | 0.28 | 0.28 | 0.24 | 0.28 | 0.28 | 0.22 | 1.00 | 0.55 | 0.55 | 0.56 | 0.58 | 0.32 | 0.28 | 0.33 | 0.44 | | | | |
| 18 | 0.55 | 0.55 | 0.43 | 0.28 | 0.29 | 0.29 | 0.28 | 0.44 | 0.44 | 0.29 | 0.28 | 0.29 | 0.27 | 0.28 | 0.28 | 0.21 | 0.55 | 1.00 | 0.61 | 0.58 | 0.58 | 0.35 | 0.28 | 0.32 | 0.47 | | | |
| 19 | 0.54 | 0.53 | 0.42 | 0.27 | 0.27 | 0.27 | 0.27 | 0.42 | 0.42 | 0.27 | 0.26 | 0.27 | 0.25 | 0.27 | 0.19 | 0.54 | 0.58 | 1.00 | 0.56 | 0.57 | 0.33 | 0.27 | 0.31 | 0.45 | | | | |
| 20 | 0.56 | 0.54 | 0.39 | 0.20 | 0.20 | 0.23 | 0.23 | 0.41 | 0.40 | 0.20 | 0.19 | 0.20 | 0.35 | 0.20 | 0.20 | 0.26 | 0.56 | 0.58 | 1.00 | 0.58 | 0.28 | 0.20 | 0.41 | 0.58 | | | | |
| 21 | 0.54 | 0.53 | 0.42 | 0.27 | 0.27 | 0.27 | 0.27 | 0.42 | 0.42 | 0.27 | 0.26 | 0.27 | 0.25 | 0.27 | 0.19 | 0.56 | 0.56 | 0.57 | 1.00 | 0.33 | 0.27 | 0.31 | 0.45 | | | | | |
| 22 | 0.56 | 0.55 | 0.49 | 0.41 | 0.41 | 0.39 | 0.56 | 0.57 | 0.47 | 0.47 | 0.47 | 0.25 | 0.39 | 0.42 | 0.31 | 0.56 | 0.58 | 0.58 | 0.58 | 1.00 | 0.47 | 0.39 | 0.41 | | | | | |
| 23 | 0.51 | 0.49 | 0.49 | 0.51 | 0.43 | 0.43 | 0.41 | 0.51 | 0.53 | 0.53 | 0.39 | 0.53 | 0.09 | 0.43 | 0.41 | 0.21 | 0.51 | 0.51 | 0.51 | 0.51 | 0.51 | 0.51 | 0.51 | 0.21 | 0.23 | | | |
| 24 | 0.54 | 0.50 | 0.50 | 0.19 | 0.10 | 0.10 | 0.12 | 0.54 | 0.52 | 0.19 | 0.17 | 0.19 | 0.09 | 0.10 | 0.10 | 0.21 | 0.54 | 0.52 | 0.52 | 0.52 | 0.52 | 0.19 | 0.19 | 1.00 | 0.65 | | | |
| 25 | 0.50 | 0.49 | 0.36 | 0.18 | 0.19 | 0.19 | 0.18 | 0.36 | 0.37 | 0.19 | 0.18 | 0.19 | 0.29 | 0.18 | 0.18 | 0.23 | 0.50 | 0.53 | 0.53 | 0.53 | 0.25 | 0.19 | 0.43 | 1.00 | | | | |

