Topic Driven Access
to
Scientific Handbooks

Caterina Caracciolo

Topic Driven Access
to
Scientific Handbooks

Promotiecommissie
Promotor: Prof.dr. M. de Rijke
Co-promotor: Dr. J. Kircz
Co-promotor: Prof.dr. G. Tamburrini
Overige leden: Prof.dr. J. van Eijck
Prof.dr. F. van Harmelen
Dr.ir. J. Kamps
Prof.dr. J. Mackenzie Owen
Dr. L. Serafini
Prof.dr. B. Wielinga

Faculteit der Natuurwetenschappen, Wiskunde en Informatica.

SIKS Dissertation Series No. 2008-13
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

The research reported in this thesis was carried out at the Informatics Institute of the University of Amsterdam and supported by Elsevier, Amsterdam.

Printed and bound by Delta 2000, Roma

ISBN: 978–90–5776–176–8

Ai luoghi della vita

e alle biciclette che li attraversano.

Acknowledgments

To my surprise, the moment has arrived. I am writing the ackowledgments of my thesis and I think back on the period of my life that I now call “Amsterdam.” Now, in the sunny, warm and chaotic Rome, all the study, work and life that I had in Amsterdam seem so far away that it is surprising that it all happened. But it did happen, for real.

I would like to thank many, many people and let them know that they have been so important to me. This is the occasion to do so.

The first person to mention is Rosella Gennari, without whom I would have never heard of an opening position at the ILLC department of the UvA and I would not be writing these words now. That position was under the supervision of Maarten de Rijke, somebody I only had a vague memory of because of one class I attended when I was an undergraduate exchange student in Amsterdam. I went to that class only out of curiousity, who would have thought he was going to play such an important role in the years to come. Maarten has been a great example and a continuous surprise to me, because of his quick mind and kind manners, music taste and approach to life. With Maarten I learned an uncountable number of things, all of which will stay with me forever. Together with Maarten, Joost Kircz has followed the entire development of this work. Inspiring conversations, vision, human touch: I had the unique chance to enjoy all of these from him. Guglielmo Tamburrini was my supervisor when I started my PhD in Pisa, then he kindly agreed to continue to follow my work when I moved to Amsterdam and when he moved to Naples.

Other people have been influential to the completion of this thesis. With Frank van Harmelen I learned to adopt a healthy perspective on things and to stand for one’s own point of view; Jan van Eijck thought me to appreciate the humbleness and the intellectual courage that should always accompany research work; the Elsevier team showed me the importance of good coordination to carry out projects of any size. All of them showed me the importance of setting oneself a goal to achieve and keeping it despite any interferences.

I would like to thank all the members of my doctoral committee, for accepting to read the manuscript, providing insightful comments and so contributing to make my thesis a better piece of work: Jan van Eijck, Frank van Harmelen, Jaap Kamps, John Mackenzie Owen, Luciano Serafini, Bob Wielenga.

A special thanks to Arjen de Vries for generously reading early versions of different chapters of my thesis and commenting on them. Willem van Hage also deserves special thanks, not only for contributing to this work as an undergraduate student, but also for his friendly and enthusiastic presence. David Ahn and Stefan Schlobach read several versions of chapters of this thesis and commented on content and presentation. Maarten van Schie also contributed to this work as an undergraduate student. Andrew Bagdanov read nearly all versions of this manuscript, proofread everything and improved so much the quality of my English.

I shared a large part of the journey that lead to this thesis with dear friends, of inestimable value: Juan Heguiabehere, Gabriel Infante Lopez and Gabriele Musillo. Their presence, or even the simple thinking of them always warmed me to the true depths of my soul. Veronica Prada Moroni stayed close to me despite the physical distance between Pisa and Amsterdam — she knows every fold of this story, perhaps better than myself.

Vorrei ringraziare Claudia, e Tobias e Frida (e ora Malvina!), per la loro sempre calda accoglienza e la paperella gialla; Roberta, e Andrea e Diego, per le conversazioni, le risate e le gite a Napoli e Bologna; Francesca per il suo continuo sostegno e la stima che mi ha sempre affettuosamente dimostrato; Roberto Lalli con Haci Ciugo e la vita Baires; Laura, Alessandro, Tiziana e Silvia per essere sempre stati così indulgenti verso le mie fugaci apparizioni a Pescara.

My heartfelt hugs go to Rosella Gennari, Carlos Areces, David Ahn, Raffaella Bernardi, Paola Monachesi, Stefano Bocconi, Daniele Moroni, Martina Dominici; to the spanish crowd I spent many evenings and nights with; to all my colleagues at the ILLC and at the ILPS; to Fransje Enserink who is not here anymore; to Virginie and Pauline who helped me out from a distance.

And Rome? Here I found breathtaking beauty, the contradictions that keep one's spirit alive and a great working evironment at the FAO, the Food and Agriculture Organization of the UN. In Rome, I was so lucky to find so many "new" friends: Giorgia, Federica e Simona, ManuManu Manuela, Valentina, Natalia e Francesca, Pietro, Claudio & la splendida Livia, Federico M. and Massimo, Tommaso e Daniela, the fabolous Occhionero and the whole "Ciclopicnic Entertainment," the roman Critical Mass and la Ciemmona Interplanetaria. A big happy smile goes to all of them.

La mia famiglia è sempre con me e la distanza non conta. I miei genitori e i miei fratelli Eligio, Roberto, Paola e Carlo, e le loro famiglie, Barbara, Lucrezia Luna e Francesco, Teresa e Eugenia: vorrei potervi abbracciare tutti in un solo abbraccio.

Finally, Andy. Insieme ad Amsterdam, Pisa, Firenze, Roma e chissà dove ancora. You always believed this moment would come, even when I myself would doubt. Thank you for this and much much much more.

Roma, marzo 2008

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Mary Ann has to write an essay on Gödel's theorem. Prof. Dickenson needs to check a bibliographic reference. Robert came across the expression 'categorial grammar' and wants to find out what it means. Dr. Schultz is updating the bibliography for an old paper of his. Susan wants to find out what the prominent centers of research in the area of computational linguistics are... What is common to all these scenarios is a question mark: the fact that they can be turned in (at least) one question, either vague or very sharp, simple or complex, ambitious or minimalist. For the rest, each of the above situations involve different people, with different backgrounds, aims, expectations about the type of answer to get, and in what time. Also, their ability to select and interpret answers, and their notions of relevance and satisfaction differ.

What is usually called an information need, what pushes a person to look for information, is the result of all these factors. When using a search engine, or OPAC system, the user has to type in a *query*, a phrase that instantiates the user's information need. Queries are not complete sentences, but phrases containing the salient terms the person is interested in. Formulating a query implies knowing what one is looking for, and formulating a good query implies having an idea of what documents satisfying the need should look like. Therefore, the query is a fundamental component of an *information searching* process, but information search is included in the more general process of *information seeking*, which is one aspect of a person's *information behaviour* (Figure 1.1). Many models have been proposed [Dervin, 1983, Ingwersen, 1984, Marchionini, 1995, Wilson, 1994] to describe what happens when a user seeks information, starting from the time the information need is perceived, and going through all passages necessary to achieve the goal (or to give up on it). Some of them (e.g., [Wilson, 1997]) also include a temporal dimension in their model for information behavior: there, the "active search" is the last component of the (iterative) process of information seek-

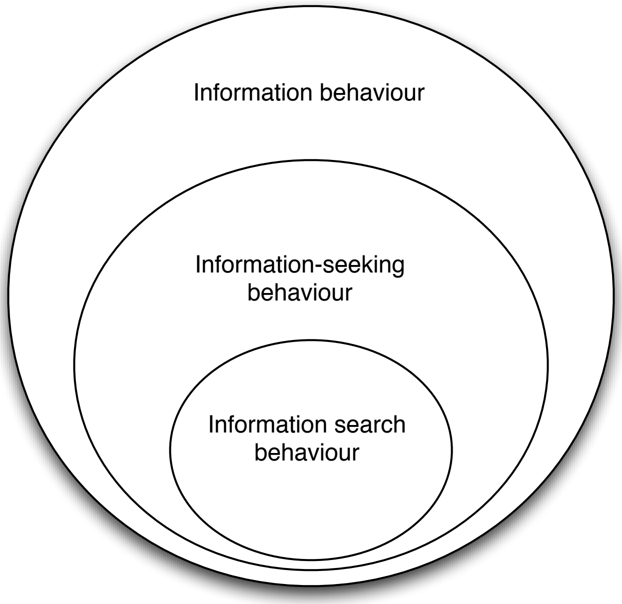


Figure 1.1: Nested model of information behavior, according to Wilson [1999].

ing. The point to make here is that the information seeking behavior has search as an important, but not unique, component. The intertwined browsing, exploring, reading activities all play a role in satisfying an information need, and should be supported in a good electronic publishing environment. In such an environment, the reader will be naturally brought "to" and "inside" the text.

In this thesis, the type of text we are interested in is scientific handbooks, a type of text that has so far received relatively little attention from the library science or information retrieval community. We address the issue of providing topic driven access to scientific handbooks, and we do so by adopting a broad perspective. We present a model that includes both browsing and searching as fundamental components, and that builds on ideas and approaches developed in the areas of library science and information retrieval.

1.1 Problem Statement

The tradition of the scientific handbook as a concise, accessible source of validated information emerged in the late nineteenth century when the factual burden of scientific and medical subjects began to overwhelm students. Nowadays, handbooks are "fat" volumes that typically contain long chapters, written by many authors and of several dozen pages each, often without a standardized structure, with the aim of providing a comprehensive overview of a scientific discipline. Now, in the electronic age, access to these (electronic) scientific handbooks themselves has become an issue requiring attention.

Users of scientific handbooks may have a specific information need, which leads them to read up on the specific topic and even on a specific aspect of it. If this is the case, such readers want to avoid having to read or scroll dozens of pages, they rather need a way to "jump" to specific excerpts of the handbook covering the topic they are looking for. This is what we call *focused access* to the text: when the reader is brought directly inside the text, to the specific excerpt where she can find the information she is looking for. In other cases, users of handbooks may have vaguer information needs, more related to the need or desire to get a more general, higher level picture of the domain.[1] In the former case, the user is likely to be able to produce a query, while in the latter case this ability may be hampered by poor knowledge of the domain.

These considerations naturally lead one to think of an environment where the user is not only able to type in queries, but can also take advantage of an organization of the material that can meet her vague information needs. In reference works, such as dictionaries and encyclopedias, subjects are usually arranged in alphabetical order so that they may be located quickly and easily. In books and handbooks, the subject is organized into chapters, and usually an index is provided to serve as the direct guide to the many topics treated in it, or to locate the smaller subdivisions of the larger subjects. The traditional back-of-the-book index may also be organized in several disjoint parts, such subject index, author index, name index, etc. Another important element in the arrangement of material in a reference work is the *cross-reference* that will refer the reader to additional related information. From the back-of-the-book index, then, comes the inspiration to organize the subject in a way that is informative for users with vague information needs and a limited background in the area. In this setting, it is natural to think of a high-level *map* of a domain, a searchable map containing topics and relations between them, as well as appropriate locations inside the text. Then the user can "land on" the map, either by search or by navigation, and then zoom in on the topics that best fit her information need. Therefore, we propose a map of the domain, that beyond containing references to the appropriate locations in the text, also includes relations between the elements in it — that we call *topics*.[2] So, the fundamental question is: can we gain anything from providing a map enriched with links to access scientific handbooks?

To make matters more concrete and tangible in the thesis, we work with a specific domain and handbook: the domain is the interface of logic and linguistics, and the handbook that we use as our test case is the *Handbook of Logic and Language* [van

[1] We do not consider here the case of consecutive reading: this does not mean that it is not an important type of reading or, worse, that it has been put out of fashion by the advent of the "electronic age." On the contrary, its role for educational purposes cannot be overestimated, but in our work we simply look at a different usage of scientific handbooks.

[2] Note that *topic maps* [TOPICMAPS, 2006] is an ISO standard for describing knowledge structures and associating them with information resources. It is essentially a format for data representation and can only express two types of *associations* between topics: *class-instance* and *superclass-subclass*. In this sense the LoLaLi map of topics is very different from a topic map.

Benthem and ter Meulen, 1997]. In this domain we identify important topics and relationships between those topics; these are used to build a browsable map, which is then presented to the reader as an interface to the handbook.

Given this approach, that we explain in more detail in Chapter 2, the research questions we address in this thesis are the following:

---

1. RESEARCH QUESTION. What requirements should we impose on a map that is to be used for human browsing and as a skeleton to provide focused access to the text of a scientific handbook?

In the case of the *Handbook of Logic and Language*, Research Question 1 comes with an important constraint: the process of populating the browsable map should be as much as possible a collaborative and bottom-up process. There are several reasons for this constraint, some principled, some pragmatic. The interface of logic and language is evolving rapidly, and is highly interdisciplinary. Moreover, unlike, say, medicine or law, there are no resources (or standard bodies) to support imposing a standard in a top-down manner: (expert) colleagues around the world, and from around the interdisciplinary area covered by the *Handbook of Logic and Language*, will be the ones populating the browsable map with topics and relations between them.

---

2. RESEARCH QUESTION. How do we present the map to readers of a handbook in such a way that we ensure broad coverage of the domain (with detailed information per topic), while also making sure that users do not get lost?

The answer to Research Question 2 will be a user interface that is able to accommodate detailed pieces of information about topics in the map, while at the same time giving the user ease of navigation and a good sense of the "general picture." The interface should also be able to equally support searching and browsing in order to allow users to satisfy a broad variety of information needs, from the more specific (search) to the more vague (browsing), and user background.

Assuming that we are able to come up with a satisfactory answer to Research Question 2, we need to connect the browsable map to the documents of which it is meant to provide a map. The next question, then, is:

---

3. RESEARCH QUESTION. What are suitable targets in the handbook to establish focused, topic driven links from topics in our browsable map?

Notice that in Research Question 3 we ask for *focused* links from topics in the browsable map into the handbook, and not just for links from a topic to an entire chapter of dozens of pages. A focused link, then, should be an excerpt of the document, *readable* despite its separation from the whole of the document, and relevant to the topic it talks about. Assuming there are different ways to identify excerpts that could provide the basis for focused links, we should find out which is the most appropriate. We phrase this desideratum as follows:

4. Research Question. What is the most suitable type of candidate link to be connected to the map?

An obvious key issue along the way will be *evaluation*: how do we assess our proposals? At different stages of our work, different types of evaluation are appropriate. When we introduce our proposal for a browsable map that generalizes the concept of the back-of-the-book index to the setting of electronic scientific handbooks, we compare the results we obtained with the requirements we set ourselves, but also with the internal organization of the map. Later, we perform user studies to assess the effectiveness of our proposed visualization method for exploring the map. A third type of evaluation is conducted when we link the topic in our browsable map to candidate targets in the handbook; this is the type of evaluation that one encounters in information retrieval and applied language technology, where one develops "gold standard" corpora of ideal outcomes and measures the performance of algorithms against this yardstick. Since no standard test sets exist for the issues that we are tackling, we develop our own.

Throughout the thesis, we will have an additional "meta-concern" on our mind: assuming that our proposed model for accessing electronic scientific handbooks is an effective one, how are the roles of authors and editors affected?

5. Research Question. Can the scenario proposed in this thesis be adopted as a basis for the production of new handbooks and, if so, would this imply a change in the roles of authors and editors of handbooks? Will they be expected to populate the envisage browsable maps? Will they have to write differently, knowing that a map will be linked to their texts?

1.2 Organization of this Thesis

In this thesis we proceed in a step-by-step manner by providing answers to each of the research questions listed above. Figure 1.2 provides a graphical representation of the contents of the thesis. In Chapter 2 we describe our vision of focused access to scientific handbooks: we propose the idea of a browsable map displaying important terms and their interrelations as a suitable interface to electronic scientific handbooks. We also provide general background on the issue of searching for and inside books, and on structures used for indexing and classification purposes.

Chapter 3 addresses Research Question 1. There, we instantiate the proposed model to our case study: the domain of logic and language. We introduce our map, that we call the *Logic and Language Links* map (for short, the LoLaLi map), the way it was built and its constituents, including the hierarchical and non-hierarchical relations between topics. In the same chapter, we also report on the management of the map.

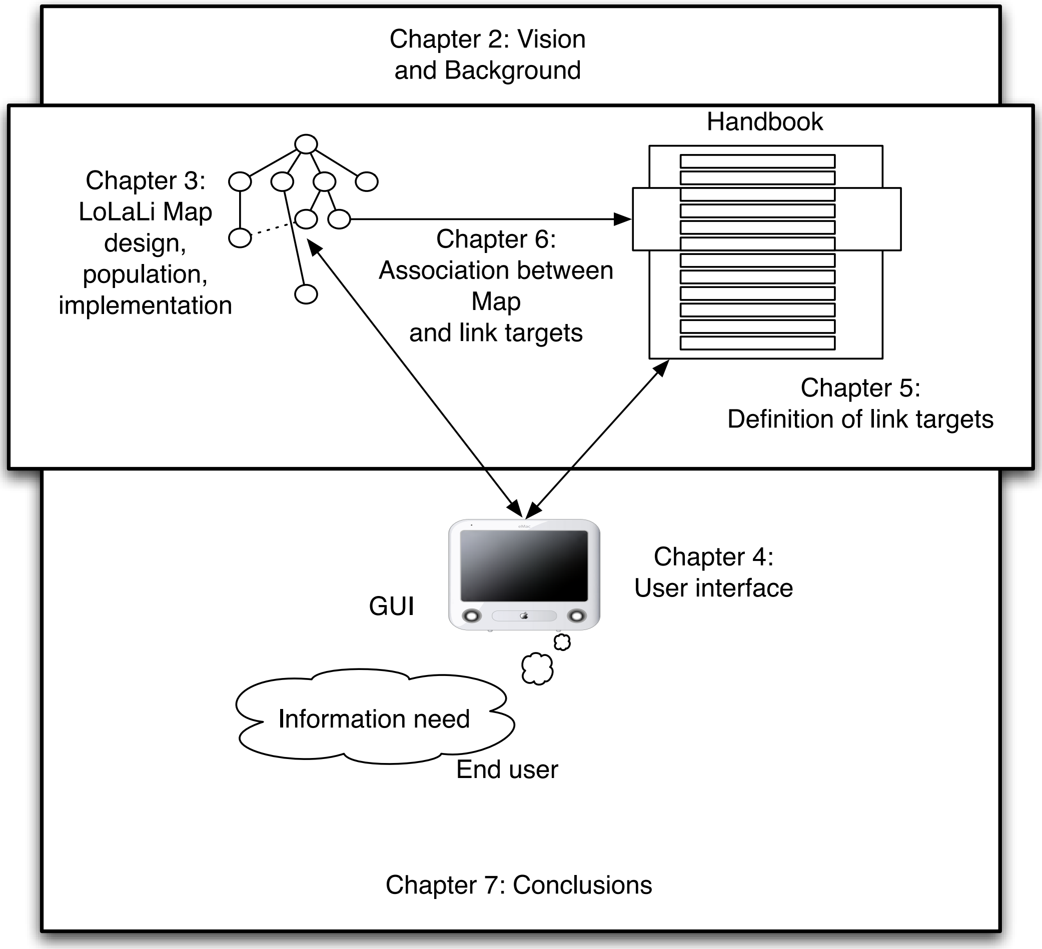


Figure 1.2: A graphical representation of the contents of this thesis.

Then, in Chapter 4 we address Research Question 2. We consider the problem of presenting to end users the browsable map we developed. We first review well-known visualization techniques for semantic structures like ours, then we describe the interface we developed in order to allow end users to navigate and search the LoLaLi map. Finally, we report on the user studies we performed with two groups of students.

In Chapter 5 we concentrate on the selection of text excerpts to be used as candidate link targets for the LoLaLi browsable map, thus addressing Research Question 3. We take into consideration two types of automatic text segmentation techniques, structurally and semantically oriented, and we compare them against a manual annotation of the handbook.

Given the collection of text segments resulting from the experiments described in the previous chapter, in Chapter 6 we address Research Question 4 and look at the matching between these segments and the topics in the map. We discuss the evaluation of that task and propose measures for this purpose. Finally, in Chapter 7 we draw our conclusions by summarizing our answers to Research Questions 1–4, and address Research Question 5. In Chapter 7 we also highlight our plans about future work.

It should be noted that this thesis does not contain a monolithic "Related Work" chapter. We decided against this: as we build on insights from a broad range of disciplines—including library science, user interface design, human computer interaction, information retrieval, and knowledge representation—such a chapter would have been a maze of disconnected discussions. Instead, we review related work and link our own contributions to the literature along the way.

1.3 Scope of this Thesis

In order to address our research questions we build on insights and methods from library science, user interface design, human computer interaction, information retrieval, and knowledge representation. Our work aims at exploring a possible way to provide focused, topic-driven access to scientific handbooks, in this respect it cannot be easily classified under any of these labels. Moreover, we do not aim at providing an end-to-end system. For these reasons, the following issues are, unfortunately, beyond the scope of this thesis.

Production. Many ingredients are necessary to have an end-to-end systems up and running, including a session manager, text encoding, large data management systems, efficient server-client interaction, integration of search over several sources and visualization and rendering. Although necessary, these components are deliberately not included in the framework presented in this thesis.

Publishing. We do not deal with issues such as copyright and intellectual property or business and publishing models for online publication of reference works. We believe these issues are central and need to be addressed so as to promote a wide and democratic circulation of knowledge and thoughts, while guaranteeing authors the possibility of continuing to create intellectual work.

Handbook authoring environments. Although in the concluding chapter we dedicate some space to future electronic environments for authoring handbooks, this thesis mainly deals with the conversion of existing handbooks in order to provide topic access to them.

E-learning. Any handbook is both a reference tool and a learning support tool, and in this thesis we concentrate on accessing electronic handbooks in their function as reference tool. Issues related to the learning function of an electronic handbook, such as cognitive aspects of online reading and online learning, are therefore not treated here.

1.4 Origins of the Material

The material in this thesis is largely based on a number of publications. Chapter 2 includes material published in:

- C. Caracciolo and M. de Rijke (2002). Structured Access to Scientific Information. In *Proc. of Global WordNet Conference*, Mysore (India).

The description of the LoLaLi map given in Chapter 3 is based on previous work published in:

- C. Caracciolo (2003). Towards Modular Access to Electronic Handbooks. *Journal of Digital Information (JODI)*, 3(4), no. 157. URL: `http://journals.tdl.org/jodi/article/view/jodi-104/84`.

- C. Caracciolo (2006). Implementing an Ontology for Logic and Linguistics. *Literary and Linguistic Computing*, 21:29–39.

The prototype used for the user studies described in Chapter 4 used tools developed by W.R. van Hage and described in:

- W.R. van Hage (2004). Living on the Edge. Master thesis, University of Amsterdam, 2004.

The results presented in Chapter 5 have been published in:

- C. Caracciolo, M. de Rijke, and J. Kircz (2002). Towards Scientific Information Disclosure Through Concept Hierarchies. In J. A. Carvalho, A. Huebler, A. Baptista, editors, *Proc. of the 6th International ICCC/IFIP Conference on Electronic Publishing* (ELPUB 2002), Karlovy Vary (Czech Republic).

- C. Caracciolo, W. van Hage, and M. de Rijke (2004). Towards Topic Driven Access to Full Text Documents. In R. Heery and L. Lyon editors, *Proc. of European Conference on Digital Library (ECDL 2004)*. Bath (UK). Springer Verlag.

The results presented in Chapter 6 have been published in:

- C. Caracciolo and M. de Rijke (2006). Generating and Retrieving Text Segments for Focused Access to Scientific Documents. In M. Lalmas, A. MacFarlane, S. Ruger, A. Trombos, T. Tsikrika, A. Yavlinsky, editors, *Proc. of European Conference on Information Retrieval (ECIR 2006)*. London (UK). Springer Verlag.

CHAPTER 2

A Vision on Access to Electronic Scientific Handbooks

Perhaps the most basic thing that can be said about human memory, after a century of intensive research, is that unless detail is placed into a structured pattern, it is rapidly forgotten.
[Bruner, 1960]

The aim of this chapter is to provide a high-level view of the browsable map that we envision and that underlies the work presented in this thesis. We also provide brief introductions to the core areas and terminology that will play leading roles in this thesis: the semantic approach to the organization of information, Information Retrieval (IR) and digital libraries

This chapter is organized as follows. In Section 2.1 we introduce our proposal for providing access to electronic scientific handbooks, i.e., the idea of a browsable map. Then, in Section 2.2 we provide background on scientific handbooks, and in Section 2.3 on searching and accessing books. In Section 2.4 we discuss various types of semantic structures that inspired the LoLaLi map (introduced in Chapter 3): thesauri and taxonomies, semantic networks and ontologies. In Section 2.5 we present related work on providing focused access to scientific documents by modularization of the text. Finally, in Section 2.6 we discuss the issues presented in the course of this chapter, and their connections to our work in this thesis.

2.1 The Vision

In our vision, a person with some knowledge of the domain at the center of our attention (i.e., the interface between logic and language), but not an expert (typically, a graduate or undergraduate student in the area), should be provided with multiple ways of accessing a handbook in electronic format. The bottom line is that the entire text should be accessible for reading, and the text should be queryable in order to locate words and phrases in it. In addition, the user should be guided through the domain covered by the handbook so that more "vague" information needs can be supported as well. The sort of support we envision is provided by a map of the domain, where each topic in the map leads the user to a specific *excerpt* (in the following, also called *segment* or *passage*) of the text in which the topic is covered.

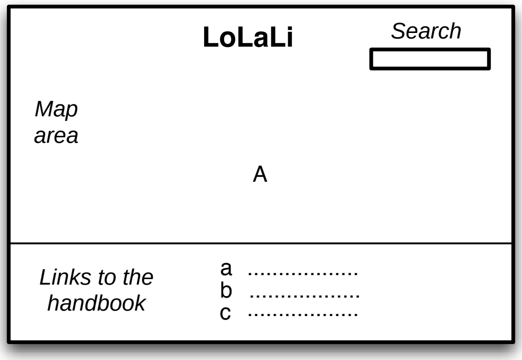
Our idea, then, is to provide the user with an integrated environment, where a browsable map of the domain is provided with two types of links: *internal to the map*, to make explicit (some of) the relations between topics in it, and *external to the map*, connecting the map to the text. The latter type of link connects topics to passages of the book that are internally homogeneous. Such a map is at the same time a tool for users to explore the domain and a tool for retrieving relevant excerpts from the text. In order to ensure maintenance and scalability of the approach, these links should be automatically selected.

What is a good way to create such a map, then? And what is a good way to select links to the text? Although some work has been done on automatically extracting hierarchies from text (see e.g., [Caraballo, 2001, Cederberg and Widdows, 2003, Girju et al., 2003, Roark and Charniak, 1998, Snow et al., 2005]), we assume that a map that is manually created by experts of the domain will be richer, more reliable and of better quality than an automatically created one. The expected downside of this decision is that different authors tend to project their knowledge in different ways, and their opinions about the relations between concepts can vary considerably. Also, the experts who contribute to the map may not coincide with the authors of the text, which, it can be argued, represents an extra source of difficulties.

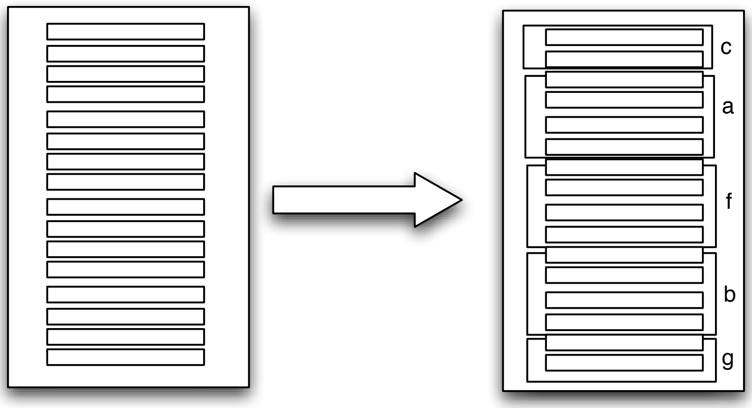
With the help of domain experts, we have organized topics from the domain in a graph[1] where topics are connected by labeled relationships and provided with glosses. The map is organized by means of semantic relationships, both hierarchical and non-hierarchical, that make explicit to the user the relationships between topics in the domain.

The connection between the topics and the text is provided by hyperlinks to link targets (Figure 2.1 (a)), which are found in two steps (Figure 2.1 (b) and (c)). First, the text is divided into *passages*, then the passages are matched to the appropriate topics from the map by means of information retrieval techniques. When dividing the text into passages, we are interested in understanding what kind of passage identification

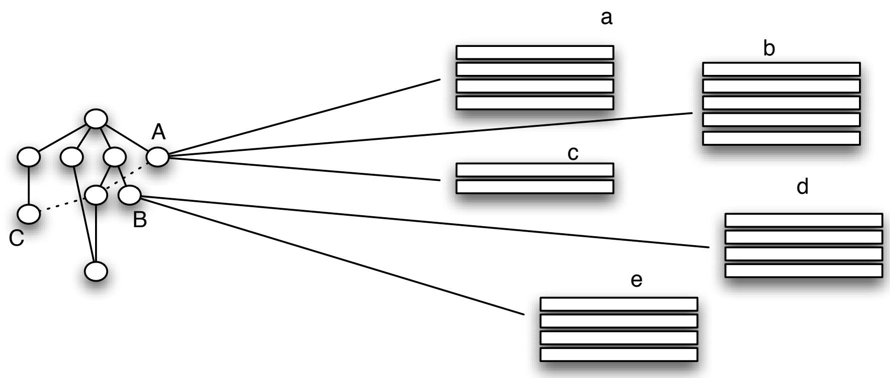
[1] According to standard terminology used to talk about graphs, we will often refer to topics in the LoLaLi map as *nodes*.



(a)



(b)



(c)

Figure 2.1: Graphical representation of the LoLaLi environment and its components. (a) The topics in focus are shown in the center of the map area together with the associated links to the handbook. A search box is always available while navigating. (b) The text is divided into segments. (c) Segments are linked to topics in the map.

technique is most appropriate for the text at hand. But, how can this be evaluated? In order to meet the user's reading needs, we do not only want to evaluate the relevancy of the passage, but also its ability to provide a good *entry point* to the text.[2] This means that the beginning of the text should coincide with the beginning of the relevant text. We tackle these issues by developing and using several evaluation measures. In particular, we interpret the entry point issue in quantitative terms by defining error measures that look at the number of relevant paragraphs missed at the start of the segment (and similarly: the number of irrelevant paragraphs added at the start of the excerpt).

2.2 Handbooks

For most disciplines a time arrives where a need is felt to write a handbook[3] gathering together the knowledge about the domain and passing it to the students. Handbooks can vary in length, organization and style. In the case of comprehensive handbooks dedicated to broad areas, it is not uncommon to have more than one author. An important use of handbooks is as a reference tool, for quick look-up of notions, bibliography checking, often in conjunction with other activities, such as essay writing. These characteristics make the handbook a genre suitable for electronic publication, exploiting the potential of search in a digital environment and facilitating integration with the document writing – and activity more and more directly done on screen. Another reason for exploiting the electronic publication of handbooks and access to them is related to their long publishing cycle, as a result of which new, revised editions become available only after several years, even though the field they cover may develop at a very fast rate. So, an environment for electronic publication of handbooks would be especially valuable, as it would decrease the publication time as well as the time and cost for updates. But in order to have a sound publication environment, we also need effective ways to access such texts, since page browsing is not convenient in electronic documents.

For documents on paper, including handbooks, a number of mechanisms are available for accessing the information contained in them: tables of content, indexes of names, topics, figures, tables and so on. The table of contents refers to the internal organization of the handbook, while an index helps locate relevant topic (figures, tables, names, ...) in the text. Both tables of contents and indexes use pages to locate information and, in the case of indexes, only mention the *important* pages where a *term* appears: they do not mention all terms, nor all occurrences of them, nor any finer mechanism than the page to indicate *where* the term is discussed in the document. We envision a more comprehensive way to retrieve topics in a electronic handbook, one that extends the functionalities of the back-of-the-book index.

[2]The notion of entry point has received a lot of attention in the setting of structured document retrieval; see [Reid et al., 2006].

[3]There is no clear distinction between handbooks and manuals, but manuals are often oriented toward procedural knowledge, while handbooks are usually dedicated to more theoretical issues.

2.3 Searching and Accessing Textual Documents

Traditionally, e.g. in a library or archive, accessing a book has meant retrieving a catalog card that contains a pointer to the "real" document. The catalog card is part of some inventory or catalog, based on *classification*, i.e., a grouping of similar or related things and arrangement of the resulting groups in a helpful sequence. Alternatively, the catalog may be an *index*, i.e., a list of words of a book so that the book they refer to can be easily found. Classifications and indexes can be organized into hierarchies such as classifications systems and thesauri (we look into these in Section 2.4).

The first retrieval experiments were performed on machine readable catalog cards and abstracts in the 1950s: there we can place the birth of computerized Information Retrieval (IR). By storing the catalog in a database system, it is possible to speed up the process of searching for the catalog card, both by searching through the description metadata and the content metadata (manually assigned indexing terms).

With today's availability of full length documents, as opposed to abstracts, keywords, or index terms, comes a more flexible indexing methodology where virtually any word can be an index term, and where virtually any term can in principle be used in query formulation. The possibility of automatically indexing the entire document allows one to retrieve a document not only on the basis of the metadata available, but also on its *content*. Another consequence of the availability of full text documents and indexes is that it is possible to access the content *below* the document level—an area in (textual) IR that has received a lot of attention in recent years (through tasks such as passage retrieval [Salton et al., 1993a], XML retrieval [INEX, 2005], question answering [Maybury, 2004], and entity retrieval [Sayyadian et al., 2004]) and one to which we will return frequently in the thesis.

The main components of an information retrieval process are: the user, an information need expressed as a query, a retrieval model and a document collection. Let us start from the end. A document in a collection can be represented in many ways, by means of its title, or editor (author), or a combination of these or other pieces of information such as abstracts and keywords. In modern information retrieval it is common to represent a document by means of an automatically generated *inverted index*, i.e., a list of the words occurring in the document associated with their position in it. This data structure facilitates fast search of index terms and does not presuppose any predefined classification system nor any knowledge of classification or criteria for manual indexing on the side of the user. The same type of representation is applied both to the document and the query.

Many IR models have been studied and implemented over the last five decades, including logical models, vector space models, probabilistic models and models based on language modeling; see, e.g., [Baeza-Yates and Ribeiro-Neto, 1999, Grossman and Frieder, 2003]. In this thesis we adopt the classic vector space model, where documents and queries are represented as vectors of words, where words can be weighted according to some criteria, such as their frequency, or a combination of term frequency in the document and its frequency in the collection. In a vector space model, the cosine of the angle formed by the document vector and the query vector represents the degree of similarity between them. Therefore, documents in the collection are ranked according to the value of the cosine of the angle they form with the query. Since documents may be lengthy and may cover more than one topic at different levels of details, there have been attempts to look inside a document and use *evidence* from parts of it to better assess whether the document is relevant to a query; see [Wilkinson, 1994] for an early example. In the course of this thesis we look at various ways to split a document according to the topics it covers and how to link the resulting excerpts to a map of the domain.

Very early on, the IR community addressed the issue of how to evaluate a system and how to compare systems to one another [Cleverdon, 1970]: over the years a number of measures have been proposed, each capturing different aspects of the retrieval process. The most popular quantitative measures are certainly *precision* and *recall*, where precision looks at the proportion of retrieved documents that are relevant, and recall gives the proportion of relevant documents that are retrieved. Other measures have been designed that take into account the ranking proposed by the system, others are more explicitly user oriented ([Cooper, 1968, de Vries et al., 2004]), and still others have been designed to evaluate specific retrieval tasks. The user plays a central role in a special branch of IR research called Interactive IR [Ingwersen, 1992, Robins, 2000], where the focus is on the interaction of the user with the IR engine: how she expresses information needs into queries, how she evaluates search results, and how she issues new queries. In the view put forward in this thesis, the LoLaLi map supports information seeking activities such as searching and browsing, and the issue of providing focused access inside a handbook is treated as a special retrieval task—for this, we follow the methodologies developed in the IR community. We also propose our own measures in order to quantify aspects that are especially relevant for us and for the reader of the retrieved excerpts; see Chapter 5.

2.4 Semantic Structures

Full text indexing is one way of describing the contents of a document. Another way to describe a document (and so to find it among many others) is to use words reflecting its content. Such words are called *keywords*, that could in principle be rare or absent from the document. In order to minimize the chances of mismatch between the keywords the user would assign to the document and the keyword actually used by the (human) indexer, it is common practice to define a set of admitted keywords, usually called a *controlled vocabulary*. For example, indexers can agree, or be compelled, to consistently use the keyword 'computer science' even when they would have preferred 'informatics.' Controlled vocabularies ensure uniformity of indexing, which should imply high accuracy at retrieval time.

The advantage of keyword indexing by controlled vocabulary is that, assuming that the user is able to find the appropriate keyword to describe her information need, she is automatically able to find all documents described by that keyword. The disadvantages are that it forces the user to "guess" the appropriate index term, and that the index terms are taken as independent, with no explicit relationship among them. The latter problem is (at least partially) overcome by structures such as taxonomies and thesauri, which enhance a controlled vocabulary with relations.

Taxonomies and Classifications

Taxonomies refer to the classification of objects (or any kind of entity) into categories[4] and in principle the classification could be based on any kind of law or principle. In practice the most common taxonomic relation is: 'A is a kind of B,' whose corresponding relation of exclusion is 'C is a different kind of B' (cf. Section 3.4.1).

A taxonomy is a very versatile structure, used to represent human knowledge (e.g., the tree of knowledge of Raymond Lull, Figure 2.2), regularities found in nature (e.g., the taxonomy of species by Carl Linnaeus (1707–1778)), and subjects for the purpose of book classification (e.g., the Dewey Decimal Classification system (1876)).[5] A taxonomy can also use a parthood relation, in that case it is called a *meronymy*.

Other classification systems, such as the ACM Classification System[6] [ACM, 1964], are specific to a given domain. The ACM classification system (Figure 2.3) is used to index scientific papers in the area of computer science. It is based on a hierarchy (a tree) of terms organized in four levels, of which the 11 nodes at the first level are never used for classification, while the nodes of the levels below are actually used for indexing the articles published by ACM in order to enhance search for papers in the collection of their publications.

Thesauri

In literature, thesauri are dictionaries in which each entry is listed together with its synonyms (e.g., *Roget's International Thesaurus of English Words and Phrases* [Roget and Davidson, 2002]), but the usage of the term thesaurus here, now widespread, dates from the early 1950s in the work of Luhn [1958], who sought ways to automatically create a list of authorized terms for indexing scientific literature. The list was to include a structure of cross-references between families of notions. Usually, though, specialized thesauri are organized by relations forming a hierarchical structure (a tree) among the terms in it. The typical pair of hierarchical relationships in a thesaurus is *broader term* (BT) and *narrower term* (NT), while non-hierarchical relations include

[4] As the ancient Greek etymology suggests: 'class' + 'rule.'

[5] Although there is no agreement on the historical origins of the DDC, Dewey himself openly refers to the Baconian classification [Wiegand, 1998]. According to Bacon, human knowledge depends on three main faculties: Memory (History), Imagination (Poetry/Art) and Reason (Philosophy). Dewey adopts an inverted Baconian system of classification, where Reason takes the classificatory numbers from 1 to 6, Imagination 7 and 8, Memory 9. General works take the number 0.

[6] The first version dates back to 1964, the second version, totally different from the first one, was issued in 1982, then other versions based on that followed. The most recent is from 1998.

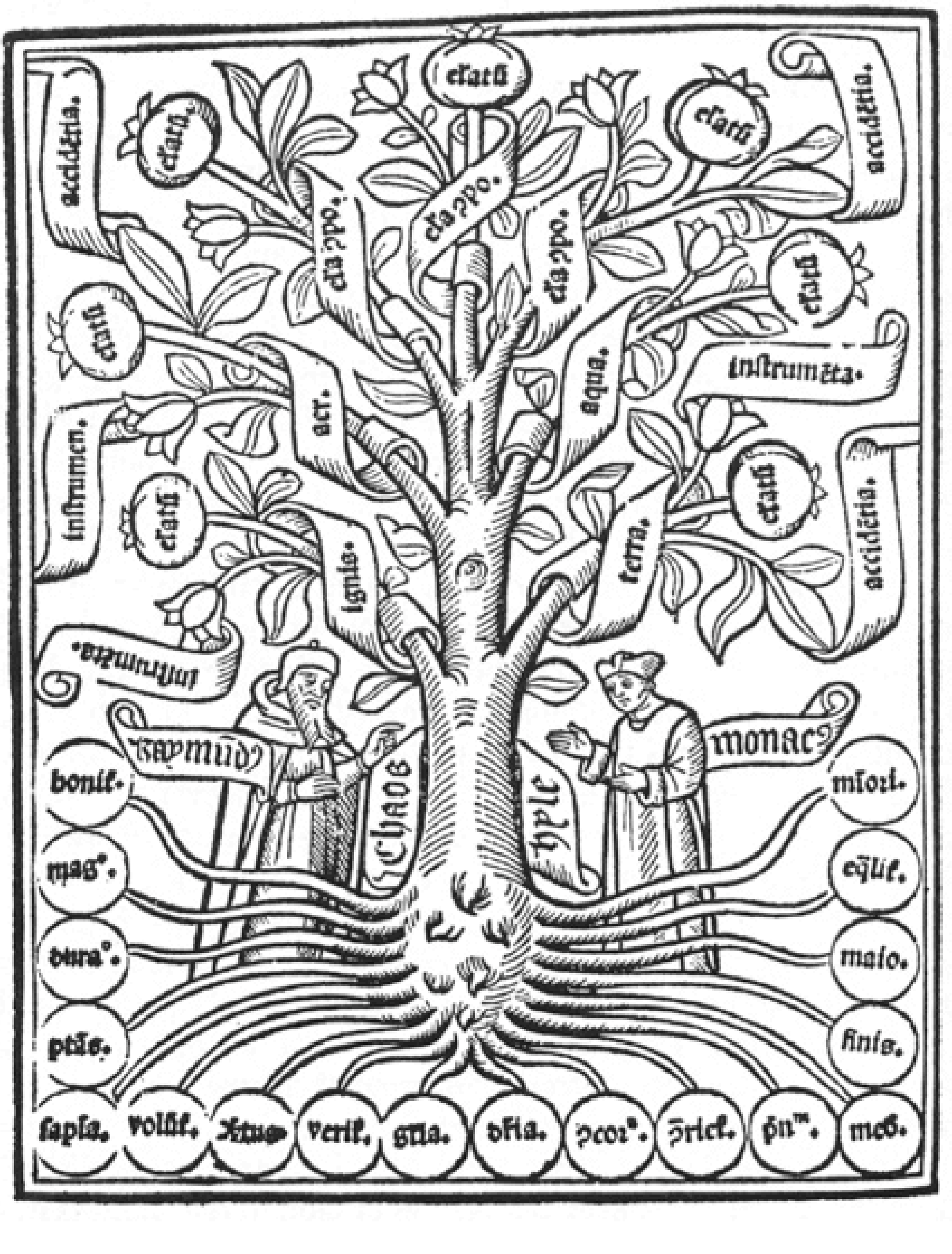


Figure 2.2: A medieval representation of the Arbor Scientiae (tree of knowledge) by Ramon Lull.

used for (UF, which marks a *preferred term* or *descriptor* to be used for indexing instead of *non-preferred terms* or *non-descriptors*; the inverse relation is *use* (USE)) and

The ACM Computing Classification System (1998)

- A. General Literature
  - A.0 GENERAL
    - *Biographies/autobiographies*
    - *Conference proceedings*
    - *General literary works (e.g., fiction, plays)*
  - A.1 INTRODUCTORY AND SURVEY
  - A.2 REFERENCE (e.g., dictionaries, encyclopedias, glossaries)
  - A.m MISCELLANEOUS
- B. Hardware
  - B.0 GENERAL
  - B.1 CONTROL STRUCTURES AND MICROPROGRAMMING (D.3.2)
    - B.1.0 General
    - B.1.1 Control Design Styles
      - *Hardwired control* [**]
      - *Microprogrammed logic arrays* [**]
      - *Writable control store* [**]
    - B.1.2 Control Structure Performance Analysis and Design Aids
      - *Automatic synthesis* [**]
      - *Formal models* [**]
      - *Simulation* [**]
    - B.1.3 Control Structure Reliability, Testing, and Fault-Tolerance [**] (B.8)
      - *Diagnostics* [**]
      - *Error-checking* [**]
      - *Redundant design* [**]
      - *Test generation* [**]
    - B.1.4 Microprogram Design Aids (D.2.2, D.2.4, D.3.2, D.3.4)
      - *Firmware engineering* [**]
      - *Languages and compilers*
      - *Machine-independent microcode generation* [**]
      - *Optimization*
      - *Verification* [**]
    - B.1.5 Microcode Applications
      - *Direct data manipulation* [**]
      - *Firmware support of operating systems/instruction sets* [**]
      - *Instruction set interpretation*
      - *Peripheral control* [**]
      - *Special-purpose* [**]
    - B.1.m Miscellaneous
  - B.2 ARITHMETIC AND LOGIC STRUCTURES
    - B.2.0 General

Figure 2.3: A fragment of the ACM classification schema (1998).

related term (RT) [ISO:2788, 1986]. Sometimes the relation *equivalent term* is also used, to express a relation of synonymy. For the sake of uniformity, only descriptors are used for indexing. Monolingual thesauri are usually a strict tree (each term has only one broader term), but multilingual thesauri often allow multiple parenthood.

Thesauri are widely used for indexing and cataloging in library and information sciences, especially when dealing with restricted domains where a high degree of detail is required. A wealth of organizations have developed their own thesauri, in areas as diverse as engineering [Aitchison, 1970] and art [Peterson, 1994], up to the point that from the 1960s throughout the 1980s thesauri became *the* tool for document indexing and occupied a leading role in the area usually called knowledge management. In fact, both standards for the creation of thesauri, i.e. for monolingual and multilingual, maintained by the International Organization for Standardization (ISO) date back to the 1980s [ISO:2788, 1986, ISO:MLT, 1985]. At the same time, the organization also approved a standard for indexing [ISO:IND, 1985].

Thesauri have also been used in IR since the very beginning of the field for the so-called "thesaurus approach" to information retrieval [Joyce and Needham, 1958, Salton, 1968]. The idea was to combine classification with indexing by grouping words together in "notional families." These families of grouped keywords are considered to define a classification. By using a thesaurus provided with relations of BT/NT one could also make an inclusive search, meaning the possibility of automatically propagating the assignment of keywords by means of the hierarchical structure of the thesaurus. The expected advantages are "easier" indexing for the indexer (because there are more indexing terms to choose from) and "easier" retrieval for the user (since indexing terms are grouped together and organized in a structure and there are more chances for the retrieval to be exhaustive). In IR, thesauri have also been used for the automatic enhancement of queries (an application usually called query expansion), a method that has proven to be useful when queries are short (2–3 words) [Voorhees, 1994].

Semantic Networks and WordNet

Recently, thesauri and thesauri-like structures have enjoyed a renewed interest in the area of the Semantic Web [Berners-Lee et al., 2001]. Before touching on the Semantic Web below, we introduce Semantic Networks and WordNet, historical antecedents of the semantic web.

A *semantic network* is a graphic notation for representing knowledge in patterns of interconnected nodes and arcs [Sowa, 1991]. In practice, it is a directed graph consisting of vertices representing concepts and edges representing relations between concepts. Semantic networks can be more complex than thesauri in that they often admit multiple parenthood, and can contain virtually any relationships (e.g., 'instrument for,' 'mother of,' 'affected by,' ...), but usually concepts are organized according to levels of generality based on taxonomic distinctions. Also, properties and relations of the nodes in the network are usually expressed in some formal language.[7]

[7] The origin of semantic networks can be traced back to Charles C. Peirce (1839–1914), who devel-

Semantic networks have been intensely studied and developed between the 1960s and 1990s within the framework of computer science, artificial intelligence, knowledge representation and logic, where emphasis was put on the *reasoning* capability of the system. The most common type of reasoning implemented was the inheritance mechanism (relations that hold for all concepts of a given type are inherited through the hierarchy).

WordNet [Miller, 1995], usually defined as a "a lexical database of English," is perhaps the best-known example of a semantic network.[8] WordNet distinguishes nouns, adjectives, verbs and adverbs and uses different hierarchical relations for each group. The basic non-hierarchical relation is the lexical relation *synonymy*: each node in the graph consists of a set of terms, a *synset*, that are synonyms in a certain context. Any pair of synsets in the graph is connected by one of the following relations: hypernymy/hyponymy, coordinate terms (i.e., siblings under the same hypernym), holonymy/meronymy (i.e., has part/part-of). Each synset is also endowed with a gloss to explain the exact meaning of the terms in it, added because the presence of synonyms in the same synset does not always allow one to disambiguate the term.

The distinctive feature of WordNet is that it does not use preferred terms (as opposed to classical thesauri where preferred terms ensure uniformity of indexing), because it groups together partial synonyms (words that are interchangeable in some context) in synsets. A synset, as a whole, provides a notion of *meaning* of a word. This structure makes WordNet a *sort of* dictionary (as glosses aim at giving the *sense* of a synset, not the definition of a single term), and also a sort of ontology (see below).

Beside its importance as a psycholinguistic experiment, WordNet has had great success in computational linguistics and in information retrieval applications. WordNet is also widely used for enriching ontologies and thesauri for indexing. In computational linguistics applications WordNet is especially used for word sense disambiguation, in IR as a tool for query expansion [Voorhees, 1994]. Other work has concentrated on the study of conceptual distances among words for purposes such as opinion detection in political speeches [Kamps and Marx, 2002]. Currently, dozens of WordNet projects are being carried out around the world, including MultiWordNet [MULTIWN, 2005], Indi and Marathi WordNet [IWN, 2005], EuroWordNet [EUWN, 2005], and Cornetto [Vossen et al., 2007].

As we will see in Chapter 3, the LoLaLi map has adopted some of WordNet's features, such as a gloss attached to each topic and a synset-like way of grouping of terms.

---

oped en *existential graph* as a graphical system of logic.

[8] WordNet was created in 1985, and is still actively maintained at the Cognitive Science Laboratory of Princeton University under the direction of psychologist George A. Miller, to test the hypothesis that the human memory groups words together according to their function in the language, or linguistic category. Miller's interest in human memory traces back to the 1950s, when he wrote his seminal paper: "The Magical Number Seven, Plus or Minus Two" [Miller, 1956].

Ontologies and the Semantic Web

The latest incarnation of semantic structures, usually called *ontologies*, has become a crucial ingredient of the Semantic Web [Berners-Lee et al., 2001, van Harmelen and Antoniou, 2004, W3C, 2006]. The idea behind the Semantic Web is that by associating metadata with objects on the Web they can be better retrieved, it is possible to achieve interoperability and exchange of data, and to perform automatic inference (as opposed to inference commonly performed by human agents reading text in natural language). In such a scenario, ontologies are crucial because they can be used to formalize the semantics of the metadata used to annotate objects in the Web.

In practice, there is a loose usage of the term ontology. It can indicate different structures, with big variations in specificity and complexity, such as controlled vocabularies, glossaries, thesauri, term hierarchies, strict subclass hierarchies, frames and value restrictions (see [McGuinness, 2002] for a short survey of them). However, a popular definition is given by Gruber [1993, page 5], according to whom an ontology is "a specification of a conceptualization." This definition corresponds to the view adopted in computer science, where an ontology consists of a set of concepts connected by relations describing a domain of interest [Vickery, 1997]. Nowadays, though, the tendency is to consider ontologies to be structures that are at least as complex as a taxonomy[9] and encoded in a formal language such as RDFS [RDFS, 2004] or OWL [OWL, 2004] that allows some kind of inference.

The key role of ontologies for the Semantic Web has fostered a wealth of research: on standardization of formal languages to encode ontologies, on tools to edit, manage and visualize them, on methods to reuse ontologies and share them, on software interoperability through metadata, and more. Although a deeper discussion about the Semantic Web is beyond the scope of this thesis, we looked at the area for technologies and tools that could help us in our effort to build and maintain the LoLaLi map; see Chapter 3 for details, where we also discuss the extent to which the level of formality often required by semantic web-based solutions is appropriate for our domain and tasks.

2.5 A Modular Approach to Focused Access

The work of Harmsze [2000] (see also [Kircz and Harmsze, 2000] and [Kircz, 1998]) relates to the model we hinted at in Section 2.1, in that she proposes a modular model for the electronic publishing of scientific articles. Her idea is that in an electronic environment, scientific information may be communicated more efficiently and effectively if it is presented as a network of articles with a modular structure rather than as a set of linear, essay-type articles. The model would provide focused access to information and enable re-use, better retrieval and clarity.

[9] In fact, the word ontology is often used as a synonym for taxonomy, although an ontology can use a broader set of relations than the subclass relation.

Harmsze's work is grounded on the analysis of the argumentative structure made by Grootendorst et al. [1996], and on the basis of that she provides a multi-dimensional classification of modules typical of certain types of scientific articles (e.g., hypothesis, data, experiments), and of their aggregation. She also provides a multi-dimensional classification of links: both of those that aggregate simple modules together, and of those that connect aggregated modules to one another. An article with a modular structure then consists of a coherent collection of explicitly linked modules, representing a coherent network of related conceptual information units within the larger network of published information. This structure allows the user to take into account the role that concepts of interest should play within a document, as well as the relations between specified concepts.

Harmsze defines requirements for relations and modules, and applies the model to the modularization of two articles. Such conversion turns out to be possible, although very difficult, because of the difficulty of extracting modules from originally paper-based sections. In particular she found that: (1) some pieces of information were inherently difficult to characterize unambiguously, which led to overlap of different modules; (2) even when the characterization of different strands was unambiguous, some information of different types was so closely interwoven that it was difficult to disentangle it; (3) in order to obtain a complete module, it was necessary in some cases to add extra information not provided in the original version. Her study concluded that scientific articles written for different media need a different structure. Therefore, she developed extended guidelines for writing modular articles from scratch.

Contrary to Harmsze, who started from the documents and concentrated on modules within the texts (articles on experimental sciences) or among them (e.g., a set of articles resulting from the same project), we start from the map, focus on the modeling of a map of the domain and then address the issue of providing an automatic way to link excerpts from the document to the map. In this sense, our approach is more general than the one described in [Harmsze, 2000], but some overlap is evident, such as in the types of relationships found. For example, the relation `mathematical result` (in the LoLaLi map, cf. Chapter 3) is reminiscent of the module 'results' in Harmsze's approach. Despite these similarities, though, Harmsze focuses more on argumentative or organizational relations than on domain relations. For example, whereas Harmsze considers relations (among others) such as 'Elucidation relations,' 'Clarification relations,' and 'Explanation relations,' we concentrate on relations such as 'part of,' and 'mathematical results.'

2.6 Discussion

The vision presented in Section 2.1 aims at enabling focused access to information, while supporting information seeking activities that include searching, browsing and exploration of a domain map with explicit relations between topics. This approach should support different information needs, ranging from vague to precise. In the course of this chapter we have presented our vision on accessing electronic handbooks, and we connected it to the areas in which our approach is grounded.

We acknowledge the importance of semantic structures to provide an overview of a subject area, and add to it the possibility of searching at various levels: the map level, the *Handbook* and the links connecting them. In the course of this thesis we concentrate on the making of the map, on the selection of links to connect to it; more details about the search on the map can be found in [van Hage, 2004].

Research in Information Retrieval started with search on catalog cards and ended up generalizing the concept of indexing to the point that inverted indexes are now the data structure of choice in the area. Full indexing allows one to search on virtually any word in a document, but it relies on the ability of the user to issue the "right" query, and may force her to issue different queries to capture documents written according to different styles and vocabularies. Structures like thesauri work on the idea that optimal search and retrieval can be achieved by looking at the content of the document, or its *meaning*. We called these structures *semantic* so as to emphasize their ambition to represent the meaning of words and the meaning of documents. We stressed the fact that classification and grouping is very intuitive and was used very early on to represent documents and subject areas. In the context of electronic search and reading environments, though, systems solely based on this approach suffer several major limitations, including the ability of the indexer to capture *all* meanings of a document, and the need for the user to have some knowledge of the classification system adopted in order to "guess" the right classification or keywords.

Against this background, the LoLaLi map we present in the next chapter was inspired by the structures we presented in this chapter (especially WordNet) but with the main purposes of providing a map of the domain and a bridge to the document. As will become clear from Chapter 3, we took from WordNet the use of synsets and some relations. From the Semantic Web area, we took some of the tools we used to implement the LoLaLi map.

CHAPTER 3

A Browsable Map for Logic and Language

In this chapter we describe the Logic and Language Links (LoLaLi) map: a structure that is inspired by the semantic approaches described in Chapter 2 (i.e., taxonomies, thesauri, semantic networks and ontologies) and whose purpose is to provide a browsable and searchable resource, to make explicit to end users, and especially non-expert end users, the internal organization of the domain. Importantly, it is also meant to provide a bridge to the *Handbook of Logic and Language* by providing links to relevant excerpts from the handbook. Thus, the LoLaLi map, as a searchable and browsable resource, is meant to support information seeking for a variety of information needs.

The chapter is organized as follows. In Section 3.1 we present the requirements we impose on a browsable map for the domain of logic and language. In Section 3.2 we describe the approach we followed to design and populate the map. In Section 3.3 we describe the pieces of information attached to the topics in it, and in Sections 3.4 the relations used. In Section 3.5 we highlight the features of the map. Section 3.6 is dedicated to a discussion of the modeling choices made, and in Section 3.7 we describe the practical aspects of editing and managing the LoLaLi map, including the editorial environment we used and the way the map is accessed through a web browser. Finally, in Section 3.8 we present the conclusions we draw from this work.

3.1 User Requirements

The intended end users of the LoLaLi map are assumed to have some, but not extensive, knowledge of the domain. They may have some technical notions but may not be aware of in-depth details of the area. This choice has implications on the organization of the map and on the selection of its content. Also, end users are not supposed to have any background in knowledge modeling, which implies that the information to be shown to them and the way it is shown should be carefully planned. Given this picture of our intended user, the map should satisfy the following requirements:

Requirement A. Include relevant topics from the chosen domain.

Requirement B. Be informative for the audience addressed.

Requirement C. Avoid information overload.

None of these requirements can be further pinned down in a formal way because they all depend on many vague factors. All requirements refer to a common notion of end user, and Requirement A also refers to a notion of *coverage* that depends on the end user, on the domain at hand, and on the text to which we want to connect the map. Requirement C refers to a phenomenon that, although difficult to define sharply, certainly results from excessive demands made on the cognitive processes, in particular memory; it also depends on the background of the envisaged end user. Between requirements B and C there is a natural tension, because the more pieces of information are made explicit, the greater the information and cognitive load for the user. We now discuss requirements A, B and C in a bit more detail.

Requirement A. Include relevant topics from the chosen domain. A *topic* included in the LoLaLi map is any topic, notion or idea pertinent to the domain at hand that the domain experts developing the map consider *relevant* for the end user. The relevance of a topics is then decided by the authors of the map. We do not impose that *all* relevant topics be included in the map, as *total coverage* (with respect to the domain, and with respect to the *Handbook of Logic and Language*) is a goal that will only be achieved with difficulty, and over a long time, assuming that it is possible to achieve at all [Cimino, 1998]. We rather aim at incremental coverage to be achieved over time.

Requirement B. Be informative for the audience addressed. By *informativity*, we refer to the *type* of pieces of information that the map should include in order to satisfy the information needs of our envisaged end users. In this respect, we consider a map to be informative if it provides a representation of topics included and of the relations between topics, that is appropriate to the understanding of the users.

In order to be informative to our intended users, topics should be provided with some essential textual information about them, such as short definitions, together with a selection of fundamental relations to other topics in the map. These relations should be more detailed than simple 'see also' or 'broader than' relations, but they should not overload the user (see the following requirement) by applying too fine-grained distinctions that can only be grasped on the basis of a deeper knowledge of the area. Semantic relationships have always played an important role both in the area of databases [Storey, 1993] and in the area of knowledge organization [Clarke, 2001], and considerable effort has been put into the classification of different types of relationships. The question for us is then: what are the right relationships (in type and number) for our purposes? We answer the question by considering the relations as dependent not solely on the domain, but also on the end users we address, both in terms of their background knowledge and in terms of the use we expect them to make of the map. We tried to meet these requirements when gathering the input provided by a group of domain experts (cf. Section 3.2).

Requirement C. Avoid information overload. The expression *information overload* is vague, as it may refer to a variety of circumstances in a variety of domains observed under a variety of possible perspectives. Information overload can be a condition resulting from receiving information that overwhelms one's short term memory capacity, or it can result from information that exceeds one's ability to benefit from it [Eisenberg and Small, 1993]. It can also be defined as the overwhelming feeling deriving from having too much information to deal with [Stanley and Clipsham, 1997]. Others have defined it as having more information than one can take in, or having information that goes unused because one simply lacks the time or motivation to process and understand it [Wilson, 1995]. Other views on information overload include those given in [Biggs, 1989, IEEE, 1995].

As far as we are concerned, information overload is related to both the "amount" of information provided and the "ability" to process it (themselves related to one another). In particular, this can result from a map that is too complex (in terms of relations among the topics), from a carelessly designed graphical user interface (GUI), and from confusing interaction modeling. These considerations played a role in the process of designing and populating the map (Section 3.2), and in designing the user interface for it (Sections 4.1 and 4.3). For example, the resulting set of relationships is relatively small compared to the number of relations that an experienced researcher might assign to topics in the domain. We are also aware of what a delicate issue the GUI is: as in any digital environment, it is all too easy to overload the user with information and visual hints, and the result is that the user is distracted and can have difficulties processing the information. One common form of information overload comes from being presented with too many items of information on a single screen, and it easily worsens when dynamic features are added to the interface. Other sources of overload are abrupt "jumps," forced by hypertextual organization of content, and by interfaces where the layout changes when moving from one screen to another. These types of information overload can be avoided by careful interface design, others have to be taken into account already when designing the map.

3.2 Design and Content of the Map

From an initial survey we conducted, it emerged that no reusable resources were available for our purposes. The ACM classification system [ACM, 1964] (see Section 2.4 and Figure 2.3), the closest to our needs, is too high-level and partially tangential to our domain and it only includes a BT/NT structure. Also, the terms (about 100) coming from work previously carried out within the LoLaLi project [Ragetli, 2001] were encoded as a set of Prolog facts and organized by means of BT/NT relations (see also Section 3.7). Therefore, we had to reorganize and extend the data we had.

Although the focus of our work is not on knowledge representation, the building of the LoLaLi map can be viewed as a knowledge engineering effort, where knowledge has to be elicited by domain experts, often with the constraints imposed by a set of user requirements. The issue of designing and populating structures representing domain knowledge has been widely studied, both in the area of knowledge engineering [Schreiber et al., 2000] and ontology design [Noy and McGuinness, 2001]. Given these similarities, we borrowed several techniques and insights from that community.

As mentioned earlier (Section 1.1), we wanted domain experts to provide both the design (i.e., which relations to use) and the population of the map (i.e., which topics to include in it). The implication of this is that we had to mediate between the different perspectives and approaches of the different domain experts, and between their input and the requirements we illustrated in the previous section. In Sections 3.3 and 3.4 we report on our current view of the map. This view results from an iterative process that included our analysis of the domain, preliminary interviews with subject experts (not knowledge representation experts), regular consultations with the User Centered Design group at Elsevier (the publisher of the *Handbook of Logic and Language*), and the user studies we report on in Chapter 4.

The content of the LoLaLi map comes from what was inherited from a previous phase of the LoLaLi project (re-organized according to the chosen set of relations), and from a substantial number of new contributions provided by domain experts in the course of the second phase of the project. Domain experts were asked to draw simple hierarchical representations of fragments of their choice, following the "laddering technique" [Corbridge et al., 1994] often used for knowledge acquisition. Domain experts were asked to only use the pre-defined set of relations, but in some cases they freely gave their own schematization of their area of expertise, pointing at relations that we did not include but they found important to represent (one of these cases is in fact the source of the discussion presented in Section 3.6). Finally, we provided some of the content ourselves, using the back-of-the-book index of the original paper version of the handbook as a basis. Our intervention was often required to provide a junction between fragments contributed by domain experts, or to ensure the desired level of correspondence between the map and the *Handbook of Logic and Language*. This very distributed way of populating the map shed interesting light on the issue of organizing a workflow involving different roles such as authors and editors. In the concluding chapter we reflect on this issue.

The development of the LoLaLi map was not set up as a formal knowledge representation or knowledge engineering effort. The LoLaLi map does not come with an associated formal semantics. We arrived at a "loose" representation of the domain at hand based on both principled and pragmatic considerations. First and foremost, the information elicited from our subject experts is not of a prescriptive nature but of a descriptive aimed one capturing how topics are actually used in the domain. Second, while the formality of knowledge representation languages may be appropriate for, say, exchanging data between knowledge bases, it does not fit well with the vagueness and ambiguity inherent in people's language usage, and may even be counter-productive—where knowledge transfer involves humans, there is no vehicle like natural language [Spärck Jones, 2004]. In other words, a formal presentation of the LoLaLi map is at odds with requirements B (informativity) and C (avoid overload). Moreover, given the data-driven and bottom-up approach used for retrieval in Chapters 5 and 6, a formalization seems superfluous; the methods we test and evaluate in those chapters are independent of the structure of the map and only take into account the topics in it.

Before going into the details of modeling issues, a few remarks on the typographical conventions used. We write in sans serif the topics included in the map and in `type writer` the name of relations in the map. Examples and topics and relations that are mentioned in a general way (not because they belong to the LoLaLi map) are mentioned in 'single quote.' All graphical representations of (fragments of) the map presented in the rest of this chapter were realized by using Graphviz [GRAPHVIZ, 2007]; the arrows that connect topics to one another point upward (from the specific to the general), so that 'A $\longrightarrow$ B' reads as 'A is some sort of B.'

3.3 Topics

A topic in the LoLaLi map is anything that is judged relevant by the authors of the map and suitable for inclusion in it (cf. Requirement A). More generally, anything about which something can be asserted can be included in the map. Topics in the LoLaLi map are represented by means of a *title* and an information definition, or *gloss*.[1] A title consists of one or more (English) terms used to express that topic. Terms in a title are related to one another by a *synonymy* relation, i.e., the lexical relation between terms that can be substituted for one another in certain, but not necessarily all, contexts.

As pointed out in Section 2.4, the relation of synonymy represents a core relationship in WordNet [Fellbaum, 1998] since the meaning of a concept is represented by means of a *synset*: all terms that can be used interchangeably in given situations (partial synonymy). The underlying idea of using synsets in WordNet is that a term can be ambiguous when taken out of context, but is disambiguated by all its synonyms in a certain context. For example the term 'wood' can signify the hard fibrous substance under the bark of trees, or the trees and other plants in a large densely wooded area,

[1] Originally, a gloss is a note made in the margin of a book, to explain the meaning of the text.

or any wind instrument other than the brass instruments. The second and third case can be distinguished by the first one by using the synonyms ‘forest’ and ‘woodwind instrument.’

In monolingual thesauri [Clarke, 2001], the relation between synonyms or quasi-synonyms (i.e., partial synonyms) is often indicated as an equivalence relation. However, in thesauri the terms used in a synonymy relation are often not true synonyms, but rather just terms whose meanings are sufficiently close, in certain contexts, for the purpose of the database to be indexed and searched. For example, the terms ‘porcelain,’ ‘bone china’ and ‘crockery’ are not real synonyms, but they could serve as such in a non-specific thesaurus for general use. Similarly, a general, non-specific thesaurus could contain ‘linguistics,’ ‘historical linguistics’ and ‘formal linguistics’ as synonyms.

Also in the domain of logic and linguistics, there are terms that can be used interchangeably. For instance, first order logic, FOL, and predicate calculus are synonyms, and together they form a title, the same way context free grammar and CFG do. Contrary to thesauri, and similarly to WordNet, we have no notion of preferred terms.

Glosses are short pieces of text, usually of about 2–3 sentences, added to each topic in order to give the user quick insight into the topic. Also, when two topics are known in the literature by the same name (with no synonyms available to distinguish one from the other) but have different meanings, their glosses will point out the differences between them. A number added next to the title also differentiates them, as is normally done in dictionaries. For example, the LoLaLi map contains a topic logic (1) under computer science, mathematics, artificial intelligence and linguistics, and a topic logic (2) under philosophy. The gloss of logic (1) is “A system of calculus or reasoning,” the gloss of logic (2) is “The branch of philosophy that analyzes inferences” (Figure 3.1). Glosses also provide the de facto definition accepted by the author for a topic in the map.

3.4 Relations

The thesaurus relations broader term/narrower term (BT/NT) are very flexible in that they can be used to link any pairs of concepts such that one is “broader” than the other. Such a flexibility is to the detriment of expressivity, as pairs of concepts whose relations are intuitively very different are linked by the same BT/NT relation. Consider for example the following two pairs taken from [FAO, 2007]: Fishing Line is NT of Fishing Gear, and Adriatic Sea is NT of Mediterranean Sea. One of the purposes of the semantic relations of subclass and part-of is to account for these differences [Clarke, 2001]. It is also often customary to distinguish between subclasses and instances, where the latter indicate individuals belonging to a class (i.e., when the class is taken as a set).

These distinctions also apply to the domain of logic and linguistics, where we identified the need to be able to express the following: that a given topic *is a* type or specification of another topic (e.g., first order logic *is a* logic), that a topic *is part of* a broader

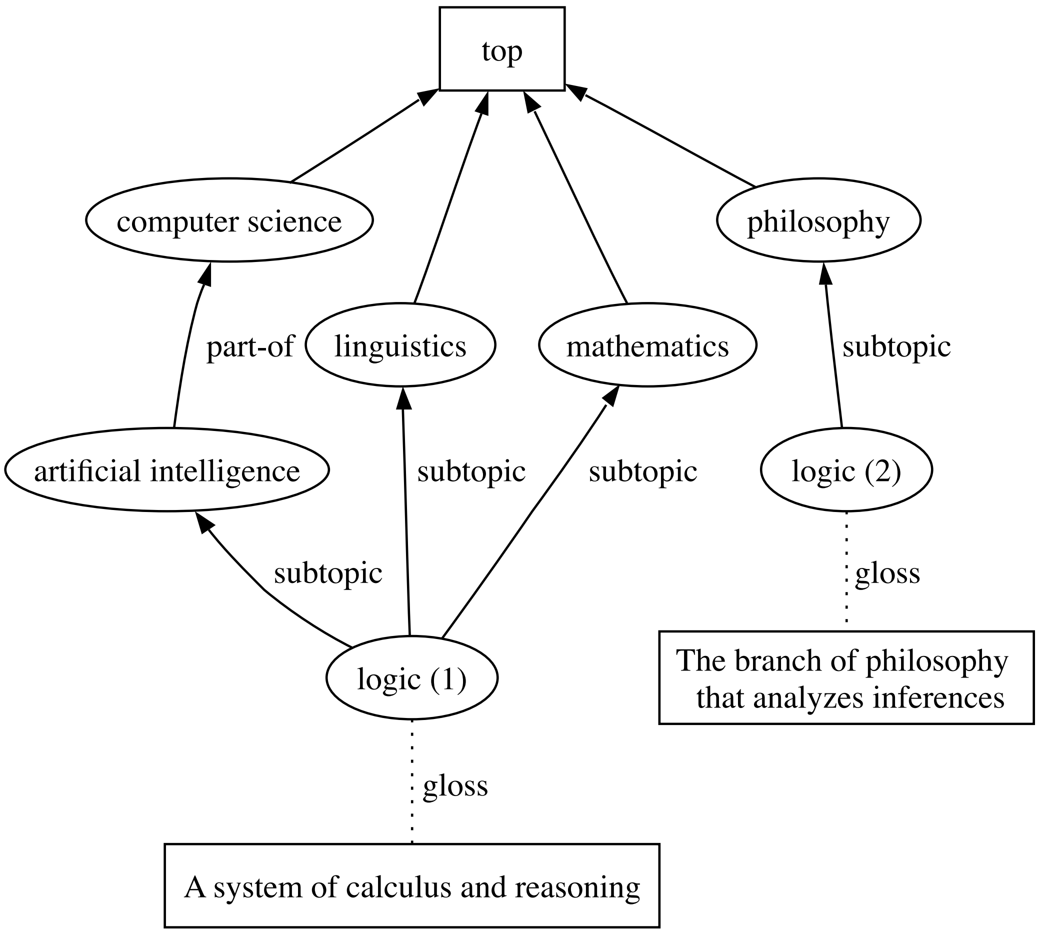


Figure 3.1: Glosses help distinguish two "senses" of the same topic.

topic (e.g., pragmatics *is part of* linguistics) and that a topic *is an instance* of an already defined topic (e.g., K4 *is an instance* of modal logics). Therefore, we included the following three relations, explained in Sections 3.4.1, 3.4.2 and 3.4.3, respectively:

1. `subtopic` (ST),
2. `part-of` (PO),
3. `instance` (IN).