Figure B.6: Evaluation scores for Dataset 3

| Nr | Example Question Types |
|----|---|
| 0 | wie spielte Deutschland bei der Qualifikation |
| 1 | wie spielte Deutschland bei der Qualifikation |
| 2 | wie spielte Deutschland bei der Qualifikation |
| 3 | wie spielte Deutschland gegen Zypern bei der Qualifikation |
| 4 | welche Tore gab es im Spiel Deutschland gegen Zypern bei der Qualifikation |
| 5 | welche gelben Karten gab es im Spiel Deutschland gegen Zypern bei der Qualifikation |
| 6 | wer schoss die Tore im Spiel Deutschland gegen Zypern bei der Qualifikation |
| 7 | wo fand das Spiel Deutschland gegen Zypern bei der Qualifikation statt |
| 8 | wie spielte Deutschland bei Freundschaftsspielen |
| 9 | wie spielte Deutschland gegen Daenemark bei Freundschaftsspielen |
| 10 | wer schoss die Tore im Spiel Deutschland gegen Daenemark bei Freundschaftsspielen |
| 11 | welche Tore schoss Nicklas Bendtner bei Freundschaftsspielen |
| 12 | welche Tore schoss Bendtner bei Freundschaftsspielen |
| 13 | welche Tore schoss Ballack bei Freundschaftsspielen |
| 14 | welche Tore schoss Ballack bei der Qualifikation |
| 15 | welche Tore schoss Podolski bei der Qualifikation |

Table B.2: Dataset 2 for evaluating the Semantic Density approach.

| Nr | Example Question Types |
|----|--|
| 0 | gegen wen spielte Deutschland in Spanien |
| 1 | gegen wen spielte Frankreich bei der WM 1998 |
| 2 | in welchen Spielen schoss Michael Ballack ein Tor |
| 3 | wann fand die Weltmeisterschaft in Italien statt |
| 4 | wann war Brasilien das letzte Mal Weltmeister |
| 5 | wann war Brasilien das letzte Mal Weltmeister |
| 6 | wann war Brasilien Weltmeister |
| 7 | welche Spiele finden in Stuttgart statt |
| 8 | welche Spiele hatten mehr als 120000 Zuschauer |
| 9 | welche Tore schoss Michael Ballack |
| 10 | welche Tore schoss Michael Ballack bei der WM 2002 |
| 11 | welche Tore schoss Ronaldo |
| 12 | wer schoss die Tore im Spiel Deutschland gegen Brasilien |
| 13 | wer war 1990 Weltmeister |
| 14 | wer war zwischen 1950 und 1965 Weltmeister |
| 15 | wer wurde in Mexiko Weltmeister |
| 16 | wie spielte Brasilien 1974 |
| 17 | wie spielte Deutschland bei der WM 1978 |
| 18 | wie spielte Deutschland gegen Brasilien |
| 19 | wie spielte Deutschland gegen Brasilien |
| 20 | wie spielte Deutschland gegen Schweden bei der WM 1974 |
| 21 | wo fanden die Spiele bei der WM in Deutschland statt |
| 22 | wo fand die WM 1990 statt |
| 23 | welche Spiele laufen gerade |
| 24 | wie steht es gerade zwischen Deutschland und Polen |

Table B.3: Dataset 3 for evaluating the Semantic Density approach.

B.8 Example of a (q,a) pair in RDF

Question 4: Wann fand die WM in Italien statt?

```
<j.6:FIFAWoRLdCup rdf:about=".../individuals/spin#s1317">
<j.0:HAPPENS-AT>
<j.0:time-interval rdf:about=".../individuals/spin#s1320">
<j.0:BEGINS>
<j.0:time-point rdf:about=".../individuals/spin#s1321"/>
</j.0:BEGINS>
</j.0:time-interval>
</j.0:HAPPENS-AT>
<j.6:heldIn>
<j.3:Country rdf:about=".../individuals/spin#s1318">
<j.0:HAS-DENOMINATION>
<j.3:country-denomination rdf:about=".../individuals/spin#s1319">
<j.0:NAME rdf:datatype=".../XMLSchema#string">ITALIEN</j.0:NAME>
</j.3:country-denomination>
</j.0:HAS-DENOMINATION>
</j.3:Country>
</j.6:heldIn>
</j.6:FIFAWoRLdCup>
```

Answer Example:

```
<j.6:Match rdf:about=".../agents/instances#id_116902326">
<j.6:team rdf:resource=".../agents/instances#id_116902590"/>
<j.6:inTournament>
<j.6:FIFAWoRLdCup rdf:about=".../agents/instances#id_116902328">
<j.6:heldIn>
<j.3:Country rdf:about=".../agents/instances#id_116902331">
<j.0:HAS-DENOMINATION>
<j.3:country-denomination rdf:about=".../agents/instances#id_116902332">
<j.0:NAME>Spanien</j.0:NAME>
</j.3:country-denomination>
</j.0:HAS-DENOMINATION>
</j.3:Country>
</j.6:heldIn>
</j.6:FIFAWoRLdCup>
</j.6:inTournament>
<j.0:HAS-DENOMINATION>
<j.0:denomination rdf:about=".../agents/instances#id_116902327">
<j.0:NAME>Deutschland (FRG)-Algerien</j.0:NAME>
</j.0:denomination>
</j.0:HAS-DENOMINATION>
<j.6:heldIn>
<j.6:Stadium rdf:about=".../agents/instances#id_116902333">
<j.0:HAS-DENOMINATION>
<j.0:denomination rdf:about=".../agents/instances#id_116902334">
<j.0:NAME>El Molinon</j.0:NAME>
</j.0:denomination>
</j.0:HAS-DENOMINATION>
</j.6:Stadium>
</j.6:heldIn>
<j.0:HAPPENS-AT>
<j.0:time-interval rdf:about=".../agents/instances#id_116902337">
```

```
<j.0:BEGINS>
  <j.0:time-point rdf:about=".../agents/instances#id_116902338">
    <j.0:YEAR>1982</j.0:YEAR>
    <j.0:MONTH>JUN</j.0:MONTH>
    <j.0:DAY>16</j.0:DAY>
  </j.0:time-point>
</j.0:BEGINS>
</j.0:time-interval>
</j.0:HAPPENS-AT>
<j.6:matchResult>1:2 (0:0)</j.6:matchResult>
<j.6:attendance>42000</j.6:attendance>
</j.6:Match>
```

Appendix C

Appendix of Chapter 7

C.1 Semantic Cloud Evaluation

| Nr | Question |
|----|--|
| Q1 | The user interface was fast and easily understood. |
| Q2 | The interface supported be in finding interesting resources. |
| Q3 | It was pleasant/entertaining to look for websites with this interface. |
| Q4 | I would use this interface to find interesting websites. |

Table C.1: User Evaluation Questions.

List of Abbreviations

| Abbreviation | Description | Definition |
|--------------|---|------------|
| ACC | Agent Communication Channel | page 153 |
| ACL | Agent Communication Language | page 213 |
| AMS | Agent Management System | page 153 |
| CI | Confidence Interval | page 255 |
| DC | Dublin Core | page 61 |
| DEL | Delicious | page 250 |
| DF | Directory Facilitator | page 153 |
| DOF | Degrees Of Freedom | page 260 |
| DOLCE | Descriptive Ontology for Linguistic and Cognitive Engineering | page 149 |
| DOM | Document Object Model | page 21 |
| EMMA | Extensible MultiModal Annotation markup language | page 147 |
| FIFA | Fédération Internationale de Football Association | page 161 |
| FIPA | Foundation for Intelligent Physical Agents | page 152 |
| FOAF | Friend Of A Friend | page 61 |
| (G)UI | (Graphical) User Interface | page 72 |
| GWAP | Games With A Purpose | page 120 |
| HC | Human Computation | page 104 |
| HCI | Human Computer Interaction | page 79 |
| HTML | Hypertext Markup Language | page 14 |
| HTTP | Hypertext Transfer Protocol | page 25 |
| IE | Information Extraction | page 53 |
| IR | Information Retrieval | page 50 |
| IS | InstanceScore Algorithm | page 201 |
| JEFF | Java Extraction Facilitator Framework | page 174 |
| LMB | Left Mouse Button | page 261 |