In addition to these, we selected a small number of relations meant to account for important relations typical of the domain at hand: that a topic expresses the feature, or the "nuts and bolts," of the domain (for example, completeness and variable to logic); that a topic is a theorem (or lemma and the like) that holds in a given area; that a topic is a computational tools used or developed in a given area, e.g., for instance, a theorem prover for logical systems and a parser for a natural language; and that a topic provides a historical perspective on a given topic, e.g., the treatment of quantifiers in aristotelian logic with respect to the current theory of quantifiers. These considerations lead us to consider the following domain-specific relations:

1. `features and internal machinery` (FI),
2. `mathematical result` (MR),
3. `computational tool` (CT), and
4. `historical view` (HV).

The domain-dependent relations are presented in Section 3.4.4. Figure 3.2 depicts a small fragment of the map.

3.4.1 Many Flavors of ISA

The notion that something *is a* type or specification of something else, is a fundamental notion for (at least) human cognition, linguistics, and mathematics. When called *subclass* [Brachman, 1983, Cruse, 1986, 2002], it is the most widely used hierarchical relation in the literature, but it is also widely known as *ISA* (i.e., is a), *AKO* (i.e., a kind of), *subsumption*, and *hyponymy* (inverse: *hypernymy*). As noted elsewhere [Brachman, 1983, Cruse, 2002], and as the abundance of different names for it should suggest, the apparent simplicity of the term subclass hides a gamut of slightly different semantics.

A common understanding of the subclass relation implies a reference to objects in the *real* world. Objects are then grouped into classes, which can be further specified to define subclasses (or subsets, subcategories). For example, a 'daisy' can be intensionally defined as "A small flower with white petals and a yellow center, which often grows in grass."[2] As a class, extensively defined, it would include all daisies in the world. Since all daisies are 'flowers,' it follows that the class 'daisies' is a subclass of the class 'flower.' Although intuitively appealing, this understanding is more problematic than it may seem at first: because it is not always possible to identify an object in the physical world to which to refer and, most importantly, because the grouping depends on some definition of the object to classify, i.e., it depends on the selection of suitable features describing an object.

The application of a strict set-theoretical view is not suitable to cover the (large) domain considered in this work, namely the area at the interface between logic and linguistics, although specific subareas do admit a formal description based on set theory. For instance, a system of logic can be taken as the set of formulas formed using

[2]This definition is taken from the Cambridge Dictionary, obviously not a dictionary for botanists. In the same dictionary we find the following definition for 'flower:' "The part of a plant which is often brightly colored with a pleasant smell, or the type of plant that produces these."

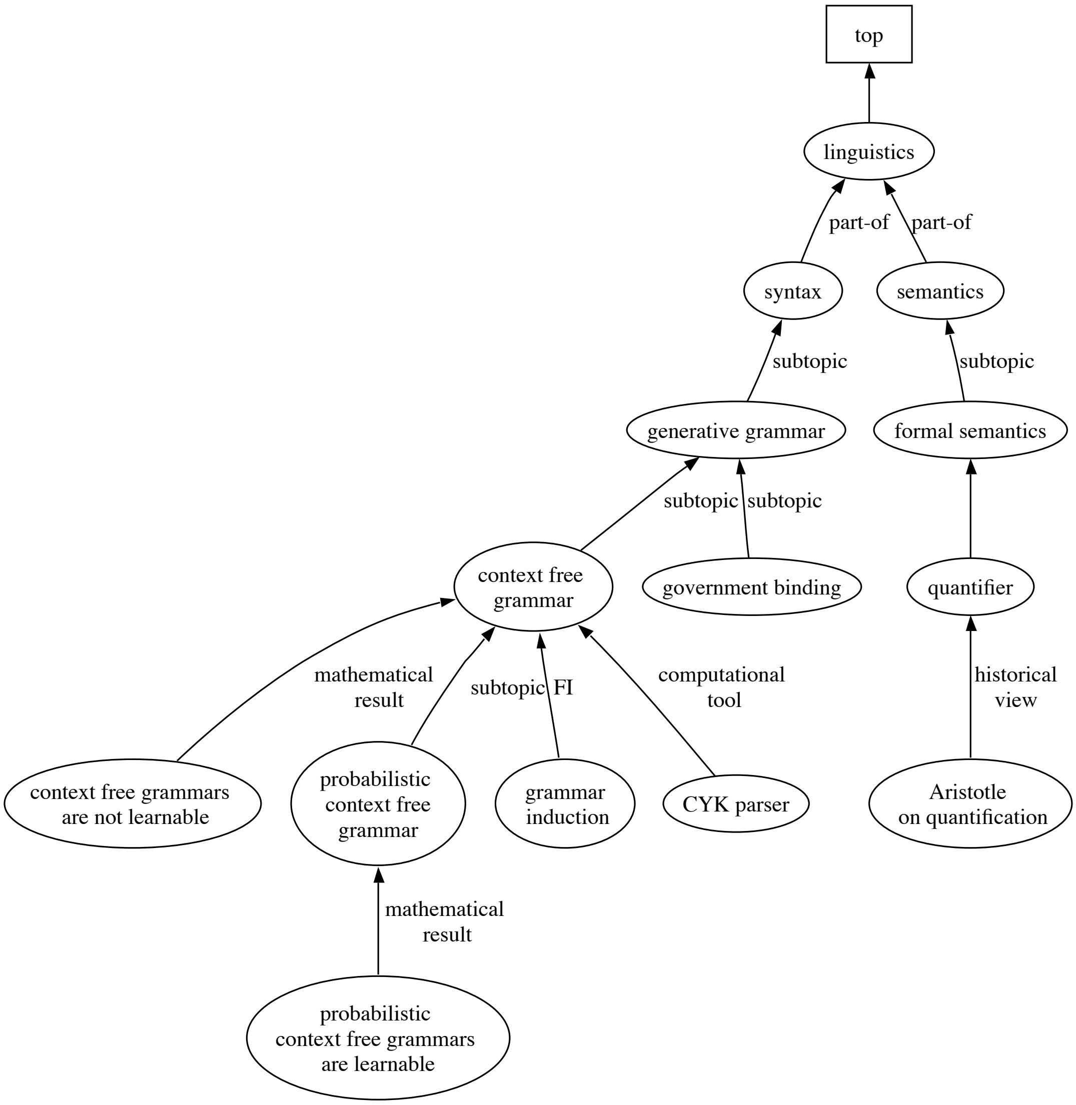


Figure 3.2: A fragment of the LoLaLi map.

a given alphabet and a given set of rules: the larger the alphabet, the more expressive power of the logic. A logic, *A*, can then be included in another logic, *B* (consisting of more formulas than *A*) and in this sense *A* is a subclass of *B*; on the other hand, since *B* utilizes a larger alphabet, it can be considered a "specification" of *A*, and often presented to non-expert readers "after" *A* for historical reasons. So for example, in a set-theoretical view, propositional calculus would be a subclass of first-order logic because propositional calculus has a smaller alphabet than first order logic, and all propositional formulas are also first order logic formulas.

Some authors, like Cruse [1986, 2002], distinguish a generic notion of subclass (then called simple hyponymy) from a more precise relation of taxonomy. A simple hyponymy corresponds to the sentence 'An X is a Y,' as in 'A white table is a table,' while a proper taxonym corresponds to the sentence 'An X is a kind of/type of Y,' as in 'A bedside table is a table' (cf. Section 2.4). It turns out that the distinction between the two relationships is not always clear, since 'A brown bear is a bear' can easily be interpreted as a taxonomic distinction. In fact, the notion of simple hyponymy seems to belong more to an investigation of natural language semantics than to a discourse on ontological relationships.

Several principles and heuristics have been proposed for "correctly" assigning the subclass relation to pairs of concepts. Here we only mention the most recent and influential. Cruse [1994] suggests that the decision about classifying subclasses should be made on the basis of a coherent *perspective* of the hypernym. The idea is that when organizing a class into subclasses (or taxonomies) we have to select one or more features and use them to identify subclasses. To use his example, 'stallion' is a bad taxonomy for 'horse' because it does not adopt the same perspective, while 'ash blond' is a good taxonomy for 'blond,' because they both adopt the same perspective. Later on, trying to further clarify the notion of perspective, Cruse [2002] proposes to choose categories that are (a) internally cohesive, (b) externally distinctive and (c) maximally informative. Storey [1993] observes that the subclass relation is often confused with various types of part-of relations, because they all imply membership of an individual in a larger set (cf. next section). In practice this distinction leads to an implicit heuristic for deciding when a subclass relation should be assigned.

Others [Guarino and Welty, 2000, 2002a,b] have made attempts to give a general, well-defined notion of the subclass relation by focusing on the ontological nature of the arguments involved in it, more than on the relation itself, and using the philosophical notions of identity, unity and essence. Using these criteria as landmarks, they analyzed several ontologies and found many inconsistencies. Unfortunately the complexity and abstraction of their method has hampered a broad adoption in real life applications.

The considerations expressed above, together with Requirement C (Section 3.1) and observations made during the user studies reported on in Chapter 4, made us opt for the notion of subclass that corresponds to the taxonomic relation of 'is a type of'. Then, in order to avoid confusion with a very loaded term, we prefer to talk about *subtopic*. For example, formal semantics `is a subtopic of` semantics, or modal operator `is a subtopic of` operator. So, in the view we adopt, first order logic and propositional calculus are both subtopics of logic as they are both formal systems, but with different properties. Figure 3.3 presents a set-theoretical view of them (a) and the organization used in LoLaLi (b). Similarly, in our view hybrid logics are subtopics of modal logics, despite the fact that syntactically hybrid logics are extensions of modal logics (i.e., they include more operators than just modal operators).

Summarizing, given the requirements imposed in Section 3.1 and the breadth of the domain at hand, we opted for an intuitive notion of subclass, that we call `subtopic`.

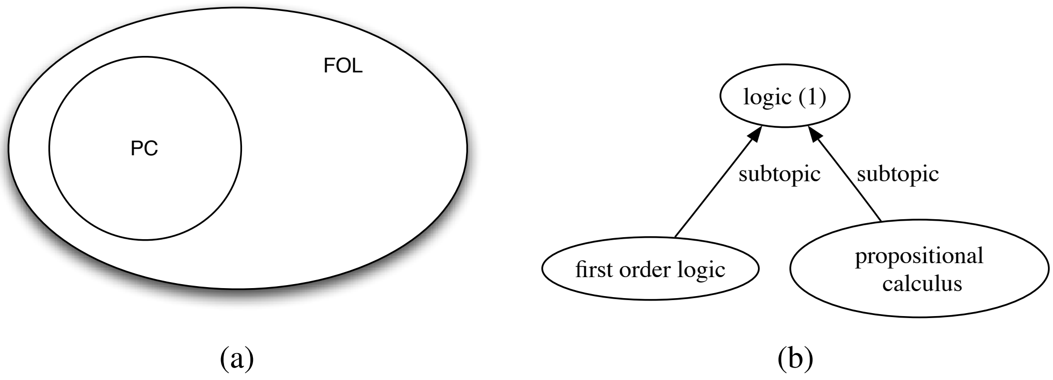


Figure 3.3: (a) First order logic (FOL) and propositional calculus (PC) viewed as sets of formulas. (b) First order logic and propositional calculus as subtopic of logic.

3.4.2 Part-of

The relation between the parts of a whole is called *part-of*, *part-whole* or *meronymy* (inverse: holonymy) [Artale et al., 1996, Cruse, 1986, Simons, 1987]. While the decomposition of classes into subclasses leads to taxonomies, the decomposition into parts leads to partonomies [Tversky, 1990].

While taxonomies have been studied extensively in many domains, this has not happened for partonomies, although from a cognitive point of view, there is evidence suggesting that children can process meronymy relations earlier than taxonomic relations [Inhelder and Piaget, 1964]. The development of a theory of parthood at the beginning of the last century aimed at defining a single theory that, unlike set theory, is founded only on concrete entities, but there is no unanimous agreement on its semantics. The basic problem of mereology as a field of study is that there appear to be various, often inconsistent, semantics associated with the term 'part-of.' In fact, the same expression 'to be part of' is used to describe arguably different situations, as in: 'A leg is part of the body,' 'A boat is part of a fleet,' and 'A window is part of a car.' Consider the following example taken from [Salustri, 1998].

1. A piston is a part of an engine.
2. An engine is a part of an automobile.
3. An automobile is a part of a fleet.

Each statement, on its own, is perfectly reasonable. Furthermore, from statement 1 and 2 we can reasonably deduce that a piston is a part of an automobile, but from all three statements, can we reasonably deduce that a piston is a part of a fleet?

Some authors, like Srzednicki et al. [1984] and Gruber [1992] consider the parthood relation to be single, universal, and transitive, meaning that all distinctions about types of parts are really conceptualizations and are not rooted in reality; they use first-order logic to introduce sufficient predicates to distinguish between kinds of things.

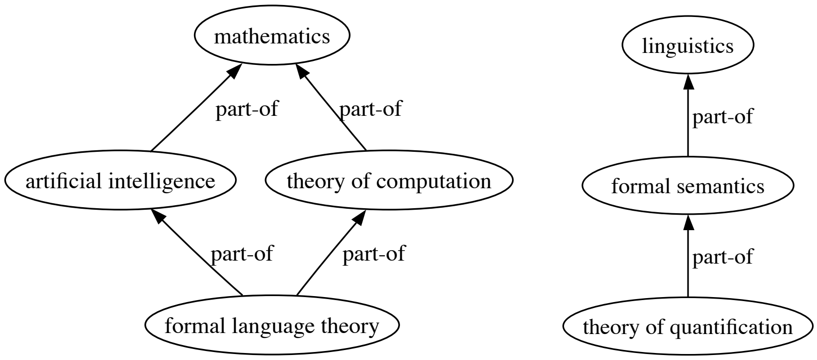


Figure 3.4: Two examples of chains of part-of relations in the LoLaLi map.

Artale et al. [1996] believe that a proper mereology must handle the transitivity problem directly by admitting distinctions between different part-of relations. In that approach, different part-of relations are explicitly defined to handle different conceptualizations (e.g., assembly/component versus space/region), and transitivity is not preserved across them.

In the LoLaLi map, the relation `part-of` (PO) is applied to abstract entities and its meaning is taken in an abstract sense, to indicate the main *parts* into which a domain is articulated. For example, syntax, semantics and pragmatics are all parts of linguistics. We also consider this relation to be transitive. We believe that, for navigational purposes this is most natural/appropriate. Figure 3.4 shows some examples of chains of `part-of` relations currently included in the LoLaLi map.

3.4.3 Instance

As previously hinted, the relations `is an instance of` (IN) is taken in many semantic structures to mark the belonging of a physical object to a class. In the LoLaLi map, this relation is used to relate specific examples to a more general group. Specifically, this happens often in the case of logical systems, when a specific set of axioms in a given language is studied *per se*. This is a common situation, e.g., in modal logic, where logical systems such as K4, S4, S3, KD4 are all `instances` of modal logic.

3.4.4 Domain Specific Relations

In this section we present the relations introduced in order to capture the relationships between topics specific to the specific domain we deal with.

Features and Internal Machinery

The relation `features and internal machinery` (FI) is a domain-dependent relation, introduced in order to point out to our end users what the fundamental notions

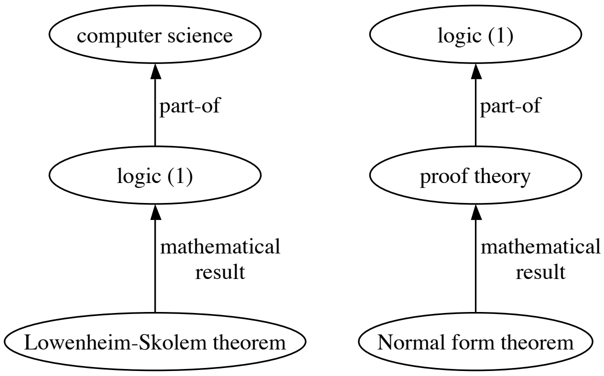


Figure 3.5: Two examples of the relations `mathematical result` (MR) in the LoLaLi map.

in the subject are, i.e., the nuts and bolts of the domain. Also notions used to classify topics in a domain are covered by this relation. Examples of these relations are: variable and constant for logic (1), and part of speech in linguistics. Completeness and soundness are both `FI` of a logic (1), while arity is a `FI` of operator, since it indicates how many arguments an operator takes.

Mathematical Result

The relation `mathematical result` (MR) links a mathematical statement to the topic representing the area to which it is relevant. Theorems, corollaries (i.e., propositions inferred immediately from a proven proposition with little or no additional proofs), and lemmas (i.e., an auxiliary proposition used in the demonstration of another proposition) are all `mathematical statements` of their supertopic. Figure 3.5 depicts two examples taken from the LoLaLi map.

Computational Tool

The relation `computational tool` (CT) is another domain-dependent subtopic relation. This relation has been introduced to emphasize the role of computational systems like theorem provers in logic or parsers in computational linguistics. For example, SPASS [Informatik, 2006], an automated theorem prover for first-order logic with equality, is a `computational tool` for first order logic, a CYK parser is a `computational tool` for context free grammars (Figure 3.2), and HyloRes [Areces et al., 2001, Areces and Heguiabehere, 2002] is a `computational tool` for hybrid logic.

Historical View

The relation `historical view` (HV) has been introduced to account for the historical evolution of topics. For example the topics Aristotle on quantification and Frege on quantification are historical views of the more general topic quantification. Given the coverage of the handbook, this relation turned out to be the least used in the LoLaLi map.

3.4.5 Non-Hierarchical Relations

The relations `related` and `antonymy` have been introduced to make explicit the connections between topics to which no hierarchical relation is applicable. They are non-hierarchical in the sense that there is no implication that one of the topics involved in the relation is more "general" than the other; these relations are symmetric.

Related

The relation `related` is a non-hierarchical binary relation, mainly introduced for ease of navigation among topics. It connects concepts that are for some reason judged as similar or connected, such as quantification and quantifier, or reference and referent. In order to provide more information about the nature of each pair of topics, whenever possible a comment is added to explain to the reader what kind of connection links the two concepts. We assume that the `related` relation does not hold between pairs of concepts already connected by any hierarchical relation, but siblings under the same parent can be linked as `related`.

Antonymy

Pairs of topics that are opposite to one other are linked by the `antonymy` relation. This relation used to be widely used in dictionaries. Though not that frequently used anymore in modern dictionaries, the antonym relation is an important relation in WordNet [Fellbaum, 1998]. For example, the following two word senses in WordNet are antonyms of each other:

- black, sense 6, ((board games) the darker pieces)
- white, sense 9, ((board games) the lighter pieces)

In the LoLaLi map the `antonymy` relation only applies to pairs of topics that are siblings under the same parent and hold the same relation to the parent (usually the `feature and internal machinery` relation). Examples are completeness and incompleteness under logic (1).

3.5 The LoLaLi Map: Features

In this section we present the figures concerning the current status of the LoLaLi map and describe its structural features.

The LoLaLi map is organized into four main branches: computer science, mathematics, philosophy and linguistics. These topics are gathered together under an empty

Total # of topics	547
Number of subtopic relations	629
Number of part-of relations	42
Number of historical view relations	6
Number of instance relations	6
Number of mathematical results relations	11
Number of computational tool relations	2
Number of FI relations	7
Maximal depth	9
Average outdegree	1.1
Maximal outdegree	32

Table 3.1: Details about the LoLaLi map.

topic called TOP, which serves as the root of the map. Under those four first level topics all other topics find a place. The nature of the relations between each of these four topics and TOP is not specified, since TOP is only used for ease of computation and for technical reasons. At the time of writing, the LoLaLi map consists of over 500 topics, mainly under the branch logic (1). Table 3.1 presents key figures concerning the map.

Mathematically, the LoLaLi map is a graph, where nodes are topics and arcs are relations. When restricted to hierarchical relations, the map is a connected, acyclic and directed graph.

Connected Graph

It is always possible to find a path from any topic in the graph to the top node. This means that disconnected topics or isolated subgraphs are not allowed. This feature is also used as an intuitive constraint in the process of building the map: all new topics must be attached to existing ones. Thus, at no stage of the creation process do we have isolated subgraphs. This feature is important also from the end user perspective, as it ensures that we have, at any point of time, a map that is always completely browsable (i.e., both top-down and bottom-up).

Oriented and Acyclic

Since edges in the graphs are directed (the edges have arrows), the graph is oriented. The graph may not contain cycles of hierarchical relations (i.e., `subtopic`, `parthood`, `instance`, `mathematical result`, `FI`, `computational tool`), meaning that no concept can be an ancestor of itself. Obviously, the non-hierarchical relations `related` and `antonymy` are not affected by this constraint, because these relations do not imply a direction between the topic they connect.

It can be argued that cycles of relations do not represent a flaw, since many dictionaries do have cycles in their definitions, but in our view this would be a source of problems and confusion for our map, especially for its navigation, because it would lead to counterintuitive situations of more general topics being placed underneath more specific ones.

Multiple Parenthood

By using a structure with multiple parenthood we stress the fact that a topic can be relevant, and therefore connected, to more than one area. In this sense, the multiple parenthood structure also allows one to use multiple classifications of the same object: for example a modal operator is a particular kind of operator (i.e., a subtopic of it), but also part of a modal language (Figure 3.6).

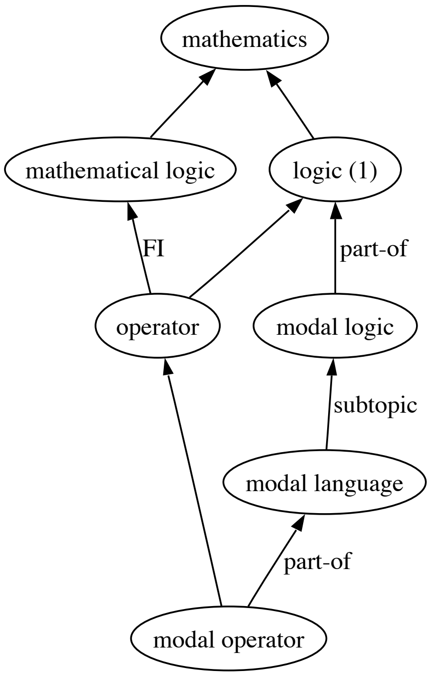


Figure 3.6: Multiple parents as multiple classification: the case of operator.

Multiple parenthood structures are typical of many multilingual thesauri, although in semantic networks they are often called “tangled hierarchies” [Fahlman, 1979]. WordNet is an example of such a tangled hierarchy.

Different Degrees of Kinship

In a graph there can be more than one path connecting a pair of nodes. In the LoLaLi map we also allow topics to have a hierarchical relation with both a topic and one of its direct descendants: we say that the two topics have more than one degree of kinship. Figure 3.7 presents such a case, where the topic logic (1) can be reached in two steps from top, either passing through mathematics, or through linguistics, or through computer science. The same topic logic (1) can also be reached in three steps from top, by passing through computer science and artificial intelligence. In other words, the topic computer science is both parent and grandparent of logic (1).

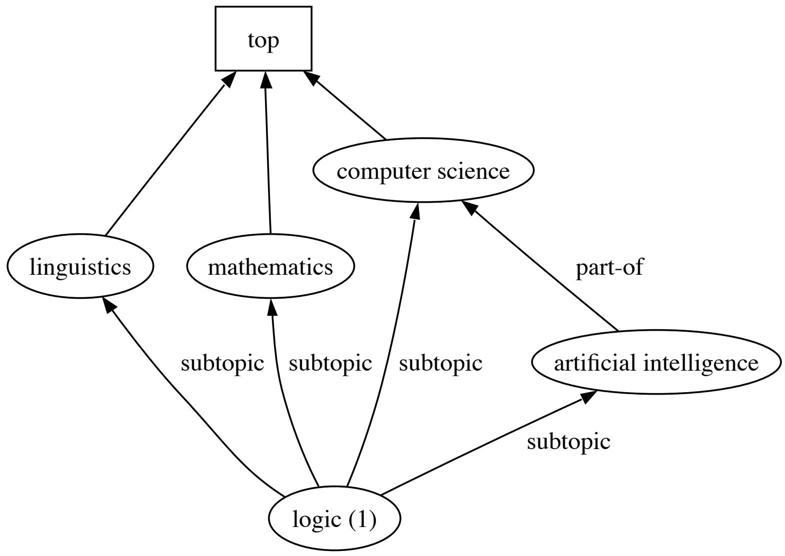


Figure 3.7: Logic (1) can be reached from top by more than one path. Different degrees of kinship: logic (1) is connected to both computer science and artificial intelligence.

3.6 Discussion

When dealing with an area as broad as the interface between logics and linguistics, the difficulties in selecting relations that are appropriate to the domain, in type and in number, and that can be informative to the end user without overwhelming her. In this section we report on what we learned from our implementation of the LoLaLi map.

3.6.1 Dealing with more Relations

Since the *Handbook* treats the domain at the interface of logic and linguistics, it includes topics that encompass inter-connected areas and that have contributed to the discovery of more connections among them. Following the style adopted in the *Handbook*, we call these subjects *frameworks*. An example of a framework is 'categorial grammar,' in which concepts from linguistics, computer science, and mathematics play an important role. Table 3.2 lists these three areas with the relevant subtopics for 'categorial grammar.' For a more pictorial view, consider Figure 3.8, where solid and dashed

Categorial Grammar relates to:

Linguistics	*Computer Science*	*Logic*
quantification	complexity theory	proof theory
pronoun		lambda calculus
coordination		category theory
polarity		modal logic
		type theory

Table 3.2: A summary of connections between the categorial grammar (a framework) and other areas.

lines are used to represent connections among some of the above-mentioned areas that were previously known and unknown, respectively. Some connections among these fields had already been established before the beginning of categorial grammar as an independent area of study (solid lines). For example, connections between complexity theory, proof theory and modal logic (in logic), and connections between lambda calculus, type theory and quantification (in linguistics). Other relations were created by categorial grammar itself (dashed lines): between proof theory and quantification, between quantification and category theory, between category theory and pronouns, coordination, and polarity items (the latter three being linguistic phenomena). In order to accommodate these connections, the LoLaLi map should use additional specialized relationships, but in so doing we would only talk to advanced readers, experts in the field or researchers. This choice would lead us to violate both Requirement B and Requirement C in Section 3.1 (about informativeness and information overload, respectively). Therefore we decided to introduce a topic categorial grammar connected to the relevant topics (i.e., linguistic phenomena, quantification, lambda calculus and so on) by means of the non-hierarchical relation `related`. A comment is included to explain the type of relatedness.

Another issue arose when treating topics for which more than one classification schema was possible. Consider, for example, the notion of operators in logic. Operators can be classified by arity (an operator can be unary, binary and so on) or by type (it can be a modal operator, a truth functional operator and so on). By adopting both perspectives, we can model different taxonomies of operators at the same time. Note that this issue is connected to the one of *perspective* we mentioned when discussing the subclass relation in Section 3.4. Instead of adopting a single "perspective" to distinguish subtopics of a topic, we list all possible taxonomies. This way we are able to give as complete an account as possible of the various subtopics of certain topics.

3.6.2 More on Subtopics

Intuitively, if a topic has a property, its subtopics should also have that property. For example, if 'apple' is a subtopic (rather, subclass) of 'fruit,' and the property 'has seeds' holds of 'fruit,' then it also holds of 'apple.' In the case of the LoLaLi map, the decision to adopt what we called an intuitive view of the subtopic relation (as opposed

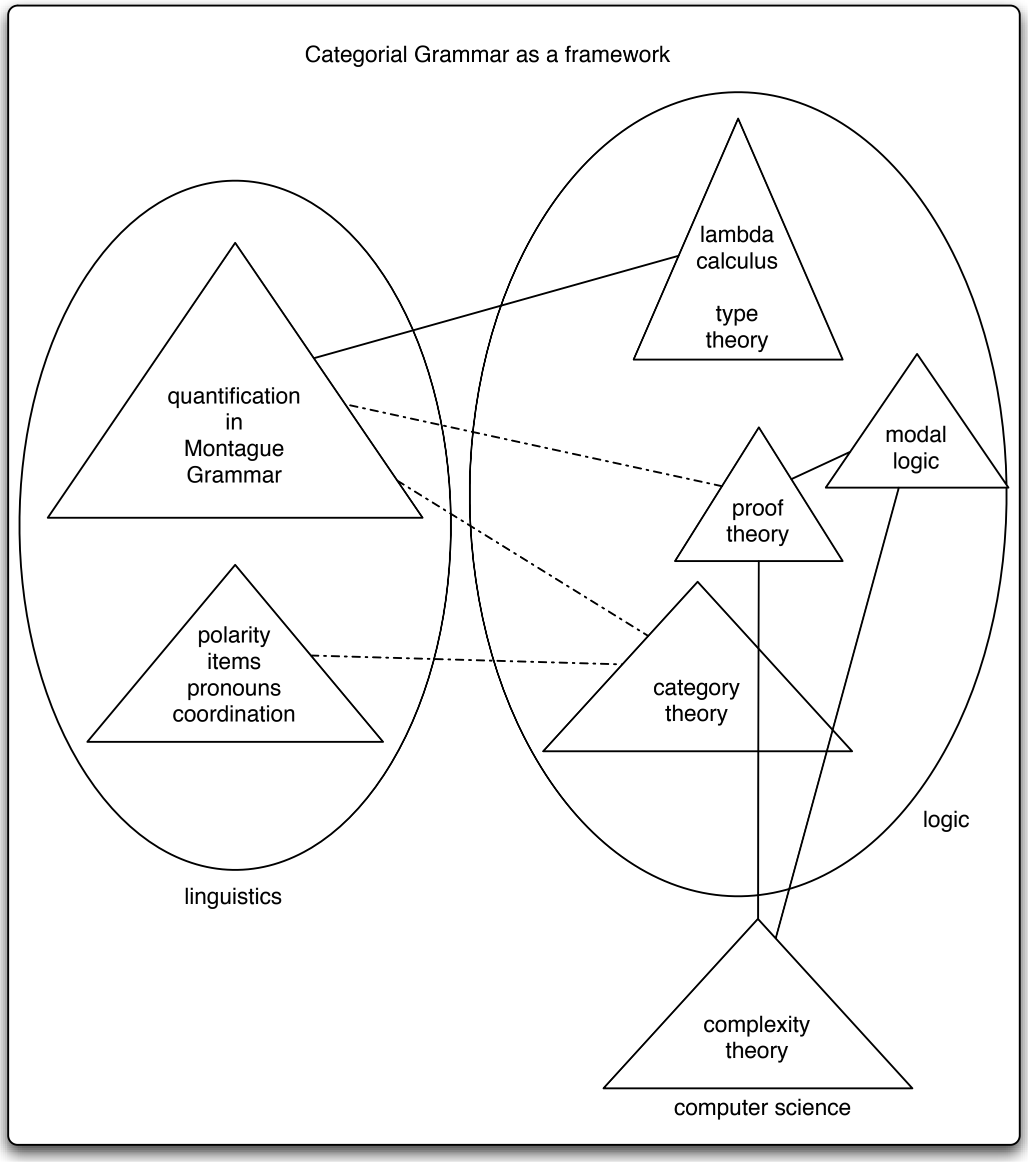


Figure 3.8: Categorial grammar as a framework. Dashed lines stand for relations created by categorial grammar. Solid lines indicate previously known relations. Ellipses group together topics in the same area.

to the set theoretical one) has interesting consequences in this respect. Consider the above-mentioned theorem prover HyloRes, which in the LoLaLi map is a computational tool for Hybrid logic (Figure 3.9). Had we adopted a set-theoretical perspective, modal logic would actually be a subtopic of hybrid logic, because hybrid logics add hybrid operators to modal logics (i.e., the set of hybrid formulas is larger than the set of modal formulas). As a consequence, HyloRes would also be a computational tool for modal logic, which is actually true, but arguably not very informative (hybrid operators

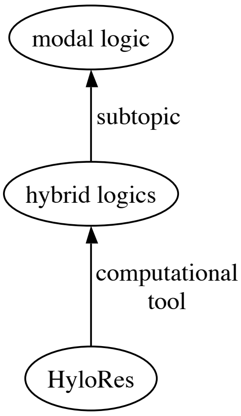


Figure 3.9: Hybrid logics in LoLaLi.

should be ignored). In this sense, our notion of subtopic relation matches well with the purpose of the LoLaLi map.

As another example, let us consider SPASS, a theorem prover for first order logic, and the example previously given about the modeling of first order logic and propositional calculus (Section 3.4.1 and Figure 3.3). In the LoLaLi map, both First order logic and propositional calculus are subtopics of *logic*, and SPASS is *only* a subtopic of FOL, which suits our purposes well. If we had taken the set-theoretical perspective (Figure 3.3 (a)) we would have "learned" that SPASS is also a theorem prover for PC which is, again, true but not very informative.

Let us consider the fragment of the map depicted in Figure 3.10 and the relation `mathematical result`. Probabilistic context free grammars is a subtopic of context free grammars, and the map reports the following mathematical results: "probabilistic context-free grammars are learnable," while "context-free grammars are not learnable." Both theorems refer to the concept grammar induction (the possibility of learning a grammar), but in opposite ways. What normally happens in the case of subclass relations is that the amount of specification (or properties) increases when going down the line of subclass. As we have just mentioned, this also happens in our examples, as by adding information (the constraint for the grammar to be probabilistic) there is a "new" result that does not hold for the supertopic context free grammar. The difference between this situation and other taxonomic structures is that, when looking "down" from supertopic to subtopic (from CFG to PCFG) we cannot assume an inheritance of properties. On the other hand, when looking up, from subtopic to supertopic, we have the same specificity of information (theorem on learnability) but of different type/content (whereas in usual taxonomies, supertopics are less specified than subtopics): we have specific results (mathematically proven) also at the level of supertopic and not only at the level of subtopics. If inheritance of theorems were allowed, we would mistakenly conclude that probabilistic context free grammars, just as context free grammars, are not learnable. Once again, given the specificity of the domain at hand, it turned out

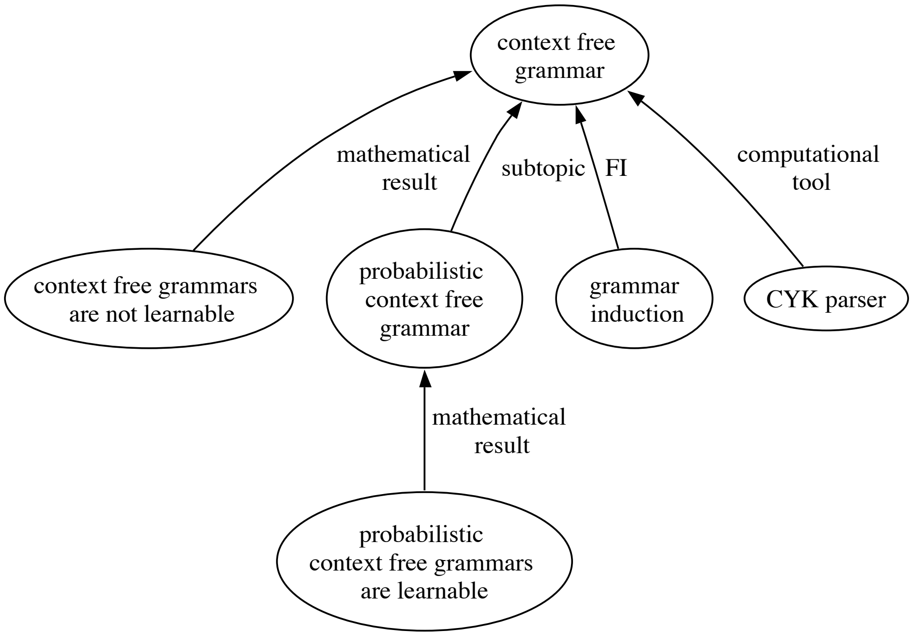


Figure 3.10: Two mathematical results for context free grammars.

to be a good choice to avoid the adoption of a set-theoretical notion of subclass and avoid the notion of inheritance altogether. These anecdotal observations support our modeling decision to adopt an intuitive interpretation of the subtopic relation.

3.7 Editing and Managing the LoLaLi Map

In this section we present and discuss the solutions we adopted to edit and maintain the map, in terms of the software used for editing, the choices made to model the LoLaLi map in RDFS, and how we made the map accessible through the web.

3.7.1 A Bit of History: First Attempts

The first version of the LoLaLi map, consisting of about 100 topics, was encoded as a collection of Prolog facts connected by BT/NT relations. The visualization of these topics was done by means of a Java applet (for more details about this version, see [Ragetli, 2001]) that resulted in something that was difficult to read and not scalable.

A second version of the LoLaLi map was encoded in XML [XML, 1998], and provided with a Document Type Definition (DTD). Each topic was represented as an XML document, and all pieces of information attached to each concept, i.e., gloss, description, references to the parent and child topics, metadata about the author of the node and the last update, and references to related and antonym topics were allocated to distinct elements. The entire hierarchical structure was stored in a relational table. An HTML version of the map was periodically generated and published online.