| Abbreviation | Description | Definition |
|--------------|---|------------|
| LOD | Linked Open Data | page 31 |
| MAS | Multi Agent System | page 151 |
| MMB | Middle Mouse Button | page 261 |
| MTS | Message Transport Service | page 153 |
| NLP | Natural Language Processing | page 37 |
| OWL | Web Ontology Language | page 27 |
| PS | ProperScore Algorithm | page 201 |
| QA | Question Answering | page 126 |
| RDF | Resource Description Framework | page 24 |
| RDF-S | Resource Description Framework - Schema | page 27 |
| RMB | Right Mouse Button | page 261 |
| SeRQL | Sesame RDF Query Language | page 211 |
| SGML | Standard Generalized Markup Language | page 19 |
| SI | Semantic Interaction | page 82 |
| SIM | Semantic Interaction Metaphor | page 82 |
| SKOS | Simple Knowledge Organization System | page 61 |
| SPIN | SPeech INput | page 188 |
| SSB | Semantic Space Brower | page 250 |
| SWA | Semantic Wrapper Agents | page 143 |
| SWIntO | SmartWeb Integrated Ontology | page 143 |
| SUMO | Suggested Upper Merged Ontology | page 149 |
| TF-iDF | Term Frequency - inverse Document Frequency | page 37 |
| TTS | Text To Speech | page 145 |
| URI | Unified Resource Identifier | page 24 |
| URL | Unified Resource Locator | page 15 |
| XML | Extensible Markup Language | page 19 |
| XPATH | XML Path Language | page 21 |

List of Figures

| | | |
|------|---|----|
| 1.1 | Thesis structure. | 11 |
| 2.1 | HTML document (left) and its tag tree based on DOM (right).. | 14 |
| 2.2 | Classification of HTML elements. | 14 |
| 2.3 | A social network. Actor i is the most central one. | 16 |
| 2.4 | Web page hyperlink structure (in/out-links). | 17 |
| 2.5 | HTML tag sequence. Data elements are denoted by PCDATA. . | 18 |
| 2.6 | Example of typical (semi-)structured web pages. | 22 |
| 2.7 | A sample type tree representing a book. | 23 |
| 2.8 | HTML fragment, boundary coordinates and its DOM tree. . . . | 24 |
| 2.9 | Semantic Layering using RDFa. | 25 |
| 2.10 | A simple RDF graph - nodes and directed edges. | 26 |
| 2.11 | Adding machine-processable semantics to web data (Mika, 2007). | 27 |
| 2.12 | The Semantic Web Stack of the W3C (Dengel, 2011). | 28 |
| 2.13 | Example ontology describing (part of) a soccer domain. | 30 |
| 2.14 | Power law distribution in folksonomy systems. | 33 |
| 2.15 | Scaling of social tagging systems. | 34 |
| 2.16 | Resources, Tags and Users. | 35 |
| 3.1 | Wrapper life cycle (Lerman et al., 2003). | 54 |
| 3.2 | PAT tree representation of HTML code. | 58 |
| 3.3 | Wrapper based on Union Free Regular Expressions. | 59 |
| 3.4 | Ontologies organized according to level of semantics (Mika, 2007) | 61 |
| 3.5 | RDF graph of an example semantic assertion. | 63 |
| 4.1 | Metaphors in HCI (Fineman, 2004, pg.10). | 81 |
| 4.2 | Enhanced IR model. | 81 |
| 4.3 | Semantic Layering and Semantic Interaction. | 88 |
| 4.4 | Semantic Interaction design pattern. | 90 |

| | | |
|------|---|-----|
| 4.5 | Levels of semantic interaction via ontologies. | 92 |
| 4.6 | Social semantic tagging of structured data. | 96 |
| 4.7 | Activated nodes of associations in a network (Reinsberg, 1997). | 98 |
| 5.1 | Caption for LOF | 103 |
| 5.2 | Highlighting and sticky notes using the Diigo Tool. | 106 |
| 5.3 | Annotation of a semantic structure on a web page as N-triples. . | 107 |
| 5.4 | Workflow for creating emergent semantic structures from tags. . | 108 |
| 5.5 | tag2wrap Annotations in RDF via XPATH. | 111 |
| 5.6 | tag2wrap UI - single (left) and box selection (right). | 111 |
| 5.7 | Consolidation of tag structures. | 113 |
| 5.8 | Tag distribution in the experiment approximating power law. . . | 115 |
| 5.9 | Results of the Usability Evaluation. | 117 |
| 5.10 | Basic Architecture of Human Computation games. | 121 |
| 5.11 | Playful Tagging using Binary Verification. | 124 |
| 5.12 | Tag Generation Screen. | 125 |
| 5.13 | Agreement Screen. | 126 |
| 5.14 | A sample web page and associated questions. | 128 |
| 5.15 | The Webpardy “wall” UI for selecting category/difficulty level. . | 129 |
| 5.16 | Entering questions for given web page resources/fragments. . . . | 130 |
| 5.17 | Generating question-answer pairs in Webpardy. | 131 |
| 5.18 | Double Webpardy Mode for validating entered results. | 133 |
| 6.1 | SmartWeb - system architecture overview. | 144 |
| 6.2 | “Who won the soccer world championship in 1990?” as RDF. . | 148 |
| 6.3 | FIPA reference model of a MAS. | 153 |
| 6.4 | Visual wrapping - workflow. | 158 |
| 6.5 | LingInfo example for the soccer ontology (Buitelaar et al., 2006). . | 159 |
| 6.6 | Hierarchies in a soccer match (left) and news article (right). . . | 160 |
| 6.7 | Adaption of selected and tagged HTML-Fragment. | 162 |
| 6.8 | Example of square detection (HTML news from Yahoo.) | 163 |
| 6.9 | Square Tree for the Yahoo news example. | 166 |
| 6.10 | Attributed grammar rule definition. | 168 |
| 6.11 | Conceptual hierarchies in the dmag-news site. | 170 |
| 6.12 | Rule Tree Example. | 174 |
| 6.13 | JEFF - Semantic wrapper generator UI. | 175 |
| 6.14 | Atomic and grouping tagging dialogues. | 176 |
| 6.15 | JEFF - visual wrapper creation dialogue. | 176 |
| 6.16 | Wrapper definition in XML. | 177 |
| 6.17 | Aggregated results of P,R and F-Score for the FIFA corpus. . . | 182 |

| | | |
|------|---|-----|
| 6.18 | Mean results of P,R and F-score for the dm-tzi news pages. | 184 |
| 6.19 | Mean results of P,R and F-score for the yahoo news pages. | 185 |
| 6.20 | “Ballack scored a goal example as XML. | 189 |
| 6.21 | “Ballack scored a goal example as SWIntO instance. | 191 |
| 6.22 | A goal scoring example | 194 |
| 6.23 | The player encoding of the goal scoring example. | 194 |
| 6.24 | Evaluation scores of PS for 14 (q,a) pairs. | 204 |
| 6.25 | Evaluation scores of IS for 14 (q,a) pairs. | 205 |
| 6.26 | Evaluation scores of PS' for 14 (q,a) pairs. | 205 |
| 6.27 | Big Picture showing the Semantic Wrapper Agents approach. . | 211 |
| 6.28 | FIPA-ACL request message. | 212 |
| 6.29 | FIPA-ACL inform message. | 213 |
| 7.1 | Semantic Interaction in visual retrieval interfaces. | 221 |
| 7.2 | Related tags for the tag <i>recipes</i> with different metrics. | 228 |
| 7.3 | Semantic Cloud Topic Clustering Example. | 230 |
| 7.4 | UI of Semantic Cloud - general overview cloud. | 232 |
| 7.5 | UI of Semantic Cloud - hierarchical exploration. | 233 |
| 7.6 | Background knowledge of the 9 subjects. | 235 |
| 7.7 | The radial semantic cloud with document clusters. | 244 |
| 7.8 | Changing Perspective in the radial Semantic Tag Cloud View. . | 246 |
| 7.9 | Spatial Metaphors for 3D-visualization of document clusters. . | 247 |
| 7.10 | Aggregated mean scores for the four tested categories. | 252 |
| 7.11 | Aggregated mean scores for 3D Visualization and Semantics. . | 254 |
| 7.12 | Distribution of user ratings comparing both tag clouds. | 254 |
| 7.13 | CI for the different interaction categories compared. | 255 |
| 7.14 | Navigating via <i>ElasticSteer</i> on an interactive surface. | 261 |
| 7.15 | Mean values for tested parameters: mouse vs. touch. | 262 |
| 7.16 | Mean values for tested parameters: mouse vs. touch. | 263 |
| 7.17 | Mean values for tested parameter ct: free vs. constrained. . . . | 263 |
| 7.18 | Mean values for tested parameters: free vs. constrained. | 264 |
| B.1 | Example - Research group Digital Media news page. | 305 |
| B.2 | Jedi Grammar Generation in JEFF. | 307 |
| B.3 | “When was Brazil soccer world champion?” as SeRQL query. . | 308 |
| B.4 | Evaluation scores for Dataset 1 | 311 |
| B.5 | Evaluation scores for Dataset 2 | 312 |
| B.6 | Evaluation scores for Dataset 3 | 312 |