The XML encoding of the map had major drawbacks, especially for the management of references to parent, child, and related and antonym topics. Moreover, the lack of suitable tools for managing the XML repository hampered the gathering of contribution to the map from authors. We explored the possibility of using XLink [XML, 2001] (an XML based language able to manage possibly multidirectional links between XML documents or parts thereof), but we were confronted with the lack of mature software tools. As it turned out, it was more convenient to store the map structure in a relational table instead of using an XML document. Finally, search within the collection of XML documents was not sufficiently supported (only string search was possible; at the time it was not possible, for example, to perform a structured search, i.e., within elements), nor was visualization of the map (only possible by means of the HTML interface, to be generated after every update). In practice, very little support for the editorial process was available.

In order to cope with these limitations, the map was converted into RDFS format, the vocabulary description language for RDF, because of its emerging status as an accepted standard and the large amount of software already available for it. In particular, the adoption of RDFS allowed us to take advantage of Protégé, an ontology editor able to produce RDFS output, and of off-the-shelf tools (i.e., Sesame, see Section 3.7.3) to enable access to the map through the web. For more details about the conversion from XML into RDFS, and about the use of Sesame as back-end, see [van Schie, 2003]). In the following subsection we present the main features of Protégé and discuss our modeling decisions.

3.7.2 Protégé and RDFS Modeling

Protégé [Gennari et al., 2003, Noy et al., 2000, 2001] is an ontology editor developed at the University of Stanford (precisely, at the Stanford Medical Informatics group within the Stanford University School of Medicine.)

Protégé is an ontology editor with an easy-to-use graphical interface that allows one to enter *classes* by means of forms. It support the visualization of class hierarchies (in an indented-tree like fashion), properties and instances. Also, its architecture is well suited for the inclusion of plug-ins, many of which have been made available by a large community of users. Protégé allows one to save and export data in various formats, including RDFS.[3] Concepts in a domain are modeled as *classes*, which can have *properties*, while actual data is modeled as *instances* of class(es).

The version of Protégé we used[4] allowed the modeler to impose constraints, such

[3] As with RDFS, Protégé's internal format also uses the notions of classes, class properties and instances of classes; for a detailed discussion of the relationships between RDFS concepts and native Protégé concepts see [Noy et al., 2001].

[4] Protégé 1.8.

as cardinality constraints on the number of required properties and the definition of inverse properties, that could however not be exported into RDFS. For this reason we decided not to use them at all. Also, the version of Protégé that we used supported one namespace only, it was stand-alone, did not manage multiple users working on the same project, and it did not implement user rights restrictions. This fact forced a very centralized work flow, with one person controlling all the data.[5]

Each concept in the LoLaLi map is modeled as an RDFS class endowed with a number of properties to accommodate glosses, topic descriptions and link(s) to the *Handbook* (these pieces of information are meant to be shown to the end user), and comments, creation/modification timestamps, and data about the author of each topics (these pieces of information are mainly for editorial purposes). A Protégé *metaclass* (ConceptClass) is used to serve as a template for creating new classes. Since the subtopic relation used in the LoLaLi map does not exactly coincide with the strict set-theoretical notion of subclass, we opted for modeling all relations, indistinctly, as properties.

When the data was first converted into RDFS, all relations were rendered as a property called 'Unspecified subtopic' whose refinement (according to the relations presented in Section 3.4) was carried out in parallel with the conversion and implementation of the GUI. As we will see in Chapter 4, the user studies were run in this phase and they actually contributed to the final definition of the set of relations.

By modeling all relations as properties (including the relation subtopic) we miss the opportunity of using the built in semantics of subclass in Protégé. We argue that for our purposes the loss is minor. First of all, the lack of inheritance of slots is not a problem, because this feature was not used in the modeling of the map. Also, for editing purposes we make use of a metaclass as a template to create new classes. A possibly useful application of the Protégé built-in subclass relation is that we could in principle check for cycles of subclass relations and provide an aid to the maintenance of the map. We lose this possibility because we cannot specify such a constraint at the level of class slots. In our experience this was not a problem, but it could be a problem if the map grew considerably or if more people were allowed to modify the repository. As for the maintenance of the map, Sesame proved to be a useful tool, as we defined a set of queries (for example, to check for cycles, and extract all subtopics of given topics) that was regularly run against the repository.

3.7.3 Accessing the Map through a Web Browser

In order to make the LoLaLi concepts browsable through a web browser, we needed a middleware system to interface the RDFS repository with the web. One way to do it is to parse the RDFS repository and convert it into static HTML pages. Of course, this solution implies that any change in the RDFS repository requires that the entire repository

[5] Many of the limitations of Protégé 1.8 have been overcome by the latest versions of it. The latest version of Protégé does support multiple users, although it still does not support rights management, and can be extended by a plug-in that allows one to edit ontologies in OWL [Knublauch et al., 2004].

be re-converted. Another solution is to use an RDFS repository manager connected to the user interface: for this purpose, we adopted Sesame [SESAME, 2005], an open source RDFS-based storage and querying facility. Sesame uploads the RDFS repository, parses it, and enables querying it, by means of query languages such as the RDF Query Language, RQL [Karvounarakis et al., 2002] and SeRQL [SeRQL, 2005].

At the time the LoLaLi map was implemented, the RQL language was the more stable and developed language. RQL is similar to the well-known relational query language SQL. We created fixed queries corresponding to the navigational actions performed on screen. Each click in the interface (see Chapter 4 for details about the interface) triggers an RQL request to Sesame. The solution RDFS + Sesame + RQL proved to be a suitable solution for us (although with some overload by the client, as RQL returns data in XML format, to be parsed on the client side, in tabular form, therefore with duplications). At the time of writing the RQL language is still supported but no longer actively developed. Currently, SeRQL is becoming the default language used to query a Sesame repository.

Summarizing, Protégé was used for editing the map (after conversion in RDFS format) and Sesame was used to access data in the repository and pass it to the user interface. Sesame cannot be used as a search engine for the end user (nor is it advisable to use it to support the everyday maintenance work on the map), because of the complex query language and the format in which results are shown, Instead, we provided the map with a search engine tailored to it, that allows any text to be typed in by the user and searches against every piece of information available in the map, but with meaningful assumptions on common searches and informative results (for details about the search engine see [van Hage, 2004]).

3.8 Conclusions

In this chapter we described and discussed the modeling of the LoLaLi map, covering the area of logic and linguistics. We presented the requirements we imposed on the map and the approach used to design and populate it. We also discussed its features and presented its implementation.

The relations used within the map were especially chosen to meet two of the user requirements we imposed: that the map be informative for the end user we address, and that it avoid information overload on the user. The resulting set of relations, selected on the basis of a mixture of empirical, pragmatic, and principled considerations, directly addresses our primary concern, namely the definition of a map oriented to human browsing and navigation. Anecdotal observations on the domain confirm that the adopted solution is appropriate to our aims, while the user studies on which we report in next chapter will shed some lights on the interaction of the users with the map.

The LoLaLi map is not endowed with a formal semantics, for the twofold reason that it is neither necessary for the type of users we aim at, nor for the type of connection to the text that we are investigating and reporting on in Chapters 5 and 6. The drawback of this decision is that the possibility of consistency checking mechanisms are not exploited, but given the scope of our work, and the envisaged usage of the map, this is not a major problem. Instead, what turned out to be a source of difficulty during our work was the process of populating the map because we did not distinguish between different roles for the people involved in it and consequently could not benefit from an organized editorial flow. In Chapter 7 we reflect on this and related issues.

Finally, we remark that in the short time since we first started working on the encoding of the LoLaLi map in semantically oriented languages the field has greatly progressed, in terms of new and richer semantically-oriented languages (e.g., OWL), improved editing tools for ontologies, more sophisticated languages to query RDFS repositories (e.g., SeRQL). We are dealing with rapidly evolving technologies, where languages, tools and facilities are proposed, accepted as standard, and then given up in rapid succession.

In the next chapter we move our attention to the end users of the LoLaLi map: we present the graphical interface dedicated to them and report on the user studies we performed.

CHAPTER 4

Interacting with the LoLaLi Map

In Chapter 3 we described the design and implementation of the LoLaLi map, now we address the issue of providing the end users of the LoLaLi map with a suitable interface to it. Recall that we assume that the *end users* of the LoLaLi map may have some, but not deep, knowledge of the domain; they are not assumed to have a background in knowledge engineering. Instead, they can be assumed to have a fair level of familiarity with computers and the Internet. This type of user should be enabled to conveniently visualize, browse and search the entire LoLaLi environment consisting of the map and the handbook linked to it.

Currently, many visualization tools for graph-like structures (in particular ontologies) are available (e.g., Jambalaya [Storey et al., 2002], OntoViz [Sintek, 2003], IsaViz [ISAVIZ, 2004], Kaon [Oberle et al., 2004]), but they usually address experienced audiences, such as knowledge engineers or other types of experts. Also, the theoretical work on this issue has mainly focused on experienced users [Ernst et al., 2003]. In contrast, the visualization of structures like the LoLaLi map for end users of the type we envisage has remained relatively unexplored. The challenge is to give our end users a sense of what the structure is like, to maintain a sense of orientation in it, where appropriate to present detailed pieces of information about each element in the map, and at the same time hide technical details from the user.

We start our work by making explicit the requirements that we impose on the interface, then, after reviewing related work, we present our proposal. The interface we propose aims at balancing a global view of the map with a focus on the node(s) being visualized in more detail. It is a proof-of-concepts, our interest is in both the visualization of the map and in the interaction with it. The visualization is then based on the notion of a focus node that allows the user to concentrate on a single node without loosing sight of the surrounding area. We also report on user studies that were performed to check the usability of the interface and to explore how the background of the users influences the understanding of the interface and browsing in it.

The remainder of this chapter is organized as follows: in Section 4.1 we give the requirements that a user interface for the LoLaLi map should fulfil. In Section 4.2 we introduce the issue of balancing global and local views for visualizing maps like ours. In Section 4.3 we describe the interface designed for the LoLaLi map and highlight its main features. In Section 4.4 we present the user studies we performed and discuss their results. Finally, in Section 4.5 we draw our conclusions.

4.1 Requirements for the User Interface

Our assumption is that our end users have a fair level of familiarity with computers and the Internet, and at least some background in the area covered by the map. The typical activity for such users is learning a subject, and in particular using reference tools, such as handbooks. When consulting a reference tool, users have to identify their information need and find out how to satisfy it, either by issuing a query, or searching for it in some other way. Then, they have to inspect and read the results of their search, and possibly confront them with other sources. Obviously, no editing action on the map is contemplated. The profile just outlined leads us to consider the following requirements:

Requirement A. Present each topic with its *context* in the graph.

Requirement B. Enable smooth browsing the map.

Requirement C. Enable intuitive search of the available resources.

Let us discuss these requirements in some more details.

Requirement A. Present each topic with its *context* in the graph. Graph structures are complex, and it is challenging to balance a broad, *global view* of their overall structure, and a *local view* that privileges the focus on single nodes. When inspecting a topic, the user should have a good view of the pieces of information attached to it, and at the same time, the user should not loose sight of the surrounding area, or at least of a relevant part of it. For each node in the graph, its direct ancestor and descendants, together with its siblings and the relation it has with them, are all relevant pieces of the context. Other fundamental pieces of information to show include the node's gloss and its links to the handbook. All this should be presented on the same screen, and the number of clicks required to inspect this information should be kept minimal.

Requirement B. Enable smooth browsing. A smooth browsing experience implies that the user always has a clear grasp of the location and the meaning of the information presented in order to identify the right region of the screen to look at and the links to click on. The complexity of the graph, and in particular, the presence of multiple parents, may hamper smooth browsing. Also, the depth of the graph and the possibly large number of subtopics available may be an issue.

Requirement C. Enable intuitive search of the available resources. The LoLaLi environment includes a map, the textual resource attached to it (i.e., the handbook) and the links that bridge the map and the handbook. Therefore, the search facilities should enable the user to search across each and all these components in a unified way. These search facilities should not require extra effort from the user in terms of clicking and exploration of the interface, and they should always be available together with the browsing facilities. A "best match" search (like most Internet search engines, and as opposed to an "exact match" database-like search) should be enabled as the default search that allows the user to type in a keyword or simply a phrase with no need to specify explicit constraints or filters on the search.

4.2 Related Work on Tree and Graph Visualization

The focused study on how to visualize graphs began with the seminal paper by Tutte [1963], who suggested an algorithm that places each vertex in the center of its neighbors. Now that structures like trees and graphs are increasingly used for the organization of data and knowledge, this issue has received great attention.[1] Much of the work, however, concentrates on presenting global views, as is shown by attempts to define when a graph is *understandable* and *nice* [Purchase, 1997, Rosenstiehl et al., 2001]. Common requirements to achieve that result include the following: edges and vertexes should not overlap; vertexes should be regularly distributed in the available space; if any symmetries of the data are present, they should be made visible. In the rest of this section, we survey approaches known in the literature and present them by adopting the distinction introduced in Requirement A above between global and local views.

4.2.1 Global views

Well-known global view oriented visualizations (for trees and/or for graphs) include hyperbolic trees, treemaps, cone trees and cluster maps.

Hyperbolic trees. Lamping et al. [1995] proposed a tree visualization method that allows focusing on one part of the data in the context of the whole unfocused data set. A hyperbolic tree displays a hierarchy of any size within a finite circular area so that

[1] International conferences are devoted to this issue, such as InfoVis (standing for *Info*rmation *Vis*ualization), which started in 1995 and especially deals with the visualization of hierarchies and databases, diagrams, and information spaces. In 2003 the conference became a contest, providing an evaluation benchmark for information visualization techniques and systems. Even more focused is the series of Symposia on Graph Drawing, held regularly since 1992.

the node that is being focused on is centered and nodes away from the central node are made exponentially smaller. An example of visualization based on hyperbolic trees is the Visual Thesaurus [Plumb Design, 2004] interface, where nodes are displayed around a circle (Figure 4.1). Hyperbolic trees allow for larger representations near

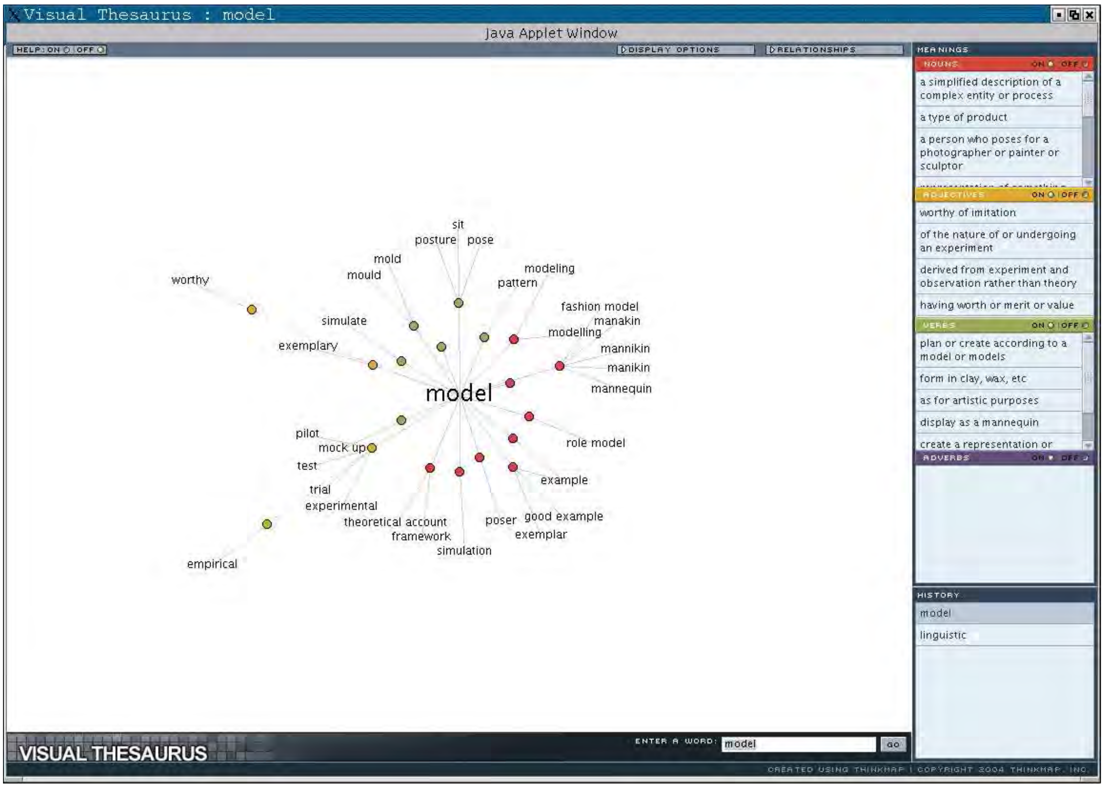


Figure 4.1: A screenshot from the Visual Thesaurus centered around the term ‘model.’

the focused area while still displaying the overall structure of the tree. Hyperbolic trees can also be animated and/or used to display diagrams in three dimensions, and can be modified to represent graphs. The problem with hyperbolic trees is that they do not scale well with the size of the tree, as the screen easily becomes too cluttered when increasing the number of nodes to accommodate. The dynamic layout of the nodes allows for a certain degree of flexibility in the amount of nodes to show, but the downside of this flexibility is that the visualization offers few points of (visual) reference, making it harder to discover paths in the structure. The fact that nodes in the tree move when clicking on them forces the user to get acquainted again with a new display. Finally, it may be hard to include more pieces of information about nodes than their very labels.

Treemaps. The treemap visualization proposed by [Bederson et al., 2002, Johnson and Shneiderman, 1991] displays hierarchies as series of embedded boxes. Each box

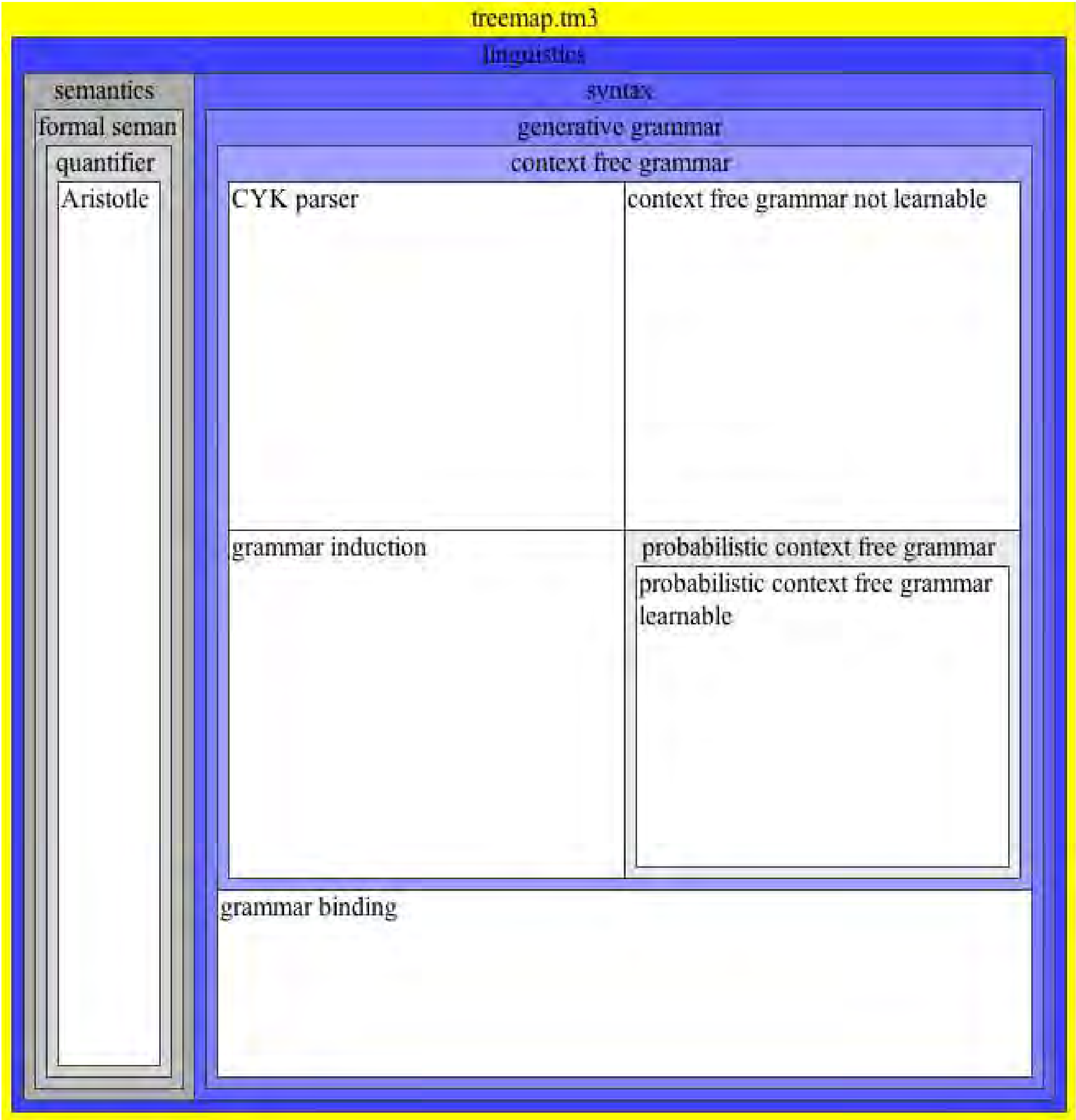


Figure 4.2: A treemap representation of some of the LoLaLi map data.

contains its children, usually by alternating at each level between laying them out vertically and laying them out horizontally (Figure 4.2). The relative size of each box can be made proportional to some property so that the parent's property is represented as the sum of the properties of the children. For instance, in a hierarchical file system, the size of a directory is represented by the sum of the sizes of all files (recursively, directories) in it. In this sense a treemap is a space-constrained tree visualization algorithm that turns a tree into a planar map. Treemaps are often used to visualize tree structures of directories and files, but they are also used for applications in domains as different as zoology taxonomy, bioinformatics, and finance [Baehrecke et al., 2004, Cable et al., 2004, Plaisant et al., 2003]. The treemap visualization method is good at showing high-level structure, and it is also more suitable than a hyperbolic tree for visualizing more details about nodes in the tree. Also, since nodes can be characterized by their size and color, it is easy to recognize patterns across the structure. The disadvantage of such a visualization is that it does not maintain consistent layout when navigating through different levels of the structure, and it does not easily represent multiple parenthood.

Cluster maps. Fluit et al. [2003] proposed a cluster-based visualization technique that aims at visualizing populated, lightweight ontologies.[2] A cluster map visualizes the subclass relations between classes and the instances of each class, together with a label stating the name of the class and the number of its instances. A cluster map is especially good at showing the level of overlap of instances between classes, and at giving an immediate visual feeling for the size of a class (Figure 4.3). The advantage of

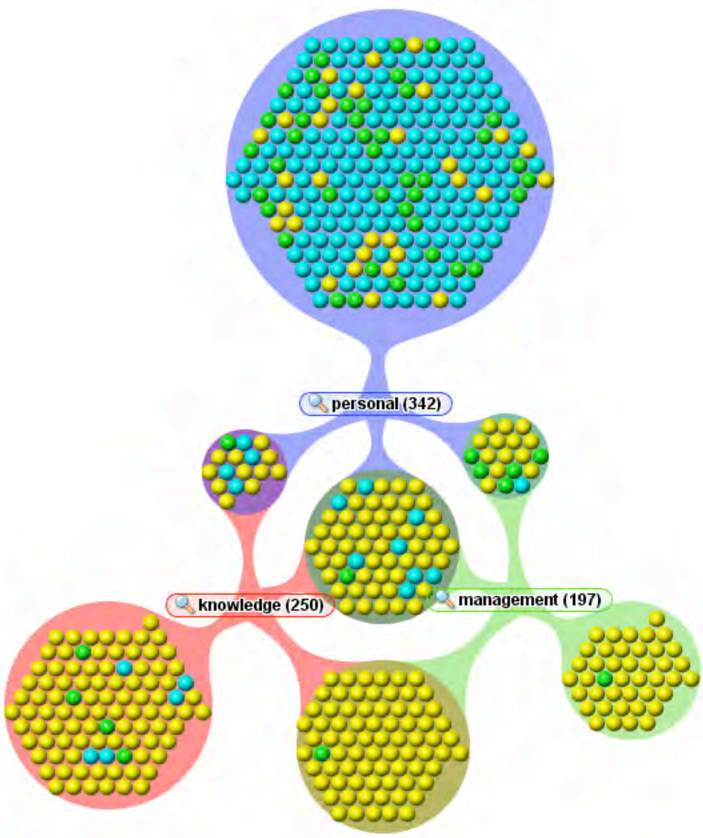


Figure 4.3: An example of cluster map visualization, taken from [ADUNA, 2005].

[2]The phrase "lightweight ontology" is used to refer to simple taxonomies with few axioms and only instances and definitions associated to each element of the taxonomy. Database schemas can be considered lightweight ontologies. Ontologies that have a richer structure or that are extensively axiomatized are referred to as heavyweight ontologies.

such a visualization is that it easily represents graph-like structures and is scalable to a large number of instances. Its major merit is, in fact, its ability to represent associations of classes and elements belonging to classes. The disadvantage is that it is not as scalable to a large number of classes as it is to large number of instances, and when the structure only includes classes this visualization becomes similar to a diagram-like visualization, like the one provided by Graphviz (see the figures in Chapter 3).

Cone trees. Robertson et al. [1991] visualize trees in 3D. In the representation they propose, the root of a tree is located at the tip of a transparent cone and the children of the root, and the subtrees stemming from them, are recursively arranged around the base of the cone. This allows a denser layout than traditional 2-dimensional diagrams, but nodes may be obscured, which forces the use of shadows or projections to make all nodes visible and give the perception of the whole shape. Cone trees are good at showing the global view of the structure, but the visualization becomes less clear when there are many nodes and many levels.

4.2.2 Local views

Visualizations that are local view oriented privilege a detailed view of single nodes of the structure. An example of that is the visualization used by many DOS and Unix-based operating systems to display hierarchical file-systems through command lines. Each level of the tree is represented by the list of its contents and the user navigates up and down the tree by means of typed commands. This way, the user can inspect one level at a time in great detail. This visualization method has the advantage of simplicity and clearness, since it provides a well focused view on each single level of the tree. Interfaces based on this approach can be defined as strongly local view oriented. They utilize a "stimulus reinforcing" approach: by repeatedly accessing directories and files, the user can reinforce and internalize her mental map of the structure. This process is facilitated by the stable schema of representation of the inspected level. Once the structure is known by heart, the user is able to quickly navigate through it. The downside is that the structure is implicit in the interface and active querying is required by the user to learn the structure and inspect its content.

Most current user interfaces for thesauri and ontologies display all pieces of information for a single node, plus links that point one level up or down. An example of this approach is the interface for the Medical Subject Headings (MeSH) [MESH, 2007] of the National Library of Medicine; Figure 4.4 shows a fragment of its structure as an indented tree. A directed graph can be presented as a tree by duplicating a child node as many times as the number of its distinct parents (i.e., tree branches are duplicated). In Figure 4.4 the duplicated child node is 'Myocardial Infarction.' The main advantage of this practice is that by displaying the graph as a tree, the complexity of the structure is simplified. The disadvantages are that much duplicated information is shown, and in order to identify and fully inspect nodes with multiple parents, the user has to compare all the branches shown. This operation can be facilitated by color-coding, like

Figure 4.4: A snapshot of the interface of the Medical Subject Headings (MeSH). The heading on focus is 'Myocardial Infarction.'

in the MeSH browser example (where the heading under inspection is marked by an arrow head to its left and printed in red), but this, however, does not fix the inherent weakness of this approach. To grasp the graph structure the user has to look at all the shown branches and compare them. While this might be a workable option with two branches, it quickly becomes more difficult when a node has more than two parents. In addition to these difficulties, this type of visualization also fundamentally misrepresents the graph because at any level it only shows one parent per node. In this way, only one of the paths connecting a node to a given ancestor is shown, while there could be an entire line of multiple parents along that path.

Summarizing, the main difficulties and limitations of the visualization approaches we presented concern the representation of multiple parenthood, since the presence of multiple parents may make the screen cluttered and scarcely readable. The solution of transforming the graph into a tree, by duplication of branches, does simplify the structure, but it also introduces much repetition and does not scale well. Interfaces based on a hypertree visualization are better solutions, but the fact that they dynamically lay nodes out requires more effort on the user side in order to place herself in the context of the map. Hypertrees are usually very pleasant to watch and interact with, but for the reasons given above we do not consider them to be a viable solution for visualizing resources to be consulted often or that carry the type of information carried by the LoLaLi map. Moreover, the fact that the tree moves and redisplays nodes when clicking on any of them can be distracting and consequently not apt for a resource such as the LoLaLi map.

4.3 A User Interface for the LoLaLi Map

The interface we designed in collaboration with the User Centered Design group at Elsevier Science[3] took the principle of information visualization called "focus + context" [Card et al., 1999] as its starting point. According to this principle, the most important data must occupy the focal point, in full size and detail, while the area around the focal point (the context) is displayed to help the user make sense of how the important information relates to the entire data structure. Often, the principle of "focus + context" is applied in such a way that regions far from the focal point are displayed smaller (as in a *fish-eye view*).

According to the "focus + context" principle, our interface divides the screen into a *map area*, the center of which is occupied by a *focus topic*. Parent and child topics of the focus topic are located respectively on the left and right side of it. Below the map area is the *data area*, where additional pieces of information, including glosses, and links to the handbook are shown (Figure 4.5). The search box and the button to start browsing from the top node are always available. The list of results is presented in the same layout as the rest of the interface (more details about the implementation of the search facility in [van Hage, 2004]). Then, the user is able to start exploring the surrounding area by clicking on topic titles or by rolling the mouse over topics in the map area.

[3] A special thank to Stephan Stipdonk for his invaluable contribution to the making of the interface.

The interface is implemented using Flash [Flash, 2005], which allows for animated movements that create a seamless browsing experience. The map is stored in RDFS format [RDFS, 2004] accessed by Sesame [SESAME, 2005], with a MySQL database [MySQL, 2007] as the back-end.

Figure 4.5: A snapshot of the LoLaLi interface.

The user can inspect the map in one of the following three ways:

1. by clicking on any of the topics (on its title) in the map area; in this case, the topic clicked on becomes the focus topic (Figure 4.6);

2. by rolling the mouse over a topic title in the map area; in this case, additional data about it is shown, keeping the focus topic fixed on the screen (Figure 4.7);

3. by using the search facility and then clicking on any of the returned nodes (Figure 4.8).

The action of clicking on one of the parents or children of the focus node makes the node occupy the focus area; then the map shifts "forward" (if one clicks on a child, cf. Figure 4.5 and Figure 4.6) or "backward" (if one clicks on a parent, cf. Figure 4.5 and Figure 4.7). The roll-over on a child topic makes its gloss appear in the data area, together with its relation with the focus node. In case it has other parent nodes beyond the focus node, those extra parents also appear in the focus area (Figure 4.7[4]). Symmetrically, when rolling the cursor over one of the parents of the focus node, all

[4]In the data area shown in the figure, modal logic appears labeled as 'subclass of logic' whereas in Chapter 3 we talked about a `subtopic` relation. The reason for this is given in Section 4.4.1, last paragraph.

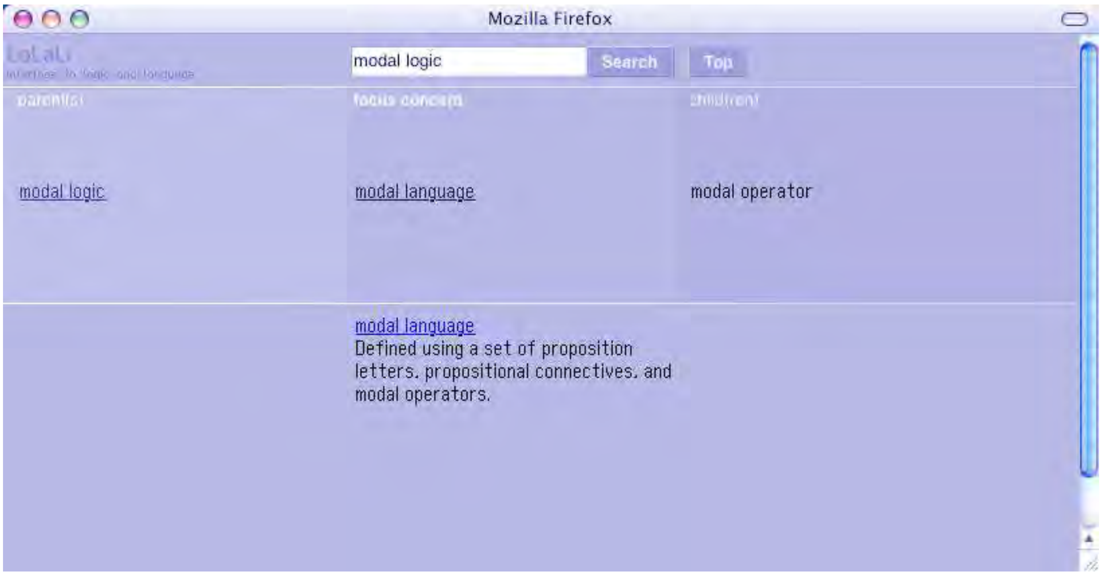


Figure 4.6: LoLaLi interface: after clicking on a child topic.

Figure 4.7: LoLaLi interface: rollover effect on modal logic.

Figure 4.8: LoLaLi interface: search results.

siblings of the topic that has the focus (with respect to the parent) appear in the focus area, while the gloss of the parent, and the relation of the focus to the parent appear in the data area. This rollover behavior is meant to allow users to quickly get information about a topic and asses its meaning and position in the map. The combination of displaying topics on rollover and the animated movement is meant to create a seamless browsing experience when moving from one node in the map to another. Summarizing, the visualization we provided has the following characteristics:

1. It makes the graph planar, i.e., without crossing of edges. In this way, it simplifies the complexity of the graph without misrepresenting it.

2. There is always a node in the focal position. In this way the structure is not presented as a whole in which users have to find their way (as in global views), but the user is always situated somewhere *inside* the map. The movement involved when changing focus concept creates the experience of navigating one coherent structure.

3. The interface limits the user's memory overload, as the user does not have to memorize the entire structure (as in the only-local views). Also, the overall layout of the map does not change during navigation, and the location of nodes relative to one another is fixed.

4. It is easily scalable, in terms of number of topics, relations, links and length of the glosses.

5. The interface maximizes the amount of information displayed for each node (as in local views), while showing essential information about the nodes in the surrounding area.

The LoLaLi interface does not exploit user familiarity with "collapsible" tree interfaces (such as the MeSH browser), but the motion effect created when clicking on concepts gives a feeling of continuity, as opposed to the feeling of "jumpiness" typical of hypertext documents. Because at any one time only a small part of the map is shown, it is crucial that moving in the map is as easy and natural as possible so the user does not feel inhibited. The seamless movement through the map also creates the sense of being in a larger coherent structure [Friedrich and Eades, 2002].

The roll-over effect is used to limit the amount of information displayed at one time in the screen, as sometimes less data has to be shown in order to fit on the screen without overloading the user [Tufte, 1988]. Also, the information shown on roll-over allows the user to decide which way to go before actually clicking. This mechanism gives users full control of the information flow, and they can absorb information at their own pace.

4.4 User Studies

Given our interests in end users and our aim of supporting their information seeking process, we ran a user study to assess the usability of the interface described above. More specifically, we wanted to identify our design's most important usability problems [Wilson, 2000]. Usability is the quality concerning how easy user interfaces are to use; usability tests are primarily, if not uniquely, concerned with 'effectiveness,' and so they imply a quantitative evaluation. For example, if the usability test concerned an interface for an email program, a quantitative goal could be stated as follows: "After inspecting the interface for five minutes, the user is able to send a message to a group of people in the address book in less than two minutes and making less than three errors." User studies are necessarily more qualitative, although they do take into consideration the quantitative data collected during the studies.

The user studies we conducted were aimed at answering a number of questions:

1. As for the usability of the interface, we were interested in knowing what, if any, are the main problems with the interface as a tool for navigating the LoLaLi map. And, can users find their way around the map using the interface described in the previous section?

2. As a secondary goal, we aimed at testing if and how the usability of the interface changed depending on the level of education of users with comparable background and on the level of familiarity with computers and the Internet.

To collect information, we designed a questionnaire. The questionnaire was not meant to be filled in by the users. Instead, two interviewers asked the questions directly to one participant at a time and the questionnaire served as a reminder for the question to ask. During each interview one interviewer was active, asking questions, explaining the tasks for the user to perform and taking notes, while actions on the screen were recorded by suitable software. The second interviewer was always present, to take notes and occasionally ask additional questions.

The questionnaire included closed (i.e., evaluative) and open (i.e., asking users to freely provide comments) questions. The evaluative questions were formulated according to a classic Likert-style format [Likert, 1932], and formulated in a way so as to avoid as much as possible the risk of influencing the users. As for the answers to the questions, we report all answers and comments provided by the users and wherever useful, we compile summary tables with all figures divided by groups (see next section) and the totals.

The interviews were conducted in Dutch with the Dutch speakers, in English with the others. As a sanity check for the questionnaire, before proceeding with the real user studies, we administered the questionnaire to two PhD students in the area of logic and physics respectively.

4.4.1 The Setting

Below we explain and motivate the decisions taken for the organization of the user studies, corresponding to the following issues:

- What type of interface prototype to use?
- Who should participate in the tests?
- How many participants to involve?
- What kind of questions and/or tasks should the users answer and/or perform?

What type of interface prototype to use? The prototype to use for the user studies could be either a paper mock-up, or an interactive prototype. We opted for the latter type, because of the intrinsically dynamic nature of the interface we developed, and because an interactive prototype seemed better at uncovering problems than a paper prototype. As for the features that the prototype should include, Nielsen [1989] distinguishes three types of prototypes: horizontal, vertical and scenario. A *horizontal* prototype only contains a shallow layer of the surface of the user interface (for example, a menu option, with no operation associated), a *vertical* prototype fully implements only a small number of features of the interface, while a *scenario* prototype fully implements only the operations necessary to perform a certain task. We decided to test a vertical prototype, which would enable us to see how users would use the interface to browse and search the LoLaLi map, and, partially, to inspect links to the handbook. Such a prototype should also give insight into currently missing features of the interface.

Who should participate in the tests? As for the users to invite to the study, it was natural to address an audience of university students (cf. Section 4.3). We contacted students in an area related to that covered by the map, but distinguishing two levels of educational profile, namely masters and undergraduate students.

Note that if we had been interested only in assessing the soundness of the interface, we could have performed a "heuristic evaluation," consisting of an evaluation done by a pool of human computer interaction and design experts. There is evidence suggesting that if the pool only includes experts, close to 100% of all interface problems are found [Jeffries et al., 1991]. Instead, we preferred to have a sample of the type of people that would actually use the interface so as to be able to have insight into their view on it. We did not expect to find all problems of the interface, but only the major ones, arguably the best strategy for this type of problem [Bailey et al., 1992].

How many participants to involve? Nielsen and Landauer [1993], Virzi [1992] and Lewis [1994] published influential papers on the topic of sample size in usability testing. The authors presented a mathematical model for determining the sample size for usability tests, with empirical evidence for the models and several important claims: 80% of the usability problems are detected with four or five participants, while the proportion rises to 90% if there are ten participants. The "small sample" claims and their impact on usability methodology have been popularized in Nielsen [2000]'s widely read "`useit.com`" online column. Recently, it has been argued that a sample size larger than five may be required to detect a satisfactory number of usability problems. Taking into account these considerations, we aimed at two groups with six to ten participants.

What kind of questions and/or tasks should the users answer and/or perform? The LoLaLi environment (map and handbook) is to be used as a reference tool to support our users, and ideally our user studies should include these types of real life tasks. Then, the tasks our intended users are usually called to perform include the following: "Write an essay on topic Y," "Read about topic X and answer the following questions," and "Discuss similarities and differences between topic Z and W." Unfortunately, this type of end-to-end tasks does not fit the context of our work: it would be problematic to assess, and it would hide the operations of searching and browsing the map—the very basic operations that the LoLaLi environment should support. Therefore, we opted for subtasks of a real life task, consisting in searching about a topic ("what is ...?"), both at a shallow level, that could be answered by the map, and a deeper level, answerable by following links to the handbook. Also, users could be interested in finding out connections between important topics in the area. Then, the tasks we included in the questionnaire are of the type: "Browse/search for topic modal logic and do the following/answer the following question." The drawback of this type of tasks is that it does not encourage users to pursue their own curiosities and information needs.

The user studies involved 14 people: 6 masters students (5 students in the Master of

	UG		*MSc*	
	M	*F*	*M*	*F*
English	-	-	2	1
Dutch	7	1	1	1
Other	-	-	1	-

Table 4.1: Composition of the groups of undergraduate students (UG) and masters students (MSc) on the basis of gender (M/F) and native language.

Logic program, MoL, at the University of Amsterdam, one Dutch student in Humanities Computing), and 8 undergraduate students (8 first year students enrolled in an Artificial Intelligence course, one of which was a student in Philosophy and the others in Artificial Intelligence). Table 4.1 summarizes the composition of the participants on the basis of their university status, gender and native language. In all tables in this chapter we use the short forms MSc for masters students and UG for undergraduate students.

At the time we conducted the user studies, the content of the LoLaLi map was being converted into RDFS (cf. Section 3.7.2). All hierarchical relations were initially rendered as a property called 'unspecified subtopic' that was incrementally specified over the course of the project. At the time of the user studies, the relations used were: subclass, part-of, instance, mathematical result, computational tool, notion, historical view. Theoretical considerations discussed in Section 3.4.1 (concerning the many flavors of the is-a relations), and the observations made in the course of the user studies (concerning the relations unspecified subtopic and notions) make us select the relations described in Chapter 3.

4.4.2 The Questionnaire

After we analyzed the goals of our user studies (along the lines described above), we singled out the following questions to which we wanted our user study to provide answers.

1. On finding information:
    (a) how do the users perceive the browsing with the interface?
    (b) what kind of difficulties do the users encounter?
    (c) is searching difficult?
    (d) do the users have a preference for search over browsing?
    (e) how do the users perceive the links to external resources?
    (f) what expectations do user have about them?
2. On the map structure:

   (a) is the structure of the LoLaLi map understandable in its main features?

   (b) what is the value of relations in terms of understandability of the structure?

3. On some features to be considered for later inclusion:

   (a) would users find back button, bookmarks, and grouping of subtopics useful?

We phrased our questions in as neutral a way as possible, so as to avoid pushing users towards "socially desirable" answers. At the very beginning of the questionnaire, the users were asked some questions about their computer literacy and information gathering strategies. The complete questionnaire is included in Appendix A.

4.4.3 Results

We now report on the results of the user studies following, section by section, the questionnaire administered to the participants. The sections covered are:

- Computer literacy
- Information gathering strategies
- Browsing
- Searching
- Features of the LoLaLi map
- Links to the handbook
- User preferences
- User wishes

Computer Literacy

Both groups (undergraduate students and master students) present a high level of computer literacy (Table 4.2). Even though 4 undergraduate students declared they had no formal computer training at all, this does not correspond to a lack of education, as is suggested by the high number of programmers among them. The two groups are also comparable in terms of years of computer usage (in each group, 4 people have 10 years or more experience), as well as in terms of frequency of Internet usage (almost all on a daily basis). The proportion is slightly different if we look at the years of Internet usage: among the undergraduate students, 6 people have 4–6 years experience, one person less than one year, and one more person between 1 and 3 years. Among master students, 2 people have 7–9 years experience, while 4 have 4–6 years. As for the top three activities with computer, master students mainly write documents and papers, use email and surf the web. Undergraduate students program, surf the web, and use email; less frequent activities are video editing, graphics and computer games.

Information Gathering Strategies

The way people from the two groups approach new fields is comparable (Table 4.3). Undergraduate students tend to go for a combination of web search engines (7 people)

Questions	*Answers*	*UG*	*MSc*	*Total*
Which operating system do you use more often?	Windows	5	2	7
	Linux	2	1	3
	Macintosh	-	1	1
	Windows *and* Linux	1	2	3
What formal computer training do you have?	none	4	-	4
	college courses	2	-	2
	undergrad. courses in computer science	1	1	2
	minor in computer science	-	3	3
	major in computer science	-	2	2
How many years of computer usage?	4–6	1	1	2
	7–9	3	1	4
	10	4	4	8
How many years of Internet usage?	1	1	-	1
	1–3	1	-	1
	4–6	6	4	10
	7–9	-	2	2
Frequency of Internet usage	weekly	1	-	1
	daily	7	6	13
Which are the most frequent activities on screen?	writing papers	3	6	9
	e-mailing	5	5	10
	surfing Web	6	5	11
	programming	7	2	9
	playing games	1	-	1
	graphic, video	2	-	2

Table 4.2: Questions on computer literacy of the participants in the study. Multiple answers were allowed.

and other resources, i.e., people, textbooks, and libraries. Master students use web search engines, ask people, and consult textbooks.

Glossaries are never used by undergraduate students, and by only one master students, yearly. Dictionaries are slightly more frequently used: by undergraduate students monthly (2), weekly (4), daily (1), by master student, monthly (2), and weekly (2). For obvious reasons (cf. Table 4.1), many master students consult bi-lingual dictionaries. As for encyclopedias, only one master student and 2 undergraduate students declare to use them (daily, and monthly and daily respectively). These resources tend to be consulted online, and the three people consulting them on CD-ROM are all undergraduate students.

Only three master students take notes, either on file (1) or on paper (2), while undergraduate students do it on paper (3), on file (3), or depending on the moment (2). Both groups use paper textbooks, but also consult electronic texts: master students in

Questions	*Answers*		*UG*	*MSc*	*Total*
What is your preferred way to approach a new field?	web search engine		7	4	11
	people		6	4	10
	textbooks		3	3	6
	libraries		2	-	2
Do you use reference tools?	glossaries	never	8	7	15
		yearly	-	1	1
	dictionaries	monthly	2	2	4
		weekly	4	2	6
		daily	1	-	1
	encyclopedias	monthly	1	-	1
		daily	1	1	2
In what electronic format?	online		5	5	10
	CD-ROM		3	-	3
How do you take notes?	on paper		3	2	5
	on file		3	1	4
	it depends: paper, file		2	-	2
What digital resources do you use more often?	e-assignments		1	1	2
	handouts, slides		3	4	7
	journals, papers		1	5	6
	web sites		1	1	2
	depends on web search engine		2	-	2
	other		3	-	3

Table 4.3: Summary of usage of electronic resources.

particular for assignments, handouts and journal articles; undergraduate students also use message boards and web sites. Interestingly, but not surprisingly, two undergraduate students claim that the kind of electronic document they consult depends on what the web search engine they use returns.

Browsing

The aim of the questionnaire's section on browsing is to see how the users orient themselves when browsing the interface, what problems they encounter, and whether they like the experience. In this part of the questionnaire, users are asked to reach a given topic by browsing (hints are given), and comment on the experience. An example of a browsing task is the following: "Browse until you get to the topic recursion theory, knowing that it is a topic in the area of theory of computation." Users were given 5 such questions.

Both groups perform the browsing tasks with no major problems. All of them, especially the undergraduate students, try first to retrieve the topics by searching (this is very evident for the first question), but once better instructed they performed the

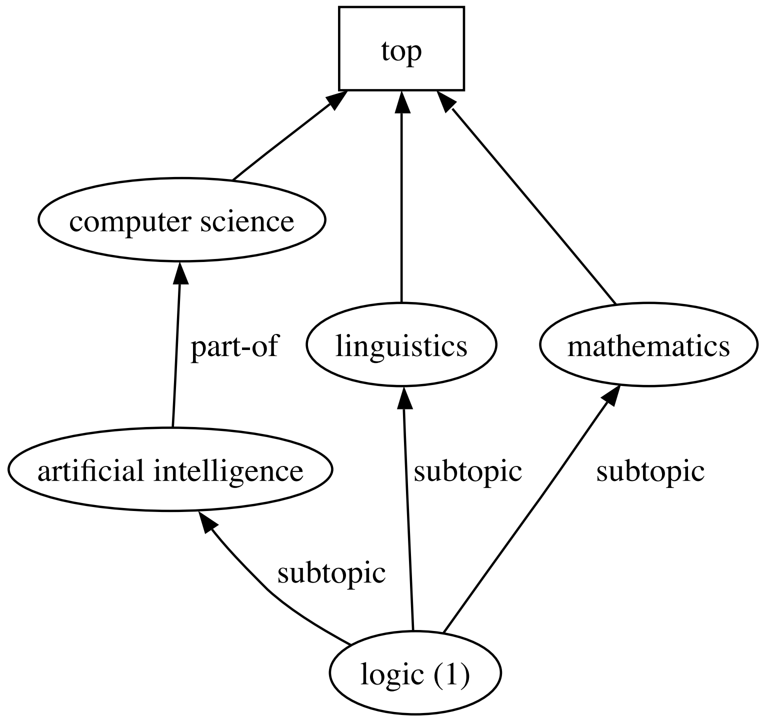


Figure 4.9: Logic (1) is a subtopic of artificial intelligence, computer science and linguistics.

browsing tasks smoothly.

A remarkable difficulty experienced by the users concerned the position of the topic logic (1). All topics indicated by the questions belong to the branch of logic (1), which is an unspecified subtopic of mathematics, computer science, and linguistics (Figure 4.9). This implies that users had to chose which of those topics to inspect in order to find logic (1). Only one master student stated openly the problem (claiming that "logic is related to all of them, I could not decide where to go"), though all participants appear to have some moments of uncertainty before deciding where to look for logic (1). Many of them inspected one of the topics mathematics, computer science, linguistics, with an evident correlation between their choice and their background, and went back to it every time they needed to consult again the same topic.

Then, after performing the browsing tasks, users were asked to comment on the experience and mention at least three problems they encountered. We can divide the reported problems into two classes: problems concerning the structure/content of the map, and problems concerning the interface.

With regards to problems related to structure and content, master students mentioned the issue of the position of logic (1) sketched above (1 person); the fact that they had to think of the best strategy to apply, i.e., whether to start again from the top node or to click back until the right path is found (1); and that browsing is easier when topics are already known, though "you get used to it fast" (1). Among the undergraduate students, one person admitted to having problems understanding the organization of the map, two people commented that browsing is easier when you have some prior knowledge of the area. As for problems concerning the interface, master students reported problems with scrolling the lists of topics, namely that they could not scroll with the rolling wheel (2 people), problems with the particular mouse used during the interview (2), and problems with the data appearing for the focus topic when the mouse is on a parent topic (1). Undergraduate students did not report problems with the interface, but three of them pointed out that they usually only search.

When asked to give a general comment on the interface, both groups were positive. Master students unanimously said the interface helped them to make a mental picture of the structure of the map and gave a generally positive comment on the browsing experience (5 people said "I like it very much," one person "rather so"). Nevertheless, one person found it annoying that terms disappear when rolling the mouse, and that more options should be in the center associated to the focus node. Undergraduate students had problems with the mouse used during the interview (did not roll smoothly enough), and complained about the focus topics being underlined, which they took for the indication of the presence of a link (see Figure 4.5.)

Searching

The users participating in the study were given a number of tasks in which they had to search for a given topic, then perform an action or answer a question. The aim of this section is to test the usability of the search facilities provided, and the user's comprehension of the information shown in the interface, such as the typed relations and the glosses.

1. Search for the topic operator and divide its subtopics into groups. The topic operator is presented with the following subtopics: quantifier, arity, modal operator, truth functional operator. Since both typed relations and glosses are shown in the interface, we expected that the users would group together modal operator and truth functional operator, and leave arity apart.

Searching did not present any difficulties for any of the users. As for the grouping, undergraduate students gave a variety of answers, from "I cannot make any distinction" (1), to "I can distinguish different types of subclasses." One person had the intuition to group together topics that are "at the end of the tree," thus making an intuitive distinction between topics with subtopics and topics without subtopics. Two people admitted that the differences among the subtopics were unclear to them. Among the master students, 2 people found the task clear and grouped the subtopics of operator as expected, 2 people said that only arity was clearly different from the others, and 2 people could not make any grouping.

After a few interviews we realized that the types of the relations were not clearly visible in the interface, which explained the difficulties of the participants in understanding the question. After realizing this, the interviewer pointed the relations out to the users, who were then able to suggest sensible groupings among the subtopics. The improvement was equal for both groups. Unfortunately, a few people were interviewed before the source of difficulties was realized, and this is likely to have introduced noise in the study.

2. Find the topic logic (1) and distinguish different types of its subtopics. At the time when the test was performed, the topic logic (1) had some 30 subtopics, as shown in the fragment below, where `SC` stands for the relation `subclass`, `PO` stands for the relation `part-of`, `MR` stands for the relation `mathematical result`:

logic (1)
- `SC` temporal logic
- `SC` modal logic
- `SC` first order logic
- `PO` recursive function theory
- `MR` Löwenheim-Skolem-Tarski theorem
- `MR` Gödel's 1st incompleteness theorem (1931)
- . . . . . .

We expected that the users would use the labels for the typed relations shown in the interface to group together topics, but people encountered the same difficulties discussed about the previous task, related to the visibility of the relations in the interface.

3. Explain the relations between logic (1), proof theory, cut elimination theorem. For this task, users had to retrieve the topics proof theory, cut elimination and logic (1). and explain the relations between them. The users were also asked to indicate what features of the interface had given them the hint. In the LoLaLi map they were presented the three topics are related in the following way: cut elimination theorem is a `mathematical result` of proof theory, which is `part-of` logic (1) (Figure 4.10). Master students appeared rather comfortable with the sequence logic (1), proof theory,

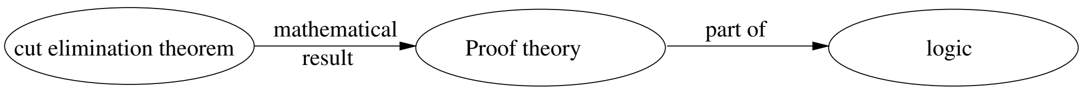


Figure 4.10: Topics to retrieve for the task on searching: explain the relations among proof theory, logic (1), cut elimination theorem.

cut elimination theorem, and 5 of them gave the expected answer. Only one person reported problems with the relation `mathematical result`, claiming that the relation `part-of` is clearer than the other. In general, the relation `part-of` appeared not problematic, though they might call it "branch of" or "subset of." When asked, all of them had a clear picture of the position of those topics in the graph (gained during previous browsing in the area). Undergraduate students also showed a good intuition of the relations among the presented topics: 5 people gave the expected answer, 2 people commented that the relation `part-of` is clearer than `mathematical result`, one person said the opposite.

4. Explain why truth function has two parents. The topic truth function has two parents because it is a subtopic of truth and semantics. Users had to explain why, according to them, that happened. All users referred to the fact that "truth function had something to do with both," and many were able to express their understanding of the relations among the parents and the child topic.

Features of the LoLaLi map

This section of the questionnaire aims at testing what difficulties users encounter with some features of the map. In particular, the section concentrates on relation types, on multiple parenthood, and on how these features are rendered in the interface.

1. Comments on the typed links. First of all, the users were asked to give a general evaluation of the typed links as seen so far. The meaning of the typed relations were generally clear to master students. Typed relations were clear to 4 people (2 said "very much clear," 2 said "more or less clear"), while only one person said the relations are "hardly" understandable. Two people noticed they had some difficulties with the relation `unspecified subtopic` and with the relation `notion`. One of the undergraduate students rated the relations as "very much clear," two as "more or less clear", three as "rather clear". One person admits to understanding the general idea of typed relations, but to having some difficulties with specific relations. In particular, difficulties occurred with the relation `notion` and with the relation `instance`, that seems to be understood as something in between `part-of` and `subclass`.

2. Assign a typed relation to pairs of topics. The users were asked to assign a typed relation to the following pairs of topics:

- modal logic – logic (1)
- theory of computation – computer science
- Gödel theorem – logic (1)
- operator – logic (1)
- Frege on quantification – quantification
- formal semantics – semantics

In Table 4.4 we report the entire list of results: the pairs proposed to the users are listed in the first column, the labels chosen by the users are reported in the second column (a • marks the label adopted in the LoLaLi map), the following two columns show the answers given by the undergraduate students and the master students, respectively, while the last column reports the total number of answers given. One master student consistently did not provide answers to any of the questions.

Question	*Answers*	*UG*	*MSc*	*Total*
modal logic - logic (1)	SC •	6	3	9
	SC or IN	-	1	1
	SC or PO	-	1	1
	PO	2	-	2
	no answer	-	1	1
theory of computation - computer science	PO •	1	1	2
	subject area	-	1	1
	PO or SC	-	1	1
	MR	4	-	4
	Notion	1	-	1
	SC	1	-	1
	unsp. subtopic	1	2	3
	no answer	-	1	1
Gödel theorem - logic (1)	MR •	4	4	8
	unsp. subtopic	-	1	1
	HV	2	-	2
	PO	1	-	1
	no answer	1	1	2
operator - logic (1)	Notion•	2	4	6
	SC	1	-	1
	unsp. subtopic	1	-	1
	PO	1	-	1
	used in logic	1	-	1
	MR	1	-	1
	IN	1	-	1
	no answer	-	2	2
Frege on quantification - quantification	HV •	3	1	4
	IN	-	1	1
	unsp. subtopic	1	2	3
	Notion	2	-	2
	SC	5	2	7
	PO	1	-	1
	no answer	-	2	2
formal semantics - semantics	SC •	1	1	2

Continued on next page...

. . . continued from previous page:

Questions	*Answers*	*UG*	*MS*	*Total*
	unsp. subtopic	1	1	2
	PO	2	-	2
	no answer	-	2	2

Table 4.4: Summary of results for the question: "Assign a typed relation to pairs of topics." More than one answer was possible.

Let us take a look at each question individually. The pair modal logic – logic (1) is well understood, as 9 out of 13 people indicate `subclass` (to be read as: modal logic is a `subclass of` logic (1), and two more were undecided between `subtopic` and `IN` and `PO`; the same reading applies also in the following), while 4 people give other answers.

The pair theory of computation – computer science appears to be more difficult. Only one master student and one undergraduate student answered that theory of computation is a `part-of` computer science, while the remaining 14 people gave 7 different answers. Masters students mentioned: "subject area" (1), `unspecified subtopic` (2), `part-of` or `subclass` (1); the answers of the undergraduate students include `unspecified subtopic` (1), `notion` (1) and `subclass` (1).

The pair Gödel's theorem – logic (1) is also well understood, since 8 people labeled Gödel's theorem as a `mathematical result` in logic (1). Interestingly, 2 undergraduate students labeled it as `historical view`, probably because the name of the topic includes the name of the author, which gives a sort of historical flavor to it. This fact raises the issue of what is *historical* in our work, which is of course related to the definition of historical in science, and to the kind of study the user we address is interested in.

For the pair operator – logic (1) we can observe a clear difference between master students and undergraduate students, as four master students opted for `notion` and two had no idea, while almost each undergraduate student proposed a different label.

The pair Frege on quantification – quantification is an interesting one, as for ten people (6 undergraduate students, 4 master students) is a either an unspecified subtopic or a subclass and only four people indicated `historical view`.

Finally, the answers for the pair formal semantics – semantics are like those for the pair modal logic – logic (1). a common confusion of `subclass` and `part-of`, among both master students and undergraduate students.

3. Explain why modal logic has three parents. Using the search functionality, the users had to retrieve the topic modal logic and explain why it is presented with three parents. Master students were all comfortable with the multiple parenthood: only 1 person said it is "rather clear," while 4 people said it is "clear" or "very much clear." All undergraduate students affirmed it was "very much clear," or "rather so." As an explanation of the multiple parenthood, people from both groups mention the fact that modal logic has three parents because it has "something to do" with all three parent topics.

4. Describe what happens when you roll over semantic ambiguity in child position. To answer this question, the users had to find the topic semantic ambiguity, make it occupy the child area and roll the cursor over it. When doing so, both parents of semantic ambiguity appear in the focus area (where usually there is only one topic at a time). This question was meant to see how visible and understandable the mechanism of roll over was to the users. Among undergraduate students, two of them said they would not have noticed that if the interviewer had not pointed it out to them. For three people it was "very clear." For all master students what happens was "clear," but 3 of them were annoyed by the fact that data disappears when moving the cursor away.

Links to the Handbook

The prototype includes some links to the handbook, accessible from the data area below the focus topic. Links to the handbook are marked by the title of the chapter from which they are taken, together with the available information about the position of the link target in the chapter. Our subjects were asked to inspect one of these links (suggested by the interviewer), state their expectations about the text they would find and comment on the differences between their expectation and what they actually found.

Master students gave articulate descriptions of their expectations, saying they would expect to find different definitions of the focus topic, related topics, topics "using" or "used by" the focus topic and other information about it. They also mentioned they would like to have a clearer description of what they would find when following the link, and that the source of the text should be highlighted so as to be aware of what text they were exploring. They also expected a shorter text than the one they were shown, and two people stressed the importance of using the same layout for the visualization of the text and of the map. Undergraduate students stressed their preference for a shorter text, well described before clicking on the link, and with a layout consistent with the starting page. Both groups agreed on the usefulness of having both glosses and links to larger pieces of documents than the gloss, and motivate this by saying that a gloss is useful to decide whether it is worth inspecting the link.

User Preferences

This groups of questions is meant to discover the users' opinion about search and browsing facilities. Table 4.5 summarizes the answers given.

1. Did you like browsing the map? The users were asked a similar question at the beginning of the test, during the session dedicated to browsing. Now they have performed tasks involving both browsing and searching, therefore they can compare

Questions	*Answers*		*UG*	*MSc*	*Total*
Did you like browsing?	very much		6	3	9
	yes, but	miss back button	1	-	1
		no keyboard shortcut	-	1	1
		too much to scroll	-	-	-
	rather so		-	2	2
	hardly		-	1	1
Did you like searching?	very much		7	5	12
	more or less		-	1	1
	rather so		1	-	1
Preference for	searching		4	1	5
	browsing		-	1	1
	combination		3	4	7

Table 4.5: Summary of the questions about searching and browsing.

the two experiences and comment on that. Undergraduate students were generally positive: the majority said they like browsing “very much” (6), “rather so” (1) and one person said “Yes, but miss the back button.” Among master students we find slightly more variability, ranging from “very much” (3), “rather so” (2), “hardly” (1); finally one person remarks that he appreciates the organization with relationships, but would like to have more commands from the keyboard, and that there are too many children to scroll.

2. Did you like searching the map? Master students were very positive about searching: only one person said he liked the experience “more or less,” the remaining 5 people said they like it “very much,” and one among them stressed the fact that “search is essential.” Undergraduate students were even more enthusiastic about searching: 7 of them said they liked it “very much,” only one person answered “rather so.”

3. Preference for searching over browsing, or vice versa? Among the undergraduate students, 4 people expressed a strong preference for search over browsing, and one of them remarked that searching is always the first thing, then browsing follows. Three people preferred a combination of search and browsing. Master students mentioned: a preference for a mixture of search and browsing (3), a strong preference for browsing (1), a preference for searching, or a combination of searching and browsing (1), a preference for searching followed by browsing (1). One person did not express a preference, but underlines that it depends on the context and the aim.

User Wishes

Finally, we asked users to express their wishes about features not currently included in the interface or in the organization of the map.

1. Would you like to see a bookmark functionality? The majority of the undergraduate students answered positively (7). Five of them also stressed the fact that it would be useful for larger maps than the one shown to them, one person observed that it would be useful for off-line use. Only one person said it was not necessary. Also among master students we have a majority of positive opinions about bookmarks, though more articulated: “Yes, but it depends on the number of topics in the map” (1), “yes” (2), “Yes, but I would not use it” (1), “I do not know” (1), “no” (1).

2. Would you like back buttons? The question was about a back button internal to the navigation system, since the back button of the browser brings the user outside of the LoLaLi map site. All master students thought that it would be a useful option, in particular 4 people mentioned that they would like to have the history for both browsing and searching. Undergraduate students were slightly less enthusiastic about it: 3 said they would not like having the back button, 3 people said they would, two people wished to have a search history, one person noted that after understanding the interface, the back button is not essential anymore (because already clicked topics are distinguished by the others by means of an underline). One person spent some time pondering about what exactly a back button would mean in the context of a graph structure.

3. Would you like to group children, for example according to the relation type? In general, the users seemed to be comfortable with this idea. Among the undergraduate students, there were 4 positive answers, one person preferred the alphabetical list, another person preferred a complete view of all children, finally one person desired a system to group children, but also wanted the possibility of viewing the entire list or having an alphabetical listing. Among master students, 3 people said they would like very much to have the grouping, 3 people said they would like to have it (one person notes that sometimes it could be good to look only at subclasses), but one of them observed that relations should be better explained. Finally, one person observed that the alphabetical list is fine if you know the area, otherwise it is better to get the complete list of all children.

4. Would you like more sophisticated search? When asking this question we left open what we meant by “sophisticated.” Users were explicitly told that it would not imply to learn a more complex query language, but that the sophistication would only involve the type of query they could answer. Master students answered yes, they would like to have a more sophisticated search system (3), “Maybe, but not immediately visible to the user” (2), “It depends on the use and size of the map” (1). Undergraduate students answered: “It depends on the size, if it grows, yes” (3), “Basic is good enough” (1), “It could be useful but I would not use it” (2), “Yes, but without having to learn an extra language” (1). Finally, one person simply answered no.

5. Comments, suggestions? Although the question was totally open, people tended to give more comments than suggestions. Undergraduate students were generally very positive: only one person commented that it took him awhile to realize it was a graph. Master students also gave positive comments; in particular one person claimed he would use the map provided with links for every day work. Two people noted that scrolling of the list of child topics (cf. Figure 4.7) is not handy.

4.4.4 Discussion

The preliminary survey on *computer literacy* and information seeking behavior confirmed that the two groups have comparable familiarity with computers, although master students have more experience with the Internet, on average. The two groups also have similar habits concerning the exploration of new fields: they both have a strong tendency to use web search engines, but master students also ask people more than the undergraduate students. Both groups make little or no use of glossaries or encyclopedias, but half of the undergraduate students use a dictionary weekly, while only one third of the master students does. If in electronic format, the dictionary is usually used online by both master and undergraduate students.

The *browsing* actions were performed with no major problems by both groups. One question required users to go to the topic logic (1), which is under computer science, linguistics, and mathematics. They tended to inspecte only one of those paths and went back to it every time they needed to consult the topic again. We noticed that those who had training in mathematics looked for logic (1) under mathematics, and those who had a training in computer science under computer science. This suggests the importance of the user background for the use of the interface, and the consequent need to provide for accurate explanation hints.

When asked to comment on the browsing experience, only one master student was able to verbalize his difficulty with the position of logic (1) sketched above (he had tried more than one path). Undergraduate students seemed to have noticed the same behavior, but could not verbalize the issue. Other comments concerned the fact that when asked to browse for a new topic one had to think of the best strategy to apply, i.e., whether to start again from the top node or to click back until the right path is found. Finally, both undergraduate students and master students noticed that browsing is easier when topics are already known, though "you get used to it fast."

When asked to give a general comment on the *interface*, both groups agreed that the interface helped them make a mental picture of the structure of the map and thereby of the underlying domain. They all gave positive comments on the browsing experience. Nevertheless, they stressed the annoyance caused by terms disappearing when rolling the mouse, and suggested that more options associated with the focus node should always be visible in the center of the display. It turns out that the motion effect when a topic shifts to the focus area is well understood, while the rollover effect seems to be more controversial. One type of rollover, over child topics (cf. Figure 4.7), seems to be easy to grasp (only two people said they did not notice the appearance of the parent topics in the focus area). In this case, the user is shown the gloss of the topic over which she rolled the mouse, its relation type with the focus topic, and other parents of the child, if any (they will appear in the focus area). More problematic seems to be the rollover on parent topics (more child topics appear in the focus area), for which unfortunately, there was no dedicated question in the questionnaire.

The interface failed to make immediately clear the types of the relationships, as we noticed already during the interviews. This might be one of the reasons why, when we asked users whether they found the hierarchical relations clear, it appeared that even though the *typed relations* were clear or mostly clear to all participants in the tests, some confusions remained. The question on attributing a relation to pairs of topics reveals that the relations of `subtopic` and `part-of` are intuitively understood, but they can be confused with one another. This observation confirms [Storey, 1993], who suggests that an accurate understanding of even the most intuitive relations cannot be taken for granted. The domain-specific relations of `historical view` and `Notion` were especially difficult for undergraduate students but for different reasons. On the one hand students had different ideas about what "historical" is, depending on whether they consider historical something that was developed in the past and no is longer used, or something that was developed in the past and is still currently in use. On the other hand, the label "notion" seemed too general to many users. These observations made us to consider the appropriateness of the names used for these relations, and in particular their aptness to the background of the users. Finally, the relation `unspecified subtopic`, although it appeared in very few questions, seemed to generate some confusion.

The undergraduate students enjoyed the browsing experience (possibly more than master students) and search was equally appreciated by master students and undergraduate students. When asked to express a preference between searching and browsing, the users, especially undergraduate students, claim to prefer searching, and especially master students consider the usefulness of browsing and searching (or in combination, usually search first, then browse) depending on the different tasks they have to perform.

As for the features not currently included in the interface, a history button would be appreciated by all, while a bookmarking functionality is much wanted especially by undergraduate students (master students would appreciate it, but say they can also do without). We obtained more articulated answers when asking about the possibility of grouping children (for example according to relation types). In principle it would be appreciated, and our test subjects mentioned different views such as alphabetical order, ordered by relations, and so on. The interviewees also pointed out that the grouping views should not be exclusive, so that the user could go from one view to the other. Master students are in favor of a grouping based on relations. Finally, more sophisticated search functionality does not seem to be a priority for any of our users. They also said that in any case the first visible option must be as simple as possible.

4.5 Conclusions

In this chapter we have explored the issue of providing an interface for the end user of the LoLaLi map. We identified our typical end user as a university student in areas related to the one treated by the *Handbook*. We defined a set of requirements for such an interface, taking into account the operations that the user should be allowed to perform and the importance of obtaining the right balance between a local and global view. The graphical user interface described in Section 4.3 and developed in collaboration with the User Centered Design group at Elsevier Science respects those principles. The interface is organized into a map area, presenting a fragment of the map (i.e., a focus topic that occupies the center of the screen, plus parent and child topics to its left and its right, respectively), and a data area where data about topics in that fragment is shown. The interface is dynamic so as to allow smooth navigation from topic to topic. By showing only a relatively small fragment of the map at once, it makes the graph planar, because it hides the complexity of the graph without changing its structure. Also, by showing a small focus area, it is possible to show the hierarchical nature of the data in a consistent manner. The complexity is put back into the graph by means of the rollover behavior of the cursor. The possibility of showing larger portions of the data in one screen is sacrificed in favor of a display that does not force the user to re-scan the entire screen (as opposed to interfaces based on hyperbolic trees), and this allows easy inspection of the pieces of information associated to the topics.

In this chapter we also reported on the user study we performed (Section 4.4). The high level aim of the user study was to identify the most important usability problems with the proposed interface. The two specific questions we aimed to address concerned:

1. the usability of the interface (what, if any, are the main problems with the interface as a tool for navigating the LoLaLi map?), and

2. the influence of the background of the user on the usability of the interface

as expressed in questions 1 and 2 in Section 4.4.

Concerning the first question, the evidence we obtained suggests that the visualization approach we adopted suits our purposes. The interface is usable, and we obtained useful insights into what should be improved and how. Also, the rollover behavior of the cursor should be combined with some static features in order to allow users to easily inspect the map. All users found the search functionalities accessible from the interface extremely natural because of their familiarity with web search engines. In fact, the users' familiarity with the free-text type of search, together with the type of background they have, strongly suggests that more refined search tools are not essential to the usability of the LoLaLi map. Interesting work for the future would be to more deeply integrate searching and browsing in a way such that search results are displayed in a coherent fashion with the rest of the interface (for example, instead of presenting a list of results, results could be presented together with a small fragment of map around them, and also making the gloss available to the users).

The graph structure of the LoLaLi map was well understood and not at all problematic. As for the relations, the main ideas behind them were understood by all participants, but some confusion remained. The lesson learned from the user studies was then to modify the set of relations used in order to minimize the source of confusion: the results of this process were presented in Chapter 3. We avoided the label "unspecified subclass" (that was meant to be temporary) and preferred to use the *subtopic* relation instead of the *subclass*. We also preferred to change the name of the relation "notion" into the longer but more expressive "feature and internal machinery." Further user studies are advisable as a future work in order to investigate the understandability of the revised set of relations and of the revised interface.

As for the second question, the user studies showed the importance of the background of the users for the understanding and navigation of the map, for example when selecting a path to follow. One way to address this issue consists in improving the visual features in the interface, possibly together with good context sensitive help. Other ways would directly take into account the background of the user, for example by allowing personalization of the kind of help to be shown, or by defining different user interfaces for different groups of end users.

The fact that users do not consider bookmarks and back buttons important can be taken as a confirmation that the map is easy to navigate and that users have little or no difficulties in finding their way through it. On the other hand, they considered with attention the option of filtering the list of child topics: in general, people like to have a complete view of the entire list of children available, but this is obviously cumbersome to achieve when the list of children increases. A semantically oriented way to apply such a filter would take into consideration the available relationships and would allow the user to visualize only those topics related by one (or more) relationships. This option is certainly worthy of further investigation, but other selection criteria could also be valid and perhaps more informative to the user. The key question then becomes how to combine the strict symbolic information encoded in maps like ours with a notion of ranking/preference that could enable agile use of the structure.

The evidence provided by our user study confirms the usability of the graphical user interface proposed in this chapter. Looking forward, they suggest that we carefully consider the visual clues within the interface, and that master students are the best target for the LoLaLi map visualized by the proposed interface.

One of the sections of the questionnaire concerned the linking from the map to the text. Even though that feature was only partially implemented in the system used for the user study, we received encouraging feedback. Both undergraduate and masters students consider this a feature to develop and possibly to enrich with "metadata" about the content of the page they are going to inspect.

In the next two chapters we work on the selection of appropriate link targets for the topics in the map. We first inspect and compare methods for topic segmentation (Chapter 5), then we work on connecting the extracted segments with the map (Chapter 6).

CHAPTER 5

Looking for Link Targets

Data is not information,
which is not knowledge,
which is not wisdom.
(Bruce Horn)

Deep inside... it's shallow.
[Cooper, 1981]

After the presentation of a map for the domain of logic and linguistics (Chapter 3), we performed user studies to assess its usability (Chapter 4). The user studies confirm that the hierarchical structure is intuitive and the interface usable. In particular, they confirm that users find it important to have associated with each topic in the map both larger text fragments and concise glosses. In this and the following chapter, we explore the issue of *linking* the map to the handbook so as to identify the "larger text fragments" that we just mentioned. The issue of linking resources to one another has been widely addressed in the hypertext community, where numerous attempts have been made to clarify the "meaning" of a link, typically through classifications or taxonomies of link types. In the present chapter we do not attempt to propose a new categorization or new categories of links. Instead, we concentrate on the automatic identification of *link targets*, i.e., text fragments to which the link points. The selected links will be links of "aboutness," where the notion of aboutness is established by means of a *similarity measure*, as used in Information Retrieval (IR), between a representation of the topic in the map and the text. (For a survey on the notion of "aboutness" in subject indexing see the classical paper [Hutchins, 1977].)

In our perspective a link target is a textual document, or rather an excerpt or segment of a text, such that:

1. it includes only relevant content, with respect to a given topic, and

2. it is "self-contained" from the point of view of readability.

In other words, according to the first constraint above, the segment should be maximally specific, while the second constraint involves what we call rules of "politeness" to the reader: for example, anaphora should be resolved within the text, and sentences with adverbs suggesting inferences (like "so," "then," "therefore," and so on) should not be separate from their premises. Also, it is important that the segment include relevant content and that the relevant part be right at the beginning of it.

Since every text has internal structure, corresponding to the topics the author of the text wants to present, one obvious approach to get at the kind of segments we seek to identify is to adopt a so-called *structural view* on text segments, and take segments to be nothing but paragraphs. How does this strategy compare to so-called *semantic segmentation* produced by state-of-the-art segmentation algorithms that are based on features of the text and aim at reflecting its content?

In Information Retrieval, passage retrieval techniques have been widely studied to improve the result of the document search task [Kaszkiel and Zobel, 1997, Liu and Croft, 2002, Salton et al., 1993b, Wilkinson, 1994]. Approaches to passage retrieval can be classified on the basis of the unit they consider as a passage. *Structural passage retrieval* takes as units the very same unit into which most texts are already divided (sections, subsections, etc.). Another passage retrieval technique is based on *fixed-sized windows*, i.e., sliding windows of text, whose size is usually expressed in terms of the number of words they contain. Finally, it is possible to look for more semantically oriented segments: if this is the case, the process of identifying passages is often called *topic segmentation*.

We explore and compare two structural passage retrieval methods (when units are single paragraphs and when they are entire sections) and two well-known algorithms for topic segmentation: TextTiling [Hearst, 1997] and C99 [Choi, 2000a,b]. We apply these segmentation methods to the *Handbook of Logic and Language*, and in order to evaluate the results, we build a gold standard, i.e., a reference corpus, by manually annotating *topic breaks* in two out of the twenty chapters in the handbook.

In the next chapter we address the connection between segments and the topics in the map. We then report on a second set of experiments, where each segment is considered as an individual "document," or rather pseudo-document, that can be linked to. All together, these segments form a collection of pseudo-documents to retrieve using queries derived from the topics in the map. The retrieved segments become then our candidate link targets. The evaluation of these targets is done on the basis of a second gold standard, that we created manually by annotating the relevance of paragraphs in the handbook with respect to the query.

The remainder of this chapter is organized as follows. In Section 5.1 we present the main approaches to passage retrieval: structural, fixed size, and semantic (often called topic segmentation). In Section 5.2 we describe two algorithms for linear topic segmentation: TextTiling and C99. In Section 5.3, we discuss the evaluation of algorithms for topic segmentation and describe the creation of an annotated corpus for that purpose. In Section 5.4 we present our first set of experiments, which are designed to establish how good the considered topic segmentation algorithms are at identifying topic segments in a scientific domain. In Section 5.5 we discuss the results we obtained, and in Section 5.6 we draw conclusions.

5.1 Passage Retrieval

Information retrieval deals with the retrieval of documents in answer to a query expressing a user's information need. In the case of long documents, the estimated relevance to a query can be "diluted" by the size of the document. For this reason IR researchers have looked at the possibility of analyzing *passages* of a document, either to gain *evidence* about the relevance of the whole document, or to retrieve passages instead of full documents. We use the expression "passage retrieval" to refer to IR approaches that look at passages for one of these purposes. The contribution of a passage to document retrieval performance, typically in terms of ranking, is often called *evidence*.

In the rest of this section we will illustrate the principles of the three approaches to passage retrieval, structural, fixed size and semantic, then we concentrate more in depth on the semantic passage retrieval.

Applications of passage retrieval in document retrieval include the following:

- reducing the amount of text to present to the user [Salton et al., 1993b],
- selecting excerpts to index independently [Hearst, 1994b],
- improving ranking [Callan, 1994, Kaszkiel and Zobel, 1997].

Reduction of the amount of text presented to the user is often applied after the document retrieval phase, as happens in automatic Question Answering (QA), to select the portion of text from which to extract the answer. When the focus is on indexing, passage retrieval helps to select homogeneous excerpts of text characterized by a high degree of internal coherence and a high degree of dissimilarity to the rest of the text. The resulting excerpts are indexed independently of the rest of the document. When used to improve ranking, similarity values are computed for all passages with respect to the query. Then the score of the document (determining its position in the ranking) is computed on the basis of a combination of the scores for the whole document and for the passages. In general, passage retrieval is used when a high degree of precision is required, as in QA, where it is particularly helpful in the case of long documents [Monz, 2003]. Passage retrieval also finds applications in text summarization [Mani and Maybury, 1999], where it is used to select homogeneous and relevant excerpts to summarize.

5.1.1 Structural Passage Retrieval

Usually, published textual documents are divided into parts, such as sections, subsections and paragraphs. In structural passage retrieval those parts are used as passages. If documents are encoded in a markup language (e.g., XML, SGML, or LaTeX), passages are naturally identified by some elements of the markup (e.g., sections or subsections).[1] Salton and Buckley [1991] propose a method to use the vector space similarity between segments of texts (e.g., paragraphs) in order to compute a finer notion of similarity between (full) documents. According to their proposal, two documents are similar if they exhibit both a *local* and a *global similarity*. Global similarity means looking at the similarity between the two documents (or between a document and a query), while local similarity means looking at the maximum value of pairwise similarity between the components of the document. Salton and Buckley [1991] normalize the term frequency component in order to make the measure independent of paragraph length. Later, Allan [1995] uses the notion of local/global similarity for the automatic generation of typed links among full documents. Wilkinson [1994] investigates the contribution of structural passage evidence to document retrieval, where passages are sections. His conclusion is that passage evidence can help document retrieval, and in particular that the benefit from using passage evidence is most obvious when (high) precision is more important than recall.

Hearst [1994a] argues that markup information, such as the division of text into paragraphs, is less reliable for segmentation than information extracted from the content. According to Hearst, this is a reason to chose semantic passages. In the case of high quality publications such as the *Handbook of Logic and Language*, the internal division of the text into paragraphs tends to be reliable in terms of internal homogeneity. What changes across the book is the average size of the paragraphs and the internal division of the chapters, both related to the writing style of the authors and to the internal organization of the text. In particular, the fact that chapters can be organized with different levels of granularity (cf. Table 5.2) makes it difficult to decide beforehand what unit to use. Also, sometimes a paragraph can be too short to exhaustively include a topic, while entire sections can be too long.

5.1.2 Fixed Size Passage Retrieval

Paragraphs, and sections for that matter, may vary much in length, thus affecting passage retrieval. One way to overcome this problem is to use length normalization procedures, like Salton and Buckley [1991]. Another way is to consider a passage as consisting of a fixed number of words (or characters, or sentences). In the following we investigate this approach.

[1] Since 2001, INEX, the INitiative for the Evaluation of XML retrieval [Fuhr et al., 2006], has studied the potential of using XML document structure for retrieval purposes, considering every element to be a retrievable unit. For reasons explained below, we do not want to go “below” the level of paragraph, and will view paragraphs as our “atomic units.”

Callan [1994] studied the contribution made by fixed size passage retrieval to document retrieval. A passage consists of a window of 100, 200, or 300 words (the value is fixed empirically, depending on the collection), evaluated at query evaluation time instead of during a preprocessing phase. The window slides along the text, overlapping by half of its length (so that each window, except the first and the last ones, overlaps two others): this is the reason why it is called a *sliding window* technique. The starting point is the location of the first identified occurrence of a query term in the document. Callan compares three different strategies for document retrieval: (i) using the similarity between the query and the entire document, (i) using the similarity between the query and the best passage, and (iii) using a combination of the first two strategies (i.e., again a combination of local/global similarity). His results suggest that the latter method yields the best results. Note that length normalization is not a problem for this approach, but semantic ambiguity is. The sliding window approach is widely used, for example in QA, where the passage length is often expressed in terms of natural language sentences [Lee et al., 2001, Light et al., 2001, Llopis and Vicedo, 2001]. As we will see in the next section, a semantically oriented passage retrieval algorithm such as TextTiling also uses the sliding window technique as an intermediate step to identify semantically oriented passages.

The fixed size approach has proven to be useful in both document retrieval and question answering for computing evidence from passages. It has also shown to be more effective (in terms of contribution to document retrieval) than approaches to passage retrieval based on structural passages [Kaszkiel and Zobel, 1997]. Unfortunately, fixed size passage retrieval provides no indication of where to split the document in a reader-friendly manner, therefore it is not suitable for the definition of candidate link targets. In this respect, structural passage retrieval and semantically oriented passage retrieval are better options to overcome that problem.

5.1.3 Semantic Passage Retrieval

Neither of the two approaches to passage retrieval presented so far, structural and fixed size, *explicitly* addresses the issue of dealing with the "content" of the passage. Obviously, the fixed-size approach does not consider content, but even in the structural approach content is not emphasized. In fact, while a paragraph can be too short to capture a "topic" (which can span two or more paragraphs, for example, the first being devoted to a definition, the second to an explanation of the definition with examples or comments), an entire section can be too long. Semantic passage retrieval attempts to define passages on the basis of the semantics of the texts, i.e., on the basis of the topic in the text. For this reason it is also known in the literature as *topic segmentation*.

Hearst and Plaunt [1993] propose to identify semantic passages for the purpose of more accurate indexing. Other approaches have centered on the possibility of recognizing topic shifts in streams of news [Stokes et al., 2002]. From our point of view, the possibility of correctly identifying a topic means that we can present the reader with a readable excerpt from the text, are sensibly shorter than the entire document. In our view, a readable text should not include unresolved anaphoric references, nor should discourse connectives such as "on the one hand, [...] on the other hand" be disconnected. In general, salient argumentation blocks should be in the same segment. The anaphora phenomenon is the most widely studied: Morton [1999] finds that looking back two sentences makes the antecedent available 98.7% of the time, while looking back only one sentence makes the antecedent available 96.9% of the time. He does not report data about the availability of the antecedent with respect to their collocation in the paragraph, but his observation suggests that both reference and antecedent tend to be placed in the same paragraph. In order to take into consideration the antecedent, we would need a discourse parser (work on this is currently ongoing, see for example [Webber, 2004]). All these considerations encourage us, or rather force us, to take paragraphs as an atomic unit for topic segmentation (which then becomes semantics-based paragraph aggregation).

Among the many proposed algorithms for topic segmentation [Hearst and Plaunt, 1993, Ponte and Croft, 1997, Reynar, 1998, Salton et al., 1996a,b], we have selected TextTiling and C99. We chose TextTiling because of its established position in the literature, and C99 because of the good results reported in [Choi, 2000a]. In the next section we outline two families of approaches to topic segmentation, linear and hierarchical, and describe the two linear algorithms chosen in detail.

5.2 Linear Topic Segmentation: Two Algorithms

A topic segmentation is *linear* if the identified segments follow one another, with no distinction between topic and subtopic, and no possibility of nesting subtopics to form topics. In contrast, a segmentation is *hierarchical* if it allows nesting of topic segments. For example, Grosz and Sidner [1986] and Mann and Thompson [1987] adopt a hierarchical model of discourse, also embodied in the approach to topic segmentation taken in [Yaari, 1997].

A discussion on whether hierarchical segmentation better describes the argumentative structure of a text than a linear one is beyond the scope of this thesis. For our purposes, though, a linear segmentation is preferable because it allows us to concentrate on the shift from topic to topic, without keeping track of the topic/subtopic organization of the text. In other words, it is an appropriate tool to delimit excerpts from the text to connect to the map.

In the following we describe in detail TextTiling and C99.

5.2.1 TextTiling

TextTiling [Hearst, 1994a,b] is probably the best known algorithm for linear topic segmentation of expository texts. Following Skorochod'ko [1972], Hearst assumes a linear structure of text and looks at word occurrences in order to trace the development of a topic.

The algorithm tokenizes the document (i.e., divides the text into individual lexical units), and preprocesses it in order to minimize the variety of forms of the same word (i.e., it applies stop word removal, and stemming or morphological normalization of the remaining words). TextTiling divides texts into *pseudo-sentences* of fixed length, which are grouped into pseudo-paragraphs, or *blocks*, of fixed size sliding along the text. The length of the pseudo-sentence and the size of the blocks are the two adjustable parameters of the algorithm. Hearst found that pseudo-sentences of twenty words and blocks of six pseudo-sentences work best in practice. Real paragraphs are not taken as the unit for computing similarity, so as to avoid a length normalization step.

The sliding window method described above allows one to compute a cosine similarity value between all pairs of adjacent blocks. Each pseudo-sentence gap is assigned a similarity value resulting from the comparison of the blocks forming a window interfacing at that gap. Figure 5.1 gives a schematic representation of a text divided into blocks of equal size. Windows sliding along the text are represented below the line: the first window starts from the beginning of the text and spans six sentences (from gap 0 to gap 6), the algorithm computes the similarity value between its adjacent window (from gap 6 to gap 12), and assigns it to gap 6 (the little circle between the two windows). The next window starts at gap 1, the following one at gap 2 and so on.

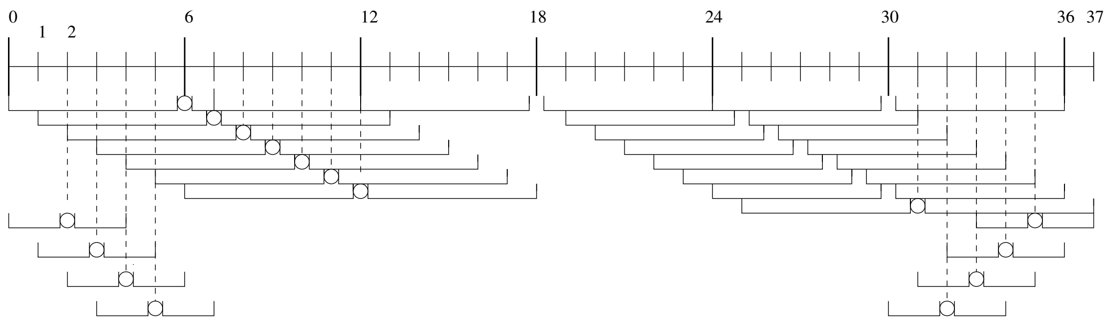


Figure 5.1: Grouping of blocks by sliding windows in TextTiling.

Gaps before 6 and those at the end of the text obtain a similarity value by using smaller windows, of size 2 blocks.

Similarity values are not directly used to compute the paragraph breaks, because they only indicate the grade of similarity between adjacent windows of text, without indication of how that value of similarity has to be interpreted in the picture of the distribution of similarity along the text. Therefore, TextTiling uses a more sophisticated procedure to compute topic breaks. After assigning a similarity value to each gap, these values are smoothed by using a discrete convolution. Next, the result is further smoothed with a simple median smoothing algorithm [Rabiner and Schafer, 1978] with a window of size 3, to eliminate small local minima. The smoothed similarity values are then plotted against the sequence of gaps, and the resulting plot is analyzed for peaks and valleys. Each gap is assigned a *depth score*, indicating how strong the evidence is that the present gap is a candidate topic break. Let $ds(g)$ be the depth score at gap g, it is computed as $ds(g) = (a_s - g_s) + (b_s - g_s)$, where a_s and b_s are the smoothed similarity values at gaps a and b, to the left and to the right of g respectively, each being a peak with respect to g. In other words, the value assigned at each gap is computed by summing the difference between the similarity value at g and the peak to the left and the difference between the similarity value at g and the peak to the right. The intuition is that the deeper g is with respect to the closest peak to the left and to the right, the more likely it is that the gap is a candidate break. Finally, TextTiling takes the gaps with highest depth score as candidate subtopic boundaries, but only places the topic boundary at a (real) paragraph break, after checking that breaks do not fall too close to one another.

Summarizing, TextTiling has a clear intuitive interpretation in terms of text structure and solves the problem of varying paragraph length by using a combination of sliding window and rounding to the original paragraph breaks.

5.2.2 C99

C99, the algorithm introduced by Choi [2000a,b], differs from TextTiling in that it takes real sentences as units and identifies topic boundaries by means of a divisive clustering method. First, the text is divided into tokenized sentences, then the preprocessing phase (stop word removal, stemming) follows. The algorithm then computes a similarity matrix at the sentence level, where the adopted similarity measure is the usual cosine similarity. Since the cosine measure is sensitive to the length of the sentences, Choi applies a *ranking scheme* (based on the one proposed in [O'Neill and Denos, 1992]) to the similarity matrix to avoid using absolute values. The *rank mask* is the square defining the neighbor region around a certain value in the matrix: Choi suggests using a 11×11 mask. Each value in the similarity matrix is then replaced by the number of neighboring elements with lower similarity value, i.e., each similarity value is replaced by its rank. Moreover, since the rank is incomplete along the borders of the matrix, the values have to be normalized somehow: Choi proposes to use the proportion of elements with a lower value over the total number of elements examined. It follows that when using an 11×11 rank mask, the range of the new value is between 0 and 10. Finally, a hierarchical divisive clustering method (based on [Reynar, 1998]) is applied. At each step in the clustering procedure, a segment boundary is selected such that it maximizes a measure of internal segment cohesion. If the number of desired segments is not given, the clustering process is continued until no further segmentation is possible. For all segmentations, the algorithm computes a measure of internal cohesion of the clustering, and the change in the internal cohesion from one segmentation to the next. Then, the algorithm chooses as optimal segmentation the one whose change in internal cohesion is above a certain threshold, computed from the change in internal cohesion of the entire sequence of segmentations.

C99 performs topic segmentation without explicitly referring to specific textual or linguistic figures. It performs a linear segmentation, where the linearity is implicitly obtained by the degree of similarity between close and distant sentences in the text.

Also, the application of the ranking scheme has no straightforward interpretation in linguistic terms, nor does it have a mechanism to stop the clustering procedure. The experiments reported in [Choi, 2000a] were performed on an artificially generated corpus of 700 samples. A sample was taken to be a concatenation of ten text segments, where each segment consists of the first *n* lines extracted from a randomly chosen document from the Brown Corpus.[2] The performance of the algorithm was evaluated in terms of efficiency (the average number of CPU seconds required to process a test sample) and in terms of effectiveness: its precision was evaluated using the P_D measure we describe in Section 5.4.1. We decided to use C99 in our experiments in order to see whether the good performance shown in those experiments could be replicated when using a "real life" text collection instead of an artificial one.

5.3 Building the Ground Truth

A text segmentation algorithm can be assessed in three ways: by comparing it against a manually annotated corpus (annotated by the author of the texts, or by other annotators), by comparing it against an "automatically" annotated corpus, or extrinsically, by assessing how the segmentation improves the computational task in which it is used.

Since the authors of the texts are usually not available for annotation, it is common practice to elicit the annotation from other people, who may be given little or no instruction about how the segmentation should be performed. If this is the case, the aim of the annotation is to capture the intuitive notion that people have about what a topic is. For example, Hearst [1994b] and Klavans et al. [1998] do not supply precise guidelines to the people that perform topic boundary annotations, in order to see if and what similarities emerge from their annotation. A common way to automatically generate an annotated corpus is to take a corpus of news-wires (with annotated topic breaks), eliminate breaks between news, and possibly mix the news at random. The goal of the topic segmentation system in that case is to rediscover the original topic breaks. This is the type of evaluation used in the experiments described in [Choi, 2000a].

We decided to build an annotated corpus by annotating two chapters from the *Handbook of Logic and Language* and use them as a gold standard. Our purpose is to experiment with the difficulties that a real corpus offers, and have a chance to reflect on the peculiarities of working with a scientific handbook. The expected advantage of this approach (as opposed, say, to using an automatically created corpus) is that we have a model of how segmentation for our purposes should be done. The disadvantage is that the process is time consuming and will lead to a relatively small corpus. In the following we describe and discuss the creation of a corpus of manually segmented topics. In Section 5.3.3 we discuss the topic boundary annotations of the *Handbook of*

[2] The Brown Corpus of Standard American English was the first of the modern, computer readable, general corpora. It was compiled by W.N. Francis and H. Kucera from Brown University (USA). The corpus consists of one million words of American English texts printed in 1961. The texts for the corpus were sampled from 15 different text categories to make the corpus a good standard reference.

Logic and Language.

5.3.1 Inter-Annotator Agreement

As with many other annotation tasks, the style and quality of the annotation depend on the specific annotators. This is a well-known issue in the area of both information retrieval and computational linguistics [Lesk and Salton, 1969].

Many annotators, many annotations Since any annotation depends on the annotator who performs it, it can be useful to recall the distinction between 'lumpers' and 'splitters,' as it is defined in lexicography to distinguish different behaviors in the building of dictionary definitions. Lumpers look at similarities and tend to provide fewer definitions, broad enough to cover several cases; splitters look at differences and tend to provide more specific definitions, each covering a smaller set of cases. The same distinction is also used by Klavans et al. [1998], who asked a group of naïve readers (i.e., with no previous notion of topic segment nor instructions about how to perform the task) to annotate a text for topic boundaries. The authors found that the differences among the performed annotations could be described in terms of splitter or lumper annotators, since a few important topic breaks would be recognized by all people, while other less prominent topic shifts would be perceived as a topic break only by some.

On a more formal level, Passonneau and Litman [1993] discuss at length the issue of agreement among annotators in a discourse segmentation task.[3] The authors ask seven subjects to segment a number of discourse transcripts using an informal notion of speaker intention; when four or more agree on a boundary, the boundary is considered significant using a variation of the Q-test [Cochran, 1950]. In practice, this methodology considers the majority as equivalent to the judgment of an experienced reader, where judgments are expressed on the basis of an intuitive notion of what a topic is. This approach is used by Hearst [1994b], although the author notes that it is not clear how useful the data about the significance is, because the simple majority can be insufficient to objectively claim the presence of a topic boundary. Perhaps a better measure of inter annotator-agreement is given by the *Kappa* test [Cohen, 1960], that also takes into account the agreement by chance as dependent on the number of categories the annotators have to mark—note that in case of segment boundaries there are exactly two categories: break/no break. Here is the definition of $Kappa$:

$$Kappa = \frac{P(A) - P_e}{1 - P_e}, \tag{5.1}$$

where $P(A)$ is the fraction of times that annotators agree, and P_e represents the agreement by chance. The value of $Kappa$ decreases when P_e increases, i.e., when there

[3]That study aimed at finding out whether an intuitive notion of speaker *intention* is sufficiently understood by naïve subjects to yield significant agreement on segment boundaries in corpora of oral narratives.

	Chapter A	Chapter B
# pages	55	54
# sections	13	3
# subsections	0	9
# subsubsec.	0	21
# paragraphs	168	223
avg. par. length	458	320

Table 5.1: Details about the corpus.

is a high degree of agreement by chance. The agreement by chance, P_e, between two annotators is computed as follows:

$$P_e = \sum_{j=1}^{n} \left(\frac{\frac{C_j}{2}}{N} \right)^2 = \frac{1}{4N^2} \sum_{j=1}^{N} C_j^2 \tag{5.2}$$

where j represents the number of categories available (in this case two: break vs. no break), C_j is the sum of the numbers of paragraph breaks that have been assigned the same label j by both annotators. Finally, N is the total number of paragraph breaks.

5.3.2 The Handbook of Logic and Language

The *Handbook of Logic and Language* consists of 20 chapters, on average about 65 pages long, each written by different authors and on independent, though related, subjects. The style of writing varies from chapter to chapter, with different usage of anaphora, different internal structure, and different use of formulas, tables, equations, examples and cross references. However, all chapters have a typical linear writing style, with the assumption that the reader would read the entire chapter, if not the entire handbook, from beginning to end. Also, cue phrases[4] are used with varying frequency: for example, in some more narrative chapters each paragraph might begin with phrases like "As we mentioned above…" or "Therefore," while others make less use of these.

We chose two chapters from the *Handbook of Logic and Language*, that we call here Chapter A and Chapter B to build our annotated corpus.[5] Although similar in length (on paper, approximately 55 pages each), the two chapters exhibit different internal structure and also different writing styles. Table 5.1 summarizes some quantitative aspects of the two chapters.

[4] A cue phrase is a phrase used to highlight specific moment in a text or speech. Examples of cue phrases include: "As we have previously seen…," and "As previously discussed…." Cue phrases in other domains include: "This was X reporting from Y," "Ladies and gentlemen…," "The next speaker is …".

[5] We used Chapter 3 [van Eijck and Kamp, 1997] and Chapter 10 [Muskens et al., 1997] of the Handbook.

We remark that the notion of paragraph in this domain turned out to be rather flexible, mainly because of the many non-textual blocks inserted in the text, such as equations, derivations, definitions, lemmas, propositions and theorems, figures, tables and so on. For example, one would assume that paragraphs are marked and recognizable by indentation, and that equations or list of examples should not force indentation, i.e., a new paragraph. The extent to which this style is applied varies throughout the handbook and in the two selected chapters. We counted as paragraph blocks of text separated by indentation, independently of the non-textual elements they can include (e.g., figures, tables, equations,...). Finally, when counting the length in paragraphs we have only considered words longer than two letters (to discard in-line formulas), and have not considered the bibliographic items when counting the length in pages. As we counted these figures by processing the LaTeX sources, which were written according to different styles, some noise is likely.

Chapter *A* is organized into 13 sections, with no other internal structuring than the paragraph division and blocks of definitions, propositions and so on, explicitly marked as such. Occasionally, paragraphs are characterized by a title, in some cases, they are as long as half a page (about 458 words), in a few other cases they are as short as two sentences, usually when they only consist of definitions, propositions or axioms.

Chapter *B* consists of 3 sections organized into respectively 0, 4 and 5 subsections, spanning 54 pages in the printed version; the text is distributed into 223 paragraphs. Chapter *B* does not contain examples and theorems distinguished as such, nor tables and only a few figures, but it does contain many in-line formulas and lists of formulas (axioms or properties). As in Chapter *A*, paragraphs in Chapter *B* can be quite long, up to the entire length of a the smallest unit (subsection), and on average 320 words long.

5.3.3 The Resulting Ground Truth

We asked two annotators to independently annotate the texts for topic breaks, and then agree on a single annotation. The annotators were given basic guidelines that did do not define a strict notion of topic segment. To the best of our knowledge, no strict guidelines for topic segmentation of written texts are available, despite the fact that there are guidelines available in related efforts such as discourse segmentation [Nakatani et al., 1995]. The recommendations given to the annotators were:

1. a topic segment is an excerpt of text smaller than the original text and of homogeneous topic;

2. segments do not overlap;

3. no segment breaks should be placed within paragraphs;

4. no segment should span more than one section;

The manual annotation of Chapter *A* results in 102 segments, each on average 1.6 paragraphs long, while the manual annotation of Chapter *B* results in 90 segments,

	Chapter *A*	Chapter *B*
# segments	102	90
# paragraphs/segments	1.6	2.5
Kappa (inter-annotator agreement)	0.69	0.84

Table 5.2: Details about the ground truth for segmentation.

on average about 2.5 paragraphs long. The inter-annotator agreement computed by *Kappa* is 0.69 for Chapter *A* (tentative reliability) and 0.84 for Chapter *B* (high reliability). When deciding on the final annotation, the two annotators did not make radical changes, as is reflected by the values of *Kappa* between each annotation and the final version (on average 0.81). All figures are summarized in Table 5.2.

The fact that the *Kappa* score for *A* is only 0.69 is due to the presence of long lists of examples and properties discussed at length in the text. This caused the annotators to have different perceptions about where an appropriate break between segments could be placed, although they placed approximately the same number of breaks (103 and 104). One difficulty in annotating the chapters was the varying frequency of cue phrases used, especially at the beginning of paragraphs, and used to link one paragraph to the preceding one. Another difficulty stemmed from the presence of cross references to examples or equations previously defined in the text. Chapter *B* is a good example of the former case, since the authors often smooth the transition between paragraphs by using phrases like: "Again, ..." or "As previously noted...." This problem was somewhat compensated for by the higher degree of internal structure of the text, compared to Chapter *A*, so that the overall level of inter-annotator agreement is high. The *Kappa* score between the final annotations and the original organization in sections and subsections gives a measure of how much more detailed the final annotation is with respect to the structure of the text: we have a very low value for Chapter *A* (0.39), the chapter with more shallow structure, and a higher level (0.69) for Chapter *B*, with a deeper structure.

The use of cue phrases is related to the publication medium. In fact, the use of references at the beginning of paragraphs is meant to smooth the connection between paragraphs when reading on paper, but it is arguably less necessary when reading on screen, where more nimble constructions (such as hyperlinks) can be adopted. The presence of cross references should be considered when giving recommendations to the annotators. For example, recommending that cross references between pieces of text far apart should not be connected: they can be connected later by hyperlinks.[6] Topic segmentation is especially useful in cases like Chapter *A* that exhibit a shallow text structure and with long lists of examples discussed through the text. The second issue is stylistic, strongly related to the publication medium.

[6]Such a recommendation was actually given to annotators in the next set of experiments (see Section 6.1).

One difficulty we did not take into account when devising the segmentation is the difficulty of distinguishing between different levels of presupposition. It turned out to be not always clear when a reference to something previously introduced could be safely split and when the splitting would prevent the reader from having the *background* necessary to understand the text.

5.4 Evaluating Link Targets Against the Ground Truth

We compared the segmentation performed by TextTiling and C99 with two structural segmentations: one in which each paragraph is a segment, and one in which each section is a segment (we could only use sections because this is the only internal subdivision common to both chapters). We used the implementation of TextTiling and C99 provided by F. Choi and freely available from his web page.[7] All segmentation methods were applied to the two annotated chapters (Chapter A and B) from the *Handbook of Logic and Language*.

The adjustable parameters for the TextTiling implementation we used are the number of smoothing cycles (default $= 5$) and the window size (default $= 30$ words). The adjustable parameters for C99 are the size of the rank mask and the number of segments to find. The default rank mask is 11×11, while if no number of segments is set, the algorithm will use the stop criterion described in Section 5.2.2.

5.4.1 Evaluation Measures for Topic Segmentation

Assuming we have a reference corpus with annotated segment boundaries, we need a metric to measure how good or bad our segmentation algorithm is compared to the standard.

The measures most commonly applied for tasks such as ours are the standard *precision* and *recall*, applied either to topic breaks or to entire segments, and the P_D *precision measure* [Beeferman et al., 1997]. When considering precision and recall as applied to topic breaks we look at how good the system is at recognizing a topic shift. By applying the measure to entire segments we look at the ability of the system to recognize homogeneity within the text. The results will vary accordingly, as the following example depicts. Let $T = \{c, d, e, f\}$ be a fragment of text, where each letter stands for a paragraph, and let $Ref = \{(c, d, e, f)\}$ be the manual annotation for it (meaning that c, d, e, f are grouped into the same segment). Let $TS1$ and $TS2$ be two alternative segmentations for T: $TS1 = \{A\}$ and $TS2 = \{B, C\}$ (Figure 5.2). Now, if we look at entire segments, $TS1$ has returned a correct segment, while $TS2$ has not; looking at topic breaks, $TS1$ has 2 correct topic breaks out of 2, while $TS2$ has two correct out of 3. Although crude measures (they do not give a measure of how distant the hypothesized segment break is from the real break), precision and recall are well understood.

[7] URL: `http://www.cs.man.ac.uk/~mary/choif/software.html`. Code downloaded in January 2004.

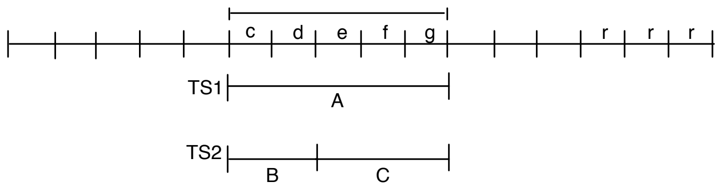


Figure 5.2: Difference between looking at precision and recall of entire segments vs. topic breaks.

Arguably, there can be various "degrees" of error, also depending on the user model adopted. For example, missing a topic break by only one paragraph is not as serious an error as missing it by ten paragraphs, and one can argue that a topic break placed too early is a more acceptable mistake than a topic break placed too late. For this reason, Reynar [1998] suggests judging a boundary correct if it appears within a fixed-sized window of words around an actual boundary. The disadvantage of this measure is that it does not distinguish between correct and incorrect boundaries within the window.

Beeferman et al. [1999] describe a probabilistic error measure P_D for document segmentation: $P_D(r,h)$ is the error of a hypothesized segmentation h with respect to a reference segmentation r. The P_D error measure is defined as follows. Let $I = \{1,2,\ldots,n\}$ be a set indexing the units (paragraphs or sentences) of a document T. For each $i,j \in I$, let r_i denote the segment to which unit i is assigned in some reference segmentation r, and h_i similarly denote the assignment of unit i in a hypothetical segmentation h. The error measure, then, is:

$$P_D(r,h) = \frac{1}{2}\sum_{i,j\in I} D(|i-j|)(\delta_{r_i}(r_j)\bar{\oplus}\delta_{h_i}(h_j)), \tag{5.3}$$

where δ_x is the Kronecker delta function:

$$\delta_x(y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise,} \end{cases}$$

the $\bar{\oplus}$ operator is the boolean `XNOR` operation:

$$x \bar{\oplus} y = \delta_x(y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise,} \end{cases}$$

and D is a pre-defined probability distribution defined over all possible distances between units in T. One can interpret P_D as rewarding hypothetical segmentations that agree with the reference by grouping related units and separating unrelated ones. If the two segmentations agree on every pair of units, $P_D(r,h) = 1$, while if they disagree on every pair, P_D will be very low ($P_D(r,h) = 0$ only if D is chosen such that $D(0) = 0$).

The choice of the probability distribution D is central to the behavior of the error measure P_D. If D is chosen to be uniform, P_D is simply the probability that any two

Segmentation method	Chapter *A*			Chapter *B*		
	Precision	Recall	# Segm.	Precision	Recall	# Segm.
Paragraphs	0.61	**1**	168	0.41	**1**	223
Sections	**1**	0.10	13	**1**	0.02	3
TextTiling	0.62	**1**	165	0.42	0.98	212
C99	0.57	0.08	14	0.57	0.14	24

Table 5.3: Values of precision and recall on topic breaks for the four segmentation methods. TextTiling and C99 are applied with all parameters set to their default values. Best scores are in **boldface.**

randomly chosen units are correctly identified as belonging to the same segment or not. This choice for D tends to be too forgiving, however, as it gives equal weight to distant, and hence likely unrelated, units that even the most simplistic segmentation algorithm is liable to correctly separate. The authors instead suggest using a distribution that concentrates the probability mass on a small range of distances, or even on a single, empirically derived distance.

The disadvantage of the P_D precision measure is that it depends on the length of the document, in the sense that in case of a non-trivial segmentation, it is likely that two distant units are not hypothesized as belonging to the same segments—i.e., provided that the whole document is not in a single segment. Since our documents are indeed rather long, we decided not to use the P_D measure. We adopted instead the precision and recall measures applied to topic breaks, because in this way a system producing an output that agrees with the annotation for topic breaks is rewarded even if the size of the segment is not correct.

5.4.2 Results

Table 5.3 shows the performance of the two structural segmentation methods, C99 and TextTiling with default parameters. The evaluation measures adopted are precision and recall computed on segment breaks; the highest values are in **boldface**.

When assuming that segments coincide with paragraphs, all possible breaks are obviously found, which leads to the maximum recall value for both Chapter *A* and Chapter *B*. Under the same assumption, precision is higher for Chapter *A* than for Chapter *B*, which has to do with the fact that reference segments for Chapter *B* are longer than Chapter *A*.

When assuming that segments coincide with sections, we have maximum precision for both chapters, because all section breaks are also topic breaks in the annotation. In contrast, recall is very low for Chapter *A* and even lower for Chapter *B*, which is a direct consequence of the few segment breaks hypothesized.

TextTiling returns almost as many segments as the number of paragraphs, which ensures a recall close (Chapter *B*) or identical (Chapter *A*) to the highest possible. Also, the values of precision follow the same pattern as in the case of segments one paragraph long, although slightly higher.

TextTiling	Chapter *A*			Chapter *B*		
	Precision	Recall	# Segm.	Precision	Recall	# Segm.
default	0.62	**1**	165	0.42	0.98	212
s5-w20	0.61	**1**	169	0.42	**0.99**	215
s5-w30	0.61	**1**	166	0.42	**0.99**	215
s5-w40	0.61	0.89	163	0.42	0.96	209
s10-w30	0.62	0.96	159	0.43	0.95	200
s10-w40	0.61	0.94	157	0.44	0.91	190
s20-w30	**0.64**	0.83	132	**0.46**	0.79	157
s20-w40	**0.64**	0.80	128	0.43	0.71	150

Table 5.4: Tuning parameters for TextTiling. Best scores are in **boldface**

C99	Chapter *A*			Chapter *B*		
	Precision	Recall	# Segm.	Precision	Recall	# Segm.
default	0.57	0.08	14	0.57	0.14	24
r7	0.53	0.08	15	0.61	0.15	24
r9	0.54	0.07	13	**0.62**	0.11	17
r11	0.57	0.08	14	0.61	0.15	24
r13	0.54	0.07	13	0.57	0.14	24
r15	0.57	0.08	13	0.57	0.14	24
r17	0.57	0.08	14	0.57	0.14	24
r57	**0.72**	**0.13**	18	0.60	0.16	25

Table 5.5: Tuning parameters for C99. Best scores are in **boldface**

Finally, C99 has the lowest precision scores for Chapter *A*, but the highest for Chapter *B*. On the other hand, recall falls dramatically for Chapter *A* and is very low for Chapter *B* (but higher than in the case of "sections as segments"). The low value of recall is related to the low number of segment breaks hypothesized, and only about half of those agree with the gold standard.

Next, we experiment with tuning parameters for TextTiling (results are shown in Table 5.4). We tried different combinations of smoothing cycles (s in the table) and window size (w in the table). By enlarging the window size from 30 to 40 words, but keeping the same number of smoothing cycles, we find little variation in both precision, recall and number of hypothesized segments. On the other hand, an increase in the number of smoothing cycles has a clearer effect, especially on the size of the segment. By increasing the number of smoothing cycles the relative differences among similarity values are smoothed and the resulting segments will be longer. As a consequence, fewer segments breaks are hypothesized.

Table 5.5 shows differences in performance when varying the size of the rank mask (r in the table) for C99. The algorithm with default parameters results in a very ag-

	Chapter *A*				Chapter *B*			
	P	R	F	# Segm.	P	R	F	# Segm.
Paragraphs	0.61	**1**	0.76	168	0.41	**1**	0.58	223
Sections	**1**	0.10	0.18	13	**1**	0.02	0.04	3
TT default	0.62	**1**	**0.77**	165	0.42	0.98	**0.59**	212
TT s5-w20	0.61	**1**	0.76	169	0.42	**0.99**	**0.59**	215
TT s5-w30	0.61	**1**	0.76	166	0.42	**0.99**	**0.59**	215
TT s20-w30	**0.64**	0.83	0.72	132	**0.46**	0.79	0.58	157
TT s20-w40	**0.64**	0.80	0.71	128	0.43	0.71	0.54	150
C99 default	0.57	0.08	0.14	14	0.57	0.14	0.22	24
C99 r9	0.54	0.07	0.12	13	**0.62**	0.11	0.19	17
C99 r57	**0.72**	**0.13**	0.22	18	0.60	0.16	0.25	25

Table 5.6: Summarizing the results for the two structural segmentations, and the best performing versions of TextTiling and C99.

gregative segmentation, and results change little when changing the parameters. The rank mask size does not affect the result of the segmentation in terms of number of segments hypothesized. We did not use the other parameter available, the number of segments in which to divide the text, because for our purposes a topic segmentation algorithm should not be told in advance how many segments there should be (and how long). Only for the sake of comparison with TextTiling, though, we imposed the same number of segments as hypothesized by the best versions of TextTiling. As Table 5.5 shows, both precision and recall increase, together with the number of hypothesized segments.

5.5 Discussion

The parameter variation in TextTiling and C99 did not yield a dramatic performance improvement (cf. Tables 5.3, 5.4 and 5.5). In particular, C99 turns out to be the worst performing algorithm: when default parameters are used, the algorithm returns few, very long segments, too long to be of effective use in this application. Varying parameters does not yield significant differences in the resulting segmentation. We hypothesize that the reason for this is that the stopping criterion used by the algorithm is not suitable to the type of text we deal with. We deem it important that an algorithm for segmentation in this domain be able to decide by itself how many segments it should return. In other words, the algorithm should be able to detect topic shifts on the basis of parameters other than the exact number of segments to return. Moreover, the parameters in C99 are difficult to interpret from a linguistic point of view, which makes it hard to hypothesize about and interpret the effect of tuning them. We suppose that the good results the algorithm achieved in the experiments reported in [Choi, 2000a] are related to the kind of collection it was evaluated against, and to the measure of accuracy used (the P_D measure we discussed in Section 5.3).

TextTiling behaves better on Chapter *A* than on Chapter *B*, while it is the other way around for C99. This is probably related to the type of text and the type of segmentation they perform: TextTiling is more like a splitter (which matches with the Chapter *A* gold standard), while C99 is more like a lumper (matching with the Chapter *B* gold standard). The precision of C99 improves greatly using a large rank mask (57) in the case of Chapter *A*, although recall remains very low. TextTiling gives the best balance between precision and recall.

Table 5.6 summarizes the performance of the two structural segmentations and that of TextTiling and C99 with default parameters and with best results obtained. The two structural segmentations exhibit a somewhat symmetrical behavior: segments that are one paragraph long score best in recall but very low in precision, while segments one section long exactly the opposite. On the other hand, C99 and the segmentation based on sections produce a similar number of segments and similar recall for Chapter *A*. In the case of Chapter *B*, they again score very low and again there is a correlation between the number of segments and the recall value. This suggests that the quality of a segmentation is strictly related to the number and size of segments in the reference annotation.

5.6 Conclusions

In the course of this chapter we addressed our Research Question 3, which reads: "What are suitable targets in the handbook to establish focused, topic driven links from topics in our browsable map?" We assumed that passage retrieval techniques should serve our purpose and compared two structural segmentation (by paragraph and by section) techniques and two semantic passage retrieval techniques (TextTiling and C99). In order to perform a comparison, we created a ground truth by manually annotating two chapters of the *Handbook*, and compared the segments generated by the four automatic methods selected with the segments marked relevant by the annotators. As evaluation measure, we used precision and recall applied to topic breaks. We found that the best algorithms for this task are TextTiling and the structural segmentation by paragraphs, while C99 and the structural segmentation by sections are to be avoided. The tuning of parameters of TextTiling did not give sensible improvement of the results.

The results in Tables 5.3 and 5.5 indicate to what extent the considered algorithms are able to repeat the segmentation made by the human annotator. As previously noted, they do not say anything about how good the resulting segmentation is as a basis for the production of candidate links to be connected to the map. In other words, our Research Question 4 is still to be addressed. The work we present in the next chapter addresses that question. There, we report on our experiments with the retrieval of the four sets of segments described in this chapter. That is, in the next chapter we will consider what happens when a collection of segments is considered as a collection of candidate link targets. For that purpose we will use both algorithms—TextTiling and C99—with default parameters, and the combination of parameters reported in Table 5.6.

CHAPTER 6

Connecting Map and Link Targets

In the previous chapter we have seen how two topic segmentation algorithms and two structural segmentation methods perform against a manually created gold standard segmentation on a domain specific corpus, viz. the *Handbook of Logic and Language*. Now, we explore the use of the segments obtained using these algorithms and methods as targets of link targets. Specifically, our aim is to find link targets for topics in the LoLaLi map, so that the topics can serve as starting points for exploring the *Handbook*. We operationalize the task of finding link targets as follows: we select a set of topics from the map and use them as queries to run against the collection of segments. For evaluation purposes we need to define a gold standard as well as appropriate measures to evaluate the results.

The remainder of this chapter is organized as follows. In Section 6.1 we discuss the creation of an annotated corpus to evaluate the results of the retrieval runs. In Section 6.2 we introduce three evaluation measures that we use to evaluate single retrieved segments. In Section 6.3 we generalize these measures to evaluate sets of segments. In Section 6.4 we describe the experimental setting, present the retrieval methods used, and discuss the results obtained. Finally, in Section 6.5, we discuss the work carried out in the course of this chapter, and formulate our conclusions.

6.1 Annotation of Relevance Assessments

We used topics from the map as sources for which link targets needed to be identified. Topics were selected based on the content of the two chapters used for the annotation described in Section 5.3.3, i.e., Chapter 3 [van Eijck and Kamp, 1997] and Chapter 10 [Muskens et al., 1997] from the *Handbook of Logic and Language*. Titles of topics were taken as queries.

Guidelines. The annotation was performed by a single annotator who annotated relevant paragraphs for each query. The notion of relevance utilized here is binary, where a paragraph is relevant to a certain topic if it "talks about" it.[1] The annotator was given the following guidelines:

1. the minimal unit of relevance is a paragraph;
2. if a paragraph contains a cross-reference (to another section of the document), ignore the cross-reference.

The rationale for the first instruction is to annotate the smallest unit used in the segmentation phase. The rationale for the second instruction is that we hypothesized that cross-references within the text will be rendered as hyperlinks and therefore immediately available to the end user. We used the same two chapters as for the experiments described in Section 5.3.3 in order to gain multiple views of the same data. The annotator was given the chapters with no indication of the annotation of segments, and he knew nothing about the previous segmentation.

For the annotation we used 37 queries and 191 paragraphs were found relevant (43 in Chapter 3, 148 in Chapter 10): on average, each query has 5.1 relevant paragraphs.

6.2 How to Evaluate a Link Target

The manual annotation for relevance is given in terms of blocks of paragraphs, and, as we have just seen, segments can be one or more paragraphs long. Then, a retrieved segment might include only paragraphs that are marked as relevant, or none, or some relevant paragraphs. Figure 6.1 gives a schematic representation of these possibilities: the horizontal line represents the entire document and the small vertical lines mark paragraphs. Above the line, relevant paragraphs are those marked by r, and below the line there are the retrieved segments. Then, segment A is totally non-relevant, segment B totally relevant, and segments C, D, E include both relevant and non-relevant paragraphs. Note that this figure does not present the results of an actual retrieval run, as the segments obtained by the application of the segmentation algorithms we used are all non-overlapping.

In our view, a retrieved segment is relevant if it contains at least one relevant paragraph (i.e., segments B, C, D and E in Figure 6.1 are all relevant), but it would also be useful to keep track of the *amount* of relevant matter in a segment returned by a system for generating link targets. Therefore, we introduce a measure for precision of a segment, that we call C-precision (C_p), to account for the number of relevant paragraphs

[1] The analytical review of relevance by Saracevic [1975] is a key paper about how the notion of relevance is understood. For more recent surveys about the notion of relevance in IR, see [Borlund, 2003] and [Mizzaro, 1997]. For the notion of aboutness, we refer to the work by Hutchins [1977].

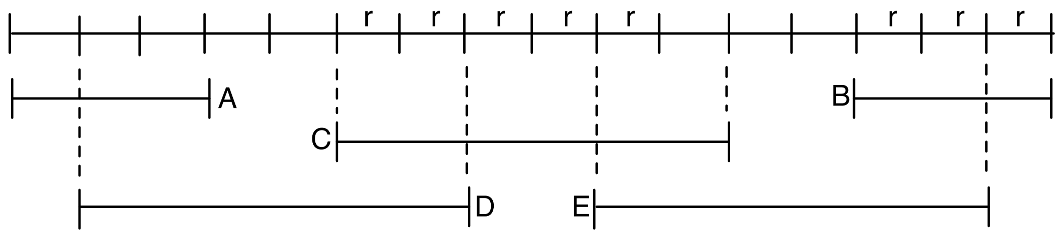


Figure 6.1: Possible configuration of retrieved segment with respect to the annotation for relevance.

out of the total number of paragraphs in the segment returned by a system; see below for more details.

Although informative, this measure fails to distinguish retrieved segments in terms of *where* the relevant paragraphs are, neither inside the segment, nor with respect to the other (contiguous) relevant paragraphs in the document (outside the segment). In other words this measure does not capture the notion of *entry point* to the document [de Vries et al., 2004]. For example, consider Figure 6.1 and the difference between segment C and segment D. Segment C starts with a relevant paragraph, does not have relevant paragraphs preceding it, and includes all relevant contiguous paragraphs in the area; segment D, on the other hand, includes four non-relevant paragraphs before the first relevant one, and leaves out a number of contiguous relevant paragraphs after that.

In order to be able to take these factors into consideration, we introduce two error measures. We say that a segment has an *onset error* if it either begins with one or more non-relevant paragraphs (*early onset error* (EoE)), or if it does begin with a relevant paragraph, but some contiguous (preceding) relevant paragraph was not included in the segment (*late onset error* (LoE)). It has *no onset error* otherwise (NoE), if it begins with a relevant paragraph and has no contiguous relevant paragraphs preceding it. The late and early onset errors are related to the notion of an entry point to the text (cf. Section 2.1). Note that we do not take into consideration the "exit point" of a segment, i.e., its ending point: since the exit point of a segment and the entry point of the following segment are contiguous, considering both would be redundant. In the following we discuss in detail each of these measures.

C-precision. Let S be a retrieved segment. The C-precision of S, $C_p(S)$, is defined as the proportion of relevant paragraphs included in S:

$$C_p(S) = \frac{|Relevant\ in\ S|}{|S|}, \tag{6.1}$$

where $|S|$ is the total number of paragraphs in S.

The segments depicted in Figure 6.1 have the following C-precision scores: $C_p(A) = 0$; $C_p(B) = 1$; $C_p(C) = 5/6$; $C_p(D) = 1/3$; $C_p(E) = 3/6$.

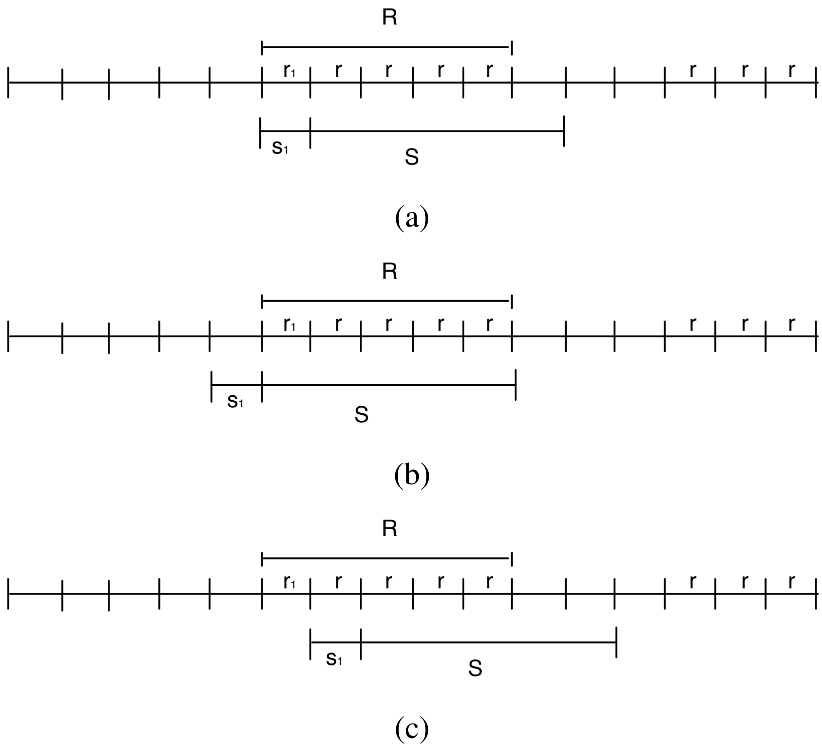


Figure 6.2: (a) Segment with NoE. (b) Segment with EoE. (c) Segment with LoE.

Early onset Error. If a segment begins with one or more non-relevant paragraphs, like segment D in Figure 6.1, we say that the segment has an early entry point (Early onset Error, or EoE), EoE measures the *proportion of superfluous paragraphs* included at the beginning of the segment.

Paragraphs in a text are linearly ordered, so that for any pair of paragraphs a and b ($a \neq b$) it is always possible to say whether $a < b$ or $b > a$ in the order (i.e., a comes before b or the other way around). Therefore, the distance between two points in the text can be expressed in terms of the number of paragraphs separating them. Having made this observation, let us call s_1 the first paragraph of a retrieved segment S. Let r_1 be the first paragraph of a sequence of contiguous relevant paragraphs R intersecting S (Figure 6.2). If $s_1 = r_1$, the segment has optimal entry point (Figure 6.2 (a)), therefore we expect that both early and late onset errors are zero (see segments B and C in Figure 6.1). When $s_1 < r_1$, segment S has an early entry point (see Figure 6.2 (b), or segment D in Figure 6.1), measured by the early onset error.

The *Early onset Error (EoE)* measures the proportion of non-relevant material included in the segment preceding the first relevant paragraph (i.e., $r_1 - s_1 > 0$). More formally:

$$EoE = \min\left\{1, \min_{R: s_1 \leq r_1} \frac{r_1 - s_1}{|S|}\right\}, \tag{6.2}$$

where $|S|$ stands for the size of S expressed in terms of number of paragraphs if contains, and $r_1 - s_1$ is the number of non-relevant paragraphs in S preceding the relevant ones. Note that $\frac{r_1-s_1}{|S|}$ is equal to 0 when there is no error, i.e., the beginning of the retrieved segment coincides with the beginning of the relevant block. And $\frac{r_1-s_1}{|S|} \geq 1$ when the error is maximum, i.e., when the onset is so early that the entire segment includes only non-relevant paragraphs. The minimum in the above equation restricts the range of EoE to exactly $[0,1]$. The advantage of restricting the range of EoE is that segments containing only non-relevant paragraphs have maximum error, independently of their distance from the closest relevant paragraph.

Late onset Error. If a retrieved segment begins with one or more relevant paragraphs, but at least one relevant paragraph precedes its beginning, like segment E in Figure 6.1, we say that the segment has a late entry point (i.e., it makes a Late onset Error, or LoE). In other words, the LoE measures the *proportion of missed relevant paragraphs* at the beginning of the segment. Analogously, if $s_1 - r_1 > 0$ the LoE measures the proportion of relevant paragraphs missed at the beginning of the segment (Figure 6.2 (c)). More formally:

$$LoE = \min\left\{1, \min_{R: r_1 \leq s_1} \frac{s_1 - r_1}{|R|}\right\}, \tag{6.3}$$

where $|R|$ stands for the number of relevant paragraphs at the beginning of segment S. Then, $s_1 - r_1$ is the number of relevant paragraphs preceding s_1, and $\frac{s_1-r_1}{|R|}$ is equal to 0 when there is no error, i.e., the beginning of the retrieved segment coincides with the beginning of the relevant block. When the error is maximum, i.e., when the onset is so late that the entire segment includes only non-relevant paragraphs $\frac{s_1-r_1}{S} \geq 1$. Again, the minimum restricts the range of LoE to exactly $[0,1]$.

A segment with perfect entry point (NoE), i.e., coinciding with the beginning of the relevant block R, will have $LoE = EoE = 0$ (segment B and C in Figure 6.1). A totally non-relevant segment, i.e., with no relevant paragraphs in it, will have $LoE = EoE = 1$ (see segment A in Figure 6.1). No segment can have $LoE = 0$ and $EoE = 1$; similarly, no segment can have $LoE = 1$ and $EoE = 0$.

Discussion. C-precision is meant to give a measure of the quantity of relevant content in a candidate link. It takes into account the size of the segment in an indirect manner, so that a segment consisting of only one relevant paragraph has $C_p = 1$, a segment containing no relevant paragraphs has $C_p = 0$, and a segment including all relevant paragraphs in the document will likely have low precision, how low exactly depends on the size of the document (and on the number of relevant paragraphs in it).

The other two measures, EoE and LoE, have the advantage of providing a relative measure, ranging from 0 to 1. EoE has a bias for longer documents, since it divides the number of non-relevant paragraphs by the total number of paragraphs in the segment. Arguably, shorter documents should not be penalized, although what should or should not be penalized depends on the user modeling we adopt. For example, some users prefer to have longer documents retrieved and navigate into them, others prefer shorter documents to inspect and discard faster.

C-precision and late onset error count relevant paragraphs in a slightly different manner. When computing C-precision, all relevant paragraphs in the segment are counted, while the late onset error only considers the sequence of relevant paragraphs at the beginning of the segment. Let us consider again the example depicted in Figure 6.1. Segment E in that figure contains three relevant paragraphs out of six (its C-precision is 0.5), "distributed" into two blocks, with some non-relevant text in the middle: here we have a segment that contains a mixture of relevant and non-relevant paragraphs. If we are only interested in the amount of relevant content in the segment, then all relevant paragraphs in it should be counted, independently of their position in the text. On the other hand, if we are interested in the readability of the segment, only its first paragraph should be considered and, in that is relevant, the contiguous relevant paragraphs after that.

The three measures just introduced will be used to assess the quality of single retrieved segments. In the next section we aggregate them in order to be able to evaluate the behaviour of entire collections of segments, obtained in the previous chapter.

6.3 Evaluating a Collection of Segments

In order to be able to assign a score to a set of segments (as opposed to a single segment) with respect to a query or even a set of queries, we consider the following ways of aggregating per-segment results:

1. average number of segments with NoE (per topic);
2. average C-precision;
3. total number of non-relevant segments retrieved;
4. average number of non-relevant paragraphs at the beginning of a relevant segment;
5. average early onset error (EoE);
6. average number of relevant paragraphs missed at the beginning of a relevant segment;
7. average late onset error (LoE).

All measures are applied at cut-off three, i.e., only considering the first three retrieved segments for each query. In IR applications based on larger datasets, performance measures based on shallow ranked lists (such as ranked lists that have length three) may be unstable when used with small set of queries [Buckley and Voorhees, 2000, 2004]; however, in our setting and with our user scenario, a cut-off at rank three is a reasonable choice, as our task is highly precision-oriented (cf. Chapter 5) and the dataset (document collection) relatively small.

By looking at the proportion of segments with NoE per query, we look at how many segments per query provide an "ideal" entry point. The value of C-precision averaged for all queries gives a measure of the "amount" of relevant matter contained in all the retrieved segments. The third measure we take into consideration is the total number of non relevant segments retrieved. The last four measures deal with the quality of an entry point to the document. The fourth and fifth measures are coupled together: the average number of non-relevant paragraphs at the beginning of a relevant segment gives a measure of the *effort* required by the reader in order to reach the first relevant paragraph available, while the average EoE gives a normalized measure of that effort. The sixth and seventh measures are analogous to the previous two measures, because the relevant text starts before the beginning of the document, and one has to move back in order to find a good entry point. These measures depend on the size of the document and on the number of contiguous relevant paragraphs available.

6.4 Experimental Setting and Results

Now that we have defined our evaluation measures, we turn to our experiments. The research question that we formulated in Chapter 1 (Research Question 4) is the following: What is the most suitable candidate link to be connected to the map? In other words, we ask what the relative performance of link target finding on different segmentations is. Do "better" segmentations give rise to "better" link target finding results?

In order to answer our question we considered the each sets of segments as a collection of documents within which to retrieve relevant segments with respect to the selected queries. By running these experiments we aim at seeing what the difference is, if any, between the collections of segments obtained with different segmentation methods. The information retrieval engine that we used in the experiments we describe in this chapter was developed at the University of Amsterdam [van Hage, 2004]. Documents are preprocessed using TreeTagger [Schmid, 1994], then indexes and inverted files are stored in a database for reasons of efficiency; the weighting schemas used are tf.idf and OKAPI [Robertson and Walker, 1994]. Both are briefly described in Appendix B: the former schema is widely know as an appropriate weighting schema in case of short documents [Allan et al., 2003], and so is the latter, because it combines idf with a notion of the "normal" size of a document in the collection, expressed as the average of the size of all documents. In this way the average size of documents in the collection is used as a reference point against which to compare the length of a document. OKAPI is still one of the best performing retrieval schemas, despite its age.

For our link target finding experiments we use the outputs of the segmentation methods described in Section 5.3.3 and summarized in Table 5.6. For evaluation purposes, we used the ground truth described in Section 6.1 and the measures described

Segm. method	Avg. NoE/topic	Avg. C_p	Tot. non-rel. segm.	Avg. non-rel. par. begin	Avg. EoE	Avg. rel. par. missed	Avg. LoE
Paragraphs	**1.08**	**0.36**	**69**	**0.00 (0)**	**0.64**	2.11 (17)	0.71
Sections	0.83	0.07	78	7.77 (13)	0.72	3.00 (1)	0.67
TT default	1.06	0.35	70	**0.00 (0)**	0.65	2.12 (16)	0.71
TT s5-w20	1.00	0.33	72	**0.00 (0)**	0.67	2.05 (15)	0.73
TT s5-w30	1.03	0.34	71	**0.00 (0)**	0.66	2.12 (16)	0.72
TT s20-w30	1.06	0.32	71	1.00 (3)	0.67	2.05 (12)	0.70
TT s20-w40	1.06	0.31	71	3.50 (2)	0.67	**1.78 (12)**	0.69
C99 default	0.92	0.09	78	6.44 (16)	0.75	4.88 (5)	**0.66**
C99 r9	0.83	0.06	81	10.78 (19)	0.81	6.00 (1)	0.67
C99 r57	0.97	0.10	77	6.00 (17)	0.76	4.88 (5)	0.67

Table 6.1: Summary values for all collections of segments, across all queries, using tf.idf. Highest scores per measure are in **boldface**.

in Sections 6.2 and 6.3.

Table 6.1 and 6.2 report the scores for the measures introduced in the previous section, respectively for tf.idf and OKAPI. These tables summarize results across all 37 queries used on both chapters.[2] The highest scores are in **boldface**.

Table 6.1 shows that, using the tf.idf weighting scheme, the segmentation into paragraphs achieves the best scores for most of the measures. The reason for this good result is that if a segment only contains one paragraph, and that one is relevant, the entire segment has maximum C-precision and perfect entry point. On the other hand, if the paragraph is non-relevant, the entire segment is non-relevant, and the proportion of early onset does not apply.

Using tf.idf, the segmentation by section gets rather low scores for all measures. In fact, the longer a segment is, the more likely it is that it also contains non-relevant paragraphs, which directly affects the C-precision of the segment. If the segment is long, it is also more likely that the non relevant paragraphs in it are placed at the beginning of the segment. For example, for segments based on sections, the proportion of late onset error is quite high, but there is only one case where it happens, clearly in one case when the annotator judged relevant a sequence of paragraphs spanning two sections. The number of non-relevant segment retrieved when segments are as long as entire sections suggests that tf.idf tends to discriminate short documents better than long ones.

The segmentation obtained by applying TextTiling show figures similar to the seg-

[2]We discarded one of the queries, as it had no relevant text in the documents.

Segm. method	Avg. NoE/topic	Avg. C_p	Tot. non-rel. segm.	Avg. Non-rel. par. begin	Avg. EoE	Avg. Rel. par. missed	Avg. LoE
Paragraphs	**1.06**	**0.35**	**70**	**0.00 (0)**	**0.65**	**2.42 (18)**	0.72
Sections	0.47	0.04	90	6.50 (10)	0.83	3.00 (1)	0.79
TT default	**1.06**	**0.35**	**70**	**0.00 (0)**	**0.65**	2.36 (16)	0.71
TT s5-w20	**1.06**	**0.35**	**70**	**0.00 (0)**	**0.65**	2.36 (16)	0.71
TT s5-w30	**1.06**	0.34	**70**	**0.00 (0)**	**0.65**	2.36 (16)	0.71
TT s20-w30	0.97	0.29	74	1.00 (3)	0.69	2.48 (11)	0.72
TT s20-w40	1.00	0.29	73	3.50 (2)	0.69	**2.42 (13)**	0.71
C99 default	0.78	0.07	85	6.15 (13)	0.77	5.50 (4)	**0.70**
C99 r9	0.67	0.05	87	8.93 (14)	0.81	6.00 (1)	0.73
C99 r57	0.72	0.07	88	4.33 (12)	0.81	5.50 (4)	0.75

Table 6.2: Summary values for all collections of segments, across all queries, using OKAPI. Highest scores per measure are in **boldface**.

mentation by paragraphs, which squares with the observations just made. In contrast, the segmentation obtained by applying C99 gets similar figures to the segmentation by sections.

Table 6.2 presents the results obtained running the retrieval algorithm using the OKAPI weighting schema and the same evaluation measures as in Table 6.1. In general, the results exhibit a pattern similar to those obtained with the tf.idf weighting schema. Values for single paragraphs are quite stable with respect to the previous run, although some values have decreased slightly. Based on the results shown in Tables 6.1 and 6.2 we cannot conclude that there is a substantial difference in performance between tf.idf and OKAPI for our application. Instead, the fact that the results we obtain are consistent, although slightly different, confirms the fact that both weighting schemas are appropriate for retrieving short documents and that tf.idf is hard to beat.

Again with OKAPI, the segmentations obtained by applying TextTiling and the structural segmentation by paragraphs have similar results, if not exactly the same. Also the highest number of non-relevant segments retrieved is obtained when the algorithm retrieves entire sections. The segmentation based on paragraphs yields the highest scores in terms of proportion of NoE, while the division into sections is the one that scores worst. All C99 versions perform only slightly better than segmentation by sections. Again, paragraphs have the highest C-precision, together with a few of the segmentation based on TextTiling. C-precision for sections is extremely low, and all versions of C99 show very similar results.

Concerning the early onset error, we recall that it implies that the segment begins with at least one non-relevant paragraph. It is equal to 1 when the segment is totally non-relevant, equal to 0 when there are no non-relevant paragraphs at the beginning of the segment. The single paragraph segmentation and TextTiling have the lowest average EoE, a fact that is explained by the high precision: since a single paragraph can only be either totally relevant or totally non-relevant, it follows that in case of many relevant segments, there will be many zeros on average. This is also witnessed by the fact that C-precision and early onset error sum to one for this system. C99 with default settings scores the highest EoE, due to the large size of the segments. Concerning the LoE, this time the lowest error rate is scored by C99 with default parameters, immediately followed by the baseline based on paragraphs.

Summarizing, when all measures are taken into account, the best collections of segments are those obtained by applying TextTiling (the closer to the default values, the better) and the structural segmentation by paragraphs. TextTiling produces short segments but not always consisting of single paragraphs (cf. Table 5.6).

6.5 Conclusions

We have considered the link generation process as a result of two distinct phases: first, texts are segmented into topically coherent segments, and then segments are retrieved on the basis of their similarity to the queries. We addressed the fourth question asked in Chapter 1 (Research Question 4): What is the most suitable candidate link to be connected to the map? In order to answer this question, we used information retrieval techniques that we applied to the collections of segments that we obtained in the experiments presented in the previous chapter.

For the evaluation of our work, we introduced a measure for the content of relevant matter in a segment (C-precision), and two measures of error in the entry point (early onset error, late onset error). We approximated the notion of readable entry point by means of a quantitative measure of the distance between the beginning of the segment and the closest relevant text. The error measures we proposed measure the quality of an entry point in terms of error with respect to the relevant paragraphs included or missed by the segments.

Segments based on entire sections score consistently worse than all other methods. On the other hand, the segments based on single paragraphs and the segments obtained with TextTiling score consistently best, with little or no difference. The success of these two types of segments is partly explained by a slight bias of the retrieval algorithms toward short text. Short segments, if relevant at all, also tend to have high C-precision, because an individual paragraph is either totally relevant or not relevant at all. Moreover, when a segment is only one paragraph long, it can only have a late onset error, never an early onset error, and the early onset error becomes a complement of C_p. This explains why the paragraphs have a small EoE value. Interestingly, the other two algorithms that score the next lowest average EoE (in both retrieval settings) are two variations of TextTiling: the size of the segment constraint the entry onset error. The average values of LoE have a smaller range than EoE, and C99 with default parameters is the algorithm with lowest value. Again, the size of the segment influences the onset error: in this case the more inclusive a segment is, the lower the LoE.

The conclusion of the work presented in this chapter is that neither the structural segmentation by sections, nor the segmentation by C99 (based on divisive clustering) can be used as a basis for topic driven access. Instead, segmentations by paragraphs and TextTiling show encouraging results as they tend to maximize the amount of relevant content and minimize the early onset error. In particular, given the fact that paragraphs are very "cheap" to obtain, as they are given together with the text, it is worth experimenting with single paragraphs used as a unit for segmentation, and apply semantic treatments à la TextTiling *after* the retrieval phase. We hypothesize that in this way also the late onset error could be minimized. Also, the cost of the processing would be minimal and it would be possible to select the segments to show to the reader at query-time.

CHAPTER 7

Conclusions

In this thesis we have explored the possibility of providing topic driven access to scientific handbooks by means of a domain map that is automatically linked to appropriate segments inside the underlying handbook texts. For our case studies we made use of the *Handbook of Logic and Language* [van Benthem and ter Meulen, 1997]. Our approach is described in detail in Chapter 2, and the core of our work is presented in Chapters 3, 4, 5 and 6.

In order to facilitate the exposition of our conclusions, we group the Research Questions presented in Section 1.1 into two groups, concerning the map on the one hand and linking on the other (Sections 7.1 and Section 7.2 respectively). Next, we discuss the resulting "bigger picture" (Section 7.3), and ideas for future work (Section 7.4).

7.1 How to Organize and Visualize a Domain Map

We started our work by asking: "What requirements should we impose on a map that is to be used for human browsing and as a skeleton to provide focused access to the text?" (Research Question 1). Focusing on non-expert end users, our answer was to impose three requirements on the LoLaLi map (Section 3.1): inclusion of relevant topics from the domain, informativity to the audience addressed, and a low risk of information overload. The trade-off between informativity and information overload, together with the observation we made during the user studies, led us to define the set of relations used within the map that we presented in Chapter 3. In the course of that chapter we also presented examples to suggest that more refined, formal, or specific relations could be used, but we argued that in that case we would actually be addressing a different type of end user.

Next, we asked: "How do we present the map to readers of a handbook in such a way that we ensure broad coverage of the domain (with detailed information per topic), while making sure that users do not get lost?" (Research Question 2). We proposed a user interface tailored to address an audience of non-expert users (Chapter 4). Our user interface was developed to be available from the Internet, therefore it does not require special software to be installed by the user. It shows the main features of the map while hiding the complexity of the structure, and it integrates browsing and searching. The usability of the proposed user interface was tested during the user studies we conducted (Chapter 4). The proposed interface was found to be clear and usable by both groups of end users on which it was tested (undergraduate and masters students), although masters students showed a better grasp of the relations and a stronger preference than undergraduate students for a combination of searching and browsing.

During the user studies we also aimed at gathering evidence about how our intended users perceive an environment that should support a variety of information seeking behaviour and support focused access to text. The user studies we conducted, one of the very few of that type, confirmed the usability of the interface and helped in refining the set of relations presented in Chapter 3. Moreover, they showed that the model of the domain we adopted is easily grasped. We found that a user's background especially affects browsing activities. Therefore, the user's background should be carefully considered when illustrating the semantics of the relationships used, as misunderstandings are likely despite the apparent intuitiveness of the relations. Finally, the studies suggested that the appreciation of the possibility of inspecting a text from the map is independent of the user background.

As the map was intended to be an aid for human navigation, an ingredient of a bottom-up data-driven approach to information, as well as a starting point to experiment with our ideas about focused information access, it was not endowed with formal semantics. We went for a map formally under-specified, deciding for a trade-off between the possibility of imposing constraints enforceable at the level of consistency checking and a more user-friendly approach to modeling. The approach we adopted is sensible given the purpose of our work. If the possibility of automatic consistency checking should be considered for later work, the level of formality should still be kept minimal given our intended users.

The major bottleneck we found during the making of the LoLaLi map was the lack of tools to organize the authoring and editorial processes. We needed methodologies and tools to organize a workflow involving *editors* and *authors* of the map, where editors are subject experts, but untrained knowledge modelers, and authors are mainly subject experts (e.g., linguists, logicians, and computer scientists). Under this view, editors are in charge of the general planning of the map, while authors are responsible for its actual population. A person may play both roles at once, but the roles are logically distinct. Besides common editorial activities, such as deciding on terminological conventions (e.g., topic names, templates for glosses, etc.) and ensuring uniformity of style, editors should ensure balanced development of the map, and share duties among subject experts. The former activity consists in sketching the highest level topics in the map, and setting its goals, purposes, and the level of detail (granularity) required. The latter activity consists in selecting and defining "areas" to assign to authors and validating their contributions.

The activity of planning requires that some modeling language be available (similar to UML for semantic structures or ontologies) to support the modeling process and enable unambiguous communication of it. The activity of sharing duties implies the possibility of: defining fragments (modules) to assign to subject experts, checking for consistency of the new module with the rest of the map, and keeping previous versions in order to restore them when needed (versioning system).

At the time when we actively worked on the development of the LoLaLi map we found very little support from existing tools and methodologies, and this represented a real bottleneck for our work. Meanwhile, the need for this type of support has been widely acknowledged in the Semantic Web research community and a wealth of studies is currently in progress, including, for example, the two European funded projects Knowledge Web[1] and Networked Ontology[2] (NeOn), about ontologies on the web and ontologies life cycle respectively.

Concerning the life cycle of the map, we stress the importance of suitable functionalities for visualization, modularization, searching and integrity checking. All the functionalities mentioned so far should be integrated in the editorial tools, and should be adapted according to the role played by the user and to the task she has to perform.

We found that for editors it is most important to manage the global view of the map, while for authors the local view is the one most commonly used, as authors need to be able to visualize and modify individual pieces of information attached to topics. Authors may also need to visualize some (selected) branches in great detail, while only sketching other branches to provide some context (i.e., a variation of the focus-context principle). Zooming facilities should enable zooming into subgraphs and single nodes: when zooming in, the author should be able to select also a textual view, for example with paths from the root node indicated in a linear way, handy for printing. Members of the editorial board will need a global, graphical view of the complete map, for all those tasks related to the overall development of the map. In such a visualization, the entire structure would be represented as a graph where each node is visualized with a minimum set of pieces of information (for example, the title only).

7.2 Linking the Map to the Handbook

Our Research Question 3 was: "What are suitable targets in the handbook to establish focused links from topics in our browsable map?" Since our aim was to provide *focused*, topic-driven access to the handbook, we concentrated on passage retrieval techniques and compared two types of structural segmentation (by paragraphs and by

[1] URL: http://knowledgeweb.semanticweb.org/.
[2] URL: http://www.neon-project.org/.

sections) and two algorithms for semantic segmentation (TextTiling and C99) against a gold standard we created (Chapter 5). We highlighted and discussed the issue of building a corpus of annotation segments, illustrating how the writing style affects the task. We also discussed measures used for topic segmentation and found that given the nature of the text, precision and recall on topic breaks are the best, although crude, measures. We found that when only the text is considered, the best link targets for topics in the LoLaLi map are segments created by applying TextTiling and structural segmentation by single paragraphs. Actually, segmentation by paragraphs performs best in terms of recall (obviously), but TextTiling behaves (slightly) better when considering both precision and recall.

Given the target links we found, how can the ones relevant to the topics in the map be identified? (Research Question 4) We ran a second set of experiments (Chapter 6) in which we used the segments obtained in the previous experiment as a collection of documents to link to topics in the map. We used two different weighting schemas (OKAPI and tf.idf) to rank segments with respect to their similarity to a given topic. In order to evaluate these experiments we built a manually annotated reference corpus, and we used several measures for evaluation. Among them, we used three novel measures that we introduced: C-precision, to give account of the amount of relevant matter in a link target; the early onset error, to measure the amount of non-relevant content placed at the beginning of a link target; and the late onset error, to measure the amount of relevant content missed by the beginning of the link target. The two error measures give a measure of the quality of the entry point of link targets. The three measures are complementary, in that they look at different aspects of a link targets. For example, when applying C-precision one looks at the entire set of relevant paragraphs in a segment, while when measuring the onset errors one only considers the beginning of a segment and, in case of the late onset error, the paragraphs preceding its beginning (the relevant paragraphs "missed").

We found that structural segmentations by sections, and the topic segmentation performed by C99 do not constitute good options, while structural segmentation by paragraphs, and the topic segmentation performed by TextTiling do. Tuning of parameters did not affect the results. We conclude that TextTiling and segmentation by paragraphs, and possibly a combination of them, are appropriate techniques for this task.

7.3 The Bigger Picture

In this thesis we proposed that a domain map endowed with links to a text would provide focused access to the text while enabling information seeking activities. We learned that a domain map can be an informative tool, and the typed relations used in it are at the same time its strength and its weakness. If, on the one hand, they could lead non-expert users to find their way through the text and the domain, on the other hand, their understanding relies on previous knowledge of the user, which may lead to difficult interpretations. From the user studies we learned that some features of the interface need to be improved and that certain types of users can profit from the LoLaLi environment better than others.

Our conclusion is that a map of topics can be the basis for an electronic environment where a variety of information seeking behaviours are supported and topic driven access to the text is enabled. However, in our opinion the grand vision of a comprehensive domain map should be abandoned in favor of smaller maps that can either be manually built or (semi-)automatically extracted from the text. They would specifically provide access to the text and represent the fragment of domain treated in the text according to the view of the author(s). These smaller maps could then be connected to one another allowing users to navigate (i.e., browse and search) a variety of related yet independent resources in a unified way. The reasons for preferring document-based maps over domain maps are both practical and theoretical. Since in that case consistency across the entire map would not be an issue, it would be possible to access documents embodying different views of the domain in a more flexible way. Also, in case the maps were manually built, they would require fewer resources also in terms of time needed for the development. To a certain degree the folksonomy (and social tagging) phenomenon that has emerged over the past few years may be viewed as a partial instantiation of this view.

The main challenge related to the extraction of candidate links for the map was related to identifying the limits of the segments. Our work concentrated on converting a text originally intended to be linearly read, and the map was written by different people than the authors of the text—and a few years later. These factors quite naturally lead us to consider an alternative scenario where text and map are built at the same time, so that one reflects the other and vice versa. In this way the map can become an aid for writing and a general publishing model, and the problems related to the conversion of an existing text would be avoided altogether. Harmsze [2000], who took a very different research approach than ours, arrived at similar conclusions (cf. Section 2.5). We do not share Harmsze's confidence, though, that such an approach would be a viable solution to providing focused access to scientific information, nor that this is the direction that electronic publishing will (or should) take. In that scenario, in fact, when writing a new text, an author should take into account the relations "allowed" in the map. The set of relations that we took into consideration in LoLaLi should necessarily be enlarged, which would stress the tension between informativeness and information overload (both for the end user and the authors of the map) that we consider fundamental. Also previous work in the hypertext community [Baron et al., 1996, Conklin, 1987, DeRose, 1989, Smolensky et al., 1987, Trigg, 1983, Trigg and Weiser, 1986] suggests that this issue should be treated with a great deal of care. In fact, the many typologies of links proposed run the risk of overloading the reader with a large number of very detailed (and often ad hoc) definitions and restrictions, between parts of the same text, or between different texts, or between text and the annotations one may want to attach to the text (as an extreme case, consider Trigg [1983], who defines about 80 link types). If, on the other hand, the author were free to decide what relation to use and include, then the risk is to have personal semantics, somehow like in folksonomies, but without the advantages of a community effort.

These considerations make us think that the role of semantic structures like our map, or more refined ones, as an aid to writing new handbooks is limited (Research Question 5) in domains or disciplines where a standards body is absent. In our view, handbooks will continue to be written "linearly" (i.e., without heavy constraints on using sets of pre-defined relationships between topics) because this form allows the author the level of flexibility that is required to express complex and articulated thought. However, this does not imply that there is no room for improvement in the way writing tools are shaped or in the functionalities they offer. Technologies like wikis have already—and dramatically—changed our understanding of collaborative writing and rapid publishing [Miller, 2005]. Experiments with the so-called "semantic wiki" are also looking at the possibility of leveraging the search functionalities available in common wikis by defining relations between pages [SemWiki, 2007]. In particular, authoring and editorial environments will leave behind the model of electronic typewriters to be more and more plugged into the Internet and connected with other available resources. Although not of practical use for the *writing* of new handbooks, we believe semantic structures can have interesting applications for *disseminating* and *accessing* information. For this reason, we emphasize the importance of exploring tools to (semi-)automatically extract maps of documents (as opposed to maps of domains), highlight topic breaks in long documents, and, in general, refine search facilities to bring the user "below" the document level.

7.4 Directions for Future Work

The work presented in this thesis shows that the issue of providing topic driven access to scientific handbooks involves a variety of ingredients. The following directions for future work resulted from our work:

- improvement of the user interface for hierarchical structures;
- more user studies to test the proposed interface and understand the information seeking behaviour of end users in complex electronic environments;
- query-based topic segmentation techniques that are to be used on the fly and that also take into account the structure of the map; and
- evaluation of link targets, and their visualization for the purpose of navigation.

Moreover, the authors and editors of the map should be provided with appropriate tools and methodologies for development and maintenance. Finally, as mentioned in the previous section, we suggest investing in methods for the automatic extraction of maps and hierarchies of topics directly from the text.

As hierarchical structures like the LoLaLi map and ontologies are increasingly being used in a variety of domains, it is essential that appropriate user interfaces be devised for end users (i.e., people with no experience with ontologies or knowledge representation). The user interface we proposed is a good starting point that needs to be refined and improved to accommodate the feedback we collected during the user studies. Next, it is recommendable that more advanced user studies be conducted to confirm the changes made. Such a second run of user studies will also be the occasion to check how the revised organization of the map (presented in Chapter 3) is taken by real users.

As for the segmentation of text to produce link targets, we suggest experimenting with integrating text segmentation into the retrieval algorithms (as opposed to having the segmentation phase as a pre-processing). The segmentation would then be performed based on the topic to which the segment should be linked. Another approach that we consider worth investigating is performing structural segmentation based on paragraphs and aggregating paragraphs at a later stage, during or after the information ranking algorithm has been applied. Next, we need to better understand how to integrate structural information from the map into the segmentation and retrieval algorithm. Finally, the issue of evaluating link targets for the purpose of focused access deserves further study. What we did was to concentrate on the quality of a segment measured in terms of the amount of relevant content (C-precision), and in terms of *where* this relevant content is placed in the segment. We encourage further work on including into the evaluation measures a dimension of "readability" of the segments. Related to this issue is the issue of how to visualize link targets. Our observation about the level of presupposition of the text at hand suggests that the notion of what a link target for focused access is cannot be separated from the way the link is going to be visualized, navigated and read by the end user.

In the course of Section 7.1 we discussed at length the bottleneck we found during the making of the LoLaLi map. That discussion gives direction to future work needed in order to enable authors and editors to develop and maintain a map of topics, in terms of both tools and methodology. Now we add to that discussion the convenience of modularization for assigning subgraphs to authors for development: in order to do this, methods to define, select and manage *modules* from the map are needed.[3] It is also important to be able to keep control of the module during its entire life cycle, therefore a versioning system and the possibility of recovering older versions are also imperative. We also suggest that search facilities for authors of the map be improved and integrated in the environment for editing. In particular, it is worth investigating how different notions of similarity among topics can be integrated in the search facility. In fact, depending on the task at hand, two topics can be defined as being similar if they have exactly the same name, if they occupy the same position in the map, if they have

[3] It can be useful to distinguish between *modules* and *partitions* of ontologies, where the difference is whether modules overlap or not. This issue is gaining increasing attention and a number of algorithms have been put forward, see for example [dAquin et al., 2006, Noy and Musen, 2004, Seidenberg and Rector, 2006, Stuckenschmidt and Klein, 2004].

similar glosses or link to the same parts of the handbook, if they have been authored or edited by the same person, and so on.

APPENDIX A

Questionnaire for the User Studies

In this appendix we include the questionnaire used in the user studies on which we report in Chapter 4. The questionnaire is divided into 8 parts: computer literacy; information gathering strategies; browsing the interface; search and read; external links; features of the map; user preferences; wishes.

A.1 Computer Literacy

1. Which operating system do you use the most? One answer.

    - Windows
    - Macintosh/Apple
    - Linux/Unix
    - Other
    - Unsure

2. What is the highest level of formal computing training you have had?

    - none
    - some non university courses
    - college course(s)
    - some undergraduate courses in computer science
    - undergraduate degree with minor in computer science

- undergraduate degree with major in computer science
- graduate degree in computer science
- other

3. What is your current academic status?

    - undergraduate student
    - graduate student
    - other

4. How long have you been using computers?

    - never
    - less than one year
    - one to three years
    - four to six years
    - seven to nine years
    - ten years or more

5. How long have you been using the Web/Internet?

    - never
    - less than one year
    - one to three years
    - four to six years
    - seven to nine years
    - ten years or more

6. How frequently do you use the Web/Internet?

    - never
    - once or twice a year
    - monthly
    - weekly
    - daily

7. Mark the three activities that you perform more often on a computer.

    - composing a paper

- emailing
- using the Web for activities other than email
- playing games
- video-conferencing
- using spreadsheets
- using databases
- programming
- accessing entertainment (movies, music, . . .)
- doing graphic design
- video editing

A.2 Information Gathering Strategies

1. How do you normally get an overview of a new field? (More than one answer is possible: order them according to frequency)

    - textbooks
    - search Google
    - search known repositories
    - ask someone (professor, colleague,. . .)
    - other

2. How often do you use: glossaries, dictionaries, encyclopedias (on paper)?

    - Glossaries: never, yearly, monthly, weekly, daily
    - Dictionary: never, yearly, monthly, weekly, daily
    - Encyclopedias: never, yearly, monthly, weekly, daily

3. Do you use any of the above in electronic format? How often?

    - Glossaries: never, yearly, monthly, weekly, daily
    - Dictionary: never, yearly, monthly, weekly, daily
    - Encyclopedias: never, yearly, monthly, weekly, daily

4. How do you consult them?

    - on-line

- CD-ROM

5. When reading on screen, do you take notes? **Yes/No.**

6. If yes, how?

   - on separate piece of paper
   - on file
   - on the printed version
   - other

7. When studying, do you combine dictionaries and books? **Yes/No.**

8. If yes, how?

9. Do you study on other types of electronic texts? **Yes/No.**

10. If yes, what types?

11. If no, why?

A.3 Browsing the Interface

1. Browse until you get the concept **recursion theory**, knowing that it is a concept in the area of the **theory of computation**.

2. Write down or draw the fragment of hierarchy visited to get to **theory of computation**. Add as many details (i.e. concepts seen or visited) as you like.

3. Browse to find the concept **semantic ambiguity**, knowing that it is a concept in the area of linguistics.

4. Browse to find the concept **descriptive set theory**, knowing that set theory belongs to the branch of logic.

5. Browse to find the concept **Kripke semantics**, knowing that it has to do with **modal logic**.

6. Did you encounter difficulties when performing the tasks of above? If so, mention at least some of them.

7. When you click on a concept, it goes and occupies the center of the screen (the focus area) What do you think of this?

   - I like it. It helps me make a mental picture of the ontology.

- I like it, but it requires me some effort to use it.
- I do not like it but it could be useful after some practice.
- I do not like it. I think is misleading.

8. Do you find the interface clear for browsing?

   - very much
   - rather so
   - more or less
   - hardly
   - not at all

A.4 Search and Read

1. Find the concept **operator** and look at its children (i.e. subtopics). On the basis of your previous knowledge, and/or on the basis of the explanation given for those children, do you perceive some differences/similarity among them? Would you be able to draw a connection between the differences you perceive and the relation that the child concept hold with the parent? Please motivate your answer.

2. Take a look at the concepts **proof theory**, **cut elimination** and **logic**: they will appear very close to each other, on the same screen. Explain in a few words what relations you think hold between them.

3. What elements of the interface (or of your previous knowledge) made you think so?

4. By means of the information shown in the interface, mention at least two "parts of" in which **logic** (as a system of calculus or reasoning) can be divided. Mention some "subfields" (subclass) and a few relevant "mathematical results" in the area of **logic**.

5. Now think of the concepts you have mentioned and tell us how you selected them. If you were only reading the type of relations that appears below the gloss, take some time to think if what you read make sense to you and comment on that. Specify if you are using previous knowledge of the area.

6. **truth function** is a subtopic of both **truth** and **semantics**. Can you understand why?

A.5 External Links

1. Take a look at the concept **consistency**. It has a link to other sources (a handbook of logic). Take some time to inspect the links, then comment on what you have found, what you expected from the link and possibly what you would like to find instead of what has been presented to you.

2. Can you compare glosses and links to the external source?

    - I prefer glosses because. . .
    - I prefer links because. . .
    - I like both: they are useful for different things. . .
    - I do not know. . .

3. Please motivate your preference.

A.6 Features of the Map

1. Do you perceive a difference between the relationship between **linguistics** and **syntax** and **logic** and **propositional logic**? Please motivate your answer.

2. While browsing around the hierarchy, you have seen that pairs of concepts are related by one of the following relationships:

    - "is a subclass"
    - "is a part of"
    - "is a notion"
    - "is a mathematical result"
    - "is an instance"
    - "is an historical view"
    - "unspecified subtopic"

3. Do you find the types of relationships understandable?

    - very much
    - rather so
    - more or less
    - hardly
    - not at all

4. Now assign one of those relations to the following pairs of concepts. If none of them make sense to you, explain why and suggest alternatives.

    (a) **logic - modal logic**

    (b) **computer science - theory of computation**

    (c) **logic - Goedel's first incompleteness theorem**

    (d) **logic - operator**

    (e) **quantification - Frege on quantification**

    (f) **semantics - formal semantics**

5. Look at the concept **modal logic**. Is it clear to you why it has three parents?

    - very much
    - rather so
    - more or less
    - hardly
    - not at all

6. Explain in a few words why **modal logic** has three parents?

7. Put the concept **ambiguity** in the focus level. Then place the cursor, without clicking, on **semantics ambiguity**. What does appear in the focus level? Can you explain why?

A.7 User Preferences

1. Did you like browsing the map?

    - very much
    - rather so
    - more or less
    - hardly
    - not at all

2. Did you like searching the hierarchy?

    - very much
    - rather so
    - more or less

- hardly
- not at all

3. Would you express a preference between browsing and searching?

   - I strongly prefer searching over browsing.
   - I strongly prefer browsing.
   - I like using a mixture of the two.
   - I do not have particular preference/I do not know.

4. Have you found something interesting and not expected while browsing to answer the questions? What? Why did you find it interesting?

A.8 User Wishes

1. Would you like to be able to bookmark particular concepts? **Yes/No.**

   Please motivate your opinion.

2. Would you like to be able to go back and forward through the concepts already visited? **Yes/No.**

   Please motivate your opinion.

3. Would you like to have a way to select children or concepts? **Yes/No.**

   Please motivate your opinion.

4. Would you like to have a more sophisticated search? For example "search for the concept that has X as a child", or "search for the concepts that have a certain text in the gloss" or "search for all concepts that are mathematical results under logic" and so on. **Yes/No.**

   Please motivate your opinion.

5. Do you have other comments or suggestions to make the interface clearer?

APPENDIX B

Glossary

C-precision Evaluation measure for retrieved segments, introduced in Chapter 6. Let S be a retrieved segment. The C-precision of S, $C_p(S)$, is defined as the proportion of relevant paragraphs included in S:

$$C_p(S) = \frac{|Relevant\ in\ S|}{|S|}, \tag{B.1}$$

where $|S|$ is the total number of paragraphs in S.

DTD Document Type Definition [XML, 1998]. The purpose of a DTD is to define the legal building blocks of an XML document. It defines the document structure with a list of legal elements.

EoE Early onset Error. Evaluation measure for retrieved segments, introduced in Chapter 6. It measures the proportion of irrelevant material included in the segment preceding the first relevant paragraph ($r_1 - s_1 > 0$).

$$EoE = \min\left\{1, \min_{R:s_1 \leq r_1} \frac{r_1 - s_1}{|S|}\right\}, \tag{B.2}$$

LoE Late onset Error. Evaluation measure for retrieved segments, introduced in Chapter 6. It measures the *proportion of missed relevant paragraphs* at the beginning of the segment. Analogously, if $s_1 - r_1 > 0$ the LoE measures the proportion of relevant paragraphs missed at the beginning of the segment (Figure 6.2 (c)). More formally:

$$LoE = \min\left\{1, \min_{R:r_1 \leq s_1} \frac{s_1 - r_1}{|R|}\right\}, \tag{B.3}$$

where $|R|$ stands for the number of relevant paragraphs at the beginning of segment S.

TF.IDF Term Frequency by Inverse Document Frequency. Weighting schema that balances the weight coming from the number of occurrences of a term in a document, with its frequency in the entire collection [Salton and Buckley, 1988]. One way to balance these frequencies is the following:

$$w_{i,D} = tf_{i,D} * \log\left(\frac{N}{df_i}\right),$$

where $tf_{i,D}$ is the frequency of term i in document D, N is the total number of documents in the collection, and df_i is the number of documents containing the term i.

Vector Space Model In a vector space model, the similarity between a document D and a query Q is computed by means of the cosine similarity:

$$SIM(D,Q) = \frac{\sum_t (w_{t,D} * w_{t,Q})}{\sqrt{\sum w_{tD}^2 * w_{tQ}^2}}$$

where w is a weight assigned to each term (i.e., $_{t,D}$ is the weight assigned to term t in document D). The $TF.IDF$ is one of the weighting schema that can be used.

OKAPI Weighting schema that does not only considers the frequency of the query terms, but also the average length of documents in the collection and the length of the document under evaluation. It combines IDF weightings with corpus-specific sensitivities to the lengths of the document's retrieved [Robertson and Walker, 1994]. In this thesis we used the BM25 variant of Okapi, which can be expressed as follows:

$$w_i = f(tf_i) * tf_{q,i} * \log \frac{N - df_j}{df_j}$$

where:

$$f(tf_i) = \frac{(k_1 + 1)tf_i}{K + tf_i},$$

and $K = k_1((1-b) + b * \frac{dl}{avgdl})$ where dl and $avgdl$ are the document length and average document length respectively. k_1 and b are global parameters that may be tuned on the basis of evaluation data.

Precision Evaluation measure for IR. It gives the proportion of retrieved documents that are relevant:

$$Precision = \frac{|Relevant \cap Retrieved|}{|Retrieved|}$$

Recall Evaluation measure for IR. It gives the proportion of relevant documents that are retrieved:

$$Recall = \frac{|Relevant \cap Retrieved|}{|Relevant|}$$

RQL RDF Query Language [Karvounarakis et al., 2002, RQL, 2003]. RQL is a query language for RDF and RDFS that allows one to query the RDF and RDFS taken as graphs, and specify edges and nodes for retrieval. RQL showed to have some limitations, for example it does not distinguish variables and URI and does not remove duplicates in the results.

SeRQL Sesame RDF Query Language [SeRQL, 2005]. This query language for RDF and RDFS is developed by Aduna as part of Sesame [SESAME, 2005]. It is very similar to RQL, but it addresses some of limitations of RQL.

XML The Extensible Markup Language (XML) [XML, 1998] is a *metalanguage*, i.e., a language for describing other languages, which lets one design customized markup languages for limitless different types of documents. Various communities have developed their own XML, including chemistry [CML, 1997] and mathematics [W3C, 2001]. XML is content oriented, so that layout and content issues are separated. The rules of combination of the elements in an XML document can be either implicit in the document, or described in an external document (a Document Type Definition (DTD), or a Schema). The XML data structure is a tree.

RDF The Resource Description Framework (RDF) language [RDF, 1999] is a metadata-oriented language whose fundamental concepts are: resources, properties and statements. A statement is an object-attribute-value triple, which makes it especially suitable for encoding metadata. The data model of RDF is the triple, or the graph. This data model allows us to represent objects and their properties in a straightforward manner and it is actually simpler than the XML data model (that only allows strict trees of nested elements).

RDFS RDF Schema. RDF allows us to encode complex metadata graphs, but it does not specify the semantics associated with these graphs. In other words, the graph results from a collection of statements without "commitment" to a specific ontology. RDFS [RDFS, 2004] tries to fill that gap by extending the RDF data model in order to allow hierarchical organization of properties. RDFS adds to RDF the definition of *subproperties*, and the grouping of concepts into *classes*.[1]

OWL Web Ontology Language (OWL) [OWL, 2004]. The limitations of RDF and RDFS include the following: properties only have local scope, classes cannot be disjoint, nor can they be combined in a boolean way. Also, transitivity of properties is not expressible, nor is it possible to impose cardinality restrictions to property values. In practice, in order to have a total inclusion of RDFS in OWL, we should allow primitives for "the class of all classes" and for "the class of all properties," which would make the underlying reasoning problems undecidable. For these reasons, OWL was defined as three different sublanguages, with varying expressive power and, consequently, varying computational complexity. OWL Full is fully upward-compatible with RDF, both syntactically and semantically, with the drawback of being undecidable. OWL DL (where DL stands for Description Logics) restricts OWL Full in a way that is fully translatable into a description logic: it loses compatibility with RDF and RDFS, but it allows for efficient reasoning support. Finally, OWL Lite further restricts OWL DL to a sublanguage that is easy to grasp and implement, at the cost of lower expressivity. At the moment, it is considered the most promising language and the best suited to address the special needs of semantically oriented approaches to information management and retrieval.

[1] Despite the name, then, RDFschemas differ somewhat from XML schemas (such as DTD or XML Schema [XMLS, 2001], in that they do not define a permissible syntax, but operate at the semantic level and are therefore appropriate for writing ontologies.

Bibliography

ACM (1964). ACM Classification System. http://www.acm.org/class. [Cited on pages 15 and 26]

ADUNA (2005). Aduna B.V. http://www.aduna.biz. [Cited on page 54]

Aitchison, J. (1970). The thesaurofacet: A multi-purpose retrieval language tool. *Journal of Documentation*, 26:187–203. [Cited on page 18]

Allan, J. (1995). *Automatic Hypertext Construction*. PhD thesis, Cornell University, Ithaca, NY (USA). [Cited on page 86]

Allan, J., Wade, C., and Bolivar, A. (2003). Retrieval and novelty detection at the sentence level. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 314–321, New York, NY, USA. ACM Press. [Cited on page 109]

Areces, C., de Nivelle, H., and de Rijke, M. (2001). Resolution in modal, description and hybrid logic. *Journal of Logic and Computation*, 11:717–736. [Cited on page 35]

Areces, C. and Heguiabehere, J. (2002). Hylores: A hybrid logic prover based on direct resolution. In *Proceedings of Advances in Modal Logic (AiML'02), 2002.* [Cited on page 35]

Artale, A., Franconi, E., Guarino, N., and Pazzi, L. (1996). Part-whole relations in object-centered systems: An overview. *Data and Knowledge Engineering*, 20(3):347–383. [Cited on pages 33 and 34]

Baehrecke, E., Dang, N., Babaria, K., and Shneiderman, B. (2004). Visualization and analysis of microarray and gene ontology data with treemaps. *BMC Bioinformatics*, 5(1). [Cited on page 53]

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison Wesley Longman. [Cited on page 13]

Bailey, R., Allan, R., and Raiello, P. (1992). Usability testing vs. heuristic evaluation: A head-to-head comparison. In *Proc. of the Human Factors Society 36th Annual Meeting*, pages 409–413. [Cited on page 63]

Baron, L., Tague-Sutcliffe, J., Kinnucan, M. T., and Carey, T. (1996). Labeled, typed links as cues when reading hypertext documents. *Journal of the American Society for Information Science*, 47(12):896–908. [Cited on page 119]

Bederson, B., Shneiderman, B., and Wattenberg, M. (2002). Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. *ACM Transactions on Graphics (TOG)*, 21(4):833–854. [Cited on page 52]

Beeferman, D., Berger, A., and Lafferty, J. (1997). Text segmentation using exponential models. In Cardie, C. and Weischedel, R., editors, *Proc. of the 2nd Conference on Empirical Methods in Natural Language Processing*, pages 35–46. Association for Computational Linguistics, Somerset, New Jersey (USA). [Cited on page 96]

Beeferman, D., Berger, A., and Lafferty, J. D. (1999). Statistical models for text segmentation. *Machine Learning*, 34(1-3):177–210. [Cited on page 97]

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American*, 194(5):34–43. [Cited on pages 18 and 20]

Biggs, M. (1989). Information overload and information seekers: What we know about them, what to do about them. *Reference Librarian*, 25:411–429. [Cited on page 25]

Borlund, P. (2003). The concept of relevance in IR. *Journal of the American Society for Information Science and Technology*, 54(10):913–925. [Cited on page 104]

Brachman, R. (1983). What isa is and isn't. *IEEE Computer*, 16(10):30–36. [Cited on page 30]

Bruner, J. (1960). *The Process of Education*. Vintage Books. [Cited on page 9]

Buckley, C. and Voorhees, E. M. (2000). Evaluating evaluation measure stability. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 33–40, New York, NY, USA. ACM Press. [Cited on page 108]

Buckley, C. and Voorhees, E. M. (2004). Retrieval evaluation with incomplete information. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 25–32, New York, NY, USA. ACM Press. [Cited on page 109]

Cable, J. H., Ordonez, J. F., Chintalapani, G., and Plaisant, C. (2004). Project portfolio earned value management using treemaps. In *Proc. of Project Management Institute research conference*. [Cited on page 53]

Callan, J. P. (1994). Passage-level evidence in document retrieval. In *Proc. of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland*, pages 302–310. [Cited on pages 85, 86, and 87]

Caraballo, S. (2001). *Automatic Acquisition of a Hypernym-Labeled Noun Hierarchy from Text*. PhD thesis, Brown University. [Cited on page 10]

Card, S. K., MacKinlay, J. D., and Shneiderman, B., editors (1999). *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, first edition. [Cited on page 57]

Cederberg, S. and Widdows, D. (2003). Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy extraction. In *Proc. of CoNLL-2003*. [Cited on page 10]

Choi, F. (2000a). Advances in independent linear text segmentation. In *Proc. of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL-00)*, pages 26–33. [Cited on pages 84, 88, 90, 91, and 100]

Choi, F. (2000b). Linear text segmentation: approaches, advances and applications. In *Proc. of 3rd Colluquium for Computational Linguistics in the United Kingdom (CLUK3)*. [Cited on pages 84 and 90]

Cimino, J. (1998). Desiderata for controlled medical vocabularies in the twenty-first century.

Methods of Information in Medicine, 37:394–403. [Cited on page 24]

Clarke, S. G. D. (2001). Thesaural relationships. In Bean, C. and Green, R., editors, *Relationships in the Organization of Knowledge*, pages 37–52. Kluwer Academic Press. [Cited on pages 25 and 28]

Cleverdon, C. (1970). Progress in documentation, evaluation tests of information retrieval systems. *Journal of Documentation*, 26(1):55–67. [Cited on page 14]

CML (1997). XML-CML. Chemical Markup Language. http://www.xml-cml.org/ Last visited: Feb 2007. [Cited on page 133]

Cochran, W. G. (1950). The comparison of percentages in matched samples. *Biometrika*, 37:256–266. [Cited on page 92]

Cohen, J. (1960). The coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 21(1):37–46. [Cited on page 92]

Conklin, J. (1987). Hypertext: An introduction and survey. *Computer*, 20(9):17–41. [Cited on page 119]

Cooper, B. (1981). Gedanken experimentation: An alternative to traditional system testing? In Sparck Jones, K., editor, *Information Retrieval Experiments*, pages 199–209. Butterworths. [Cited on page 83]

Cooper, W. S. (1968). Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems. *American Documentation*, 19:30–41. [Cited on page 14]

Corbridge, C., Rugg, G., Major, N., Shadbolt, N., and Burton, A. (1994). Laddering: technique and tool use in knowledge acquisition. *Knowledge Acquisition*, 6(3):315–341. [Cited on page 26]

Cruse, A. D. (1986). *Lexical Semantics*. Cambridge University Press. [Cited on pages 30, 31, and 33]

Cruse, A. D. (2002). Hyponymy and its varieties. In Green, R., Bean, C. A., and Myaeng, S. H., editors, *The Semantics of Relationships. An Interdisciplinary Perspective*. Kluwer Academic Press. [Cited on pages 30, 31, and 32]

Cruse, D. A. (1994). Prototype theory and lexical relations. *Rivista di Linguistica*, 6(2):167–188. [Cited on page 32]

de Vries, A. P., Kazai, G., and Lalmas, M. (2004). Tolerance to irrelevance: A user-effort oriented evaluation of retrieval systems without predefined retrieval unit. In *Recherche d'Informations Assistee par Ordinateur (RIAO 2004)*. [Cited on pages 14 and 105]

DeRose, S. J. (1989). Expanding the notion of links. In *Proc. of Hypertext'99*, pages 249–257. [Cited on page 119]

Dervin, B. (1983). An overview of sense-making: concepts, methods, and results to date. Paper delivered at the International Communication Association Annual Meeting. Dallas, Texas. [Cited on page 1]

dAquin, M., Sabou, M., , and Motta, E. (2006). Modularization: a key for the dynamic selection of relevant knowledge components. In *In Proc. of the ISWC 2006 Workshop on Modular Ontologies*. [Cited on page 121]

Eisenberg, M. and Small, R. (1993). Information-based education: an investigation of the nature and role of information attributes in education. *Information Processing & Management*, 29(2):263–275. [Cited on page 25]

Ernst, N., Storey, M.-A., Allen, P., and Musen, M. (2003). Addressing cognitive requirements in knowledge engineering with jambalaya. In *Visualizing Information in Knowledge Engi-*

neering (VIKE'03). Workshop at the 2nd Int. Conf. on Knowledge Capture, Sanibel Island, FL. [Cited on page 49]

EUWN (2005). Eurowordnet. http://www.illc.uva.nl/EuroWordNet. [Cited on page 19]

Fahlman, S. E. (1979). *NETL: A system for representing and using real-world knowledge*. MIT Press. [Cited on page 38]

FAO (2007). Agrovoc thesaurus. http://www.fao.org/aims/ag_intro.htm. [Cited on page 28]

Fellbaum, C., editor (1998). *WordNet, an Electronic Lexical Database*. MIT Press. [Cited on pages 27 and 36]

Flash (2005). http://www.macromedia.com/. Accessed July 2005. [Cited on page 58]

Fluit, C., Sabou, M., and van Harmelen, F. (2003). Supporting user tasks through visualisation of light-weight ontologies. In Staab, S. and Studer, R., editors, *Handbook on Ontologies in Information Systems*. Springer-Verlag. [Cited on page 54]

Friedrich, C. and Eades, P. (2002). Graph drawing in motion. *Journal of Graph Algorithms and Applications*, 6(3):353–370. [Cited on page 61]

Fuhr, N., Lalmas, M., Malik, S., and Kazai, G., editors (2006). *Advances in XML Information Retrieval and Evaluation, 4th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2005, Dagstuhl Castle, Germany, November 28-30, 2005, Revised Selected Papers*, volume 3977 of *Lecture Notes in Computer Science*. Springer. [Cited on page 86]

Gennari, J. H., Musen, M. A., Fergerson, R. W., Grosso, W. E., Crubézy, M., Eriksson, H., Noy, N. F., and Tu, S. W. (2003). The evolution of Protégé: an environment for knowledge-based systems development. *Int. J. Hum.-Comput. Stud.*, 58(1):89–123. [Cited on page 44]

Girju, R., A., B., and D., M. (2003). Learning semantic constraints for the automatic discovery of part-whole relations. In *Proc. of HLT-2003*. [Cited on page 10]

GRAPHVIZ (2007). Graphviz - Graphical Visualization Software. http://www.graphviz.org. [Cited on page 27]

Grootendorst, R., Henkemans, F. S., van Eemeren, F. H., and Grootendorst, R. (1996). *Fundamentals of Argumentation Theory: A Handbook of Historical Backgrounds and Contemporary Developments*. Erlbaum. [Cited on page 21]

Grossman, D