List of Tables

| | | |
|------|---|-----|
| 2.1 | Tag co-occurrence matrix with absolute counts. | 39 |
| 2.2 | Classification vs. Observation. | 45 |
| 4.1 | Systems, access methods and interaction metaphors. | 90 |
| 5.1 | Selection types, interactions and structural information. | 109 |
| 5.2 | Number of marked resources in the test web pages. | 114 |
| 5.3 | Analysis of web page structure in the five domains (Appendix A.1). | 115 |
| 5.4 | Assigned tags for “James Bond” resource. | 116 |
| 5.5 | Analysis of Tags and false assigned WordNet synsets. | 117 |
| 5.6 | Input-Output Relations in a HC Game for semantic annotation. | 122 |
| 5.7 | Number of sections per topic field. | 134 |
| 5.8 | Evaluation of the data collected 2011. | 135 |
| 5.9 | Double Webpardy election example. | 136 |
| 5.10 | Average scores from the 2012 experiment(s). | 136 |
| 5.11 | Most prominent question types. | 137 |
| 5.12 | Some interesting sample questions entered into Webpardy. . . . | 138 |
| 6.1 | A sample dialogue in turns in SmartWeb. | 145 |
| 6.2 | A sample semantic translation of a user question. | 146 |
| 6.3 | RDF statements describing a football match result. | 161 |
| 6.4 | Detected square pattern candidates and parameters. | 164 |
| 6.5 | Complex grammar productions. | 169 |
| 6.6 | Characteristics of the used datasets. | 179 |
| 6.7 | Overall results for the semantic structure <code>footballMatch</code> | 181 |
| 6.8 | FIFA Wrapper Evaluation Results. | 181 |
| 6.9 | News domain wrapper evaluation. | 183 |
| 6.10 | News wrapper evaluation results (P/R/F-score). | 183 |

| | |
|--|-----|
| 6.11 Transformation rules | 192 |
| 6.12 Semantic similarity scoring matrix. | 197 |
| 6.13 A sample question-answer pair as N-triples. | 200 |
| 6.14 Questions used for the evaluation of the ProperScore algorithm. | 202 |
| 6.15 ProperScore results for Dataset 1 (Appx. B.7). | 203 |
| 6.16 Ranking results (PS) for the two other datasets. | 206 |
| 6.17 Baseline results (AnswerScore) for Dataset 1. | 207 |
| 6.18 Semantic similarity scoring matrix for selecting wrappers. | 209 |
| 7.1 Average ratings for different similarity metrics. | 228 |
| 7.2 Mean rating, standard deviation and t-Test results. | 236 |
| 7.3 Tasks for Delicious. | 249 |
| 7.4 Tasks for the Semantic 3D Browser. | 250 |
| 7.5 Excerpt of the 3D Section in the QVIS questionnaire. | 251 |
| 7.6 Mean rating, standard deviation and t-Test results. | 253 |
| 7.7 ElasticSteer User Study (Mouse vs. Touch). | 264 |
| 7.8 ElasticSteer user study (free vs. constrained navigation). | 265 |
| A.1 Web page types in the web experiment. | 303 |
| A.2 Questionnaire for the web experiment. | 303 |
| B.1 Rule naming conventions. | 307 |
| B.2 Dataset 2 for evaluating the Semantic Density approach. | 313 |
| B.3 Dataset 3 for evaluating the Semantic Density approach. | 314 |
| C.1 User Evaluation Questions. | 317 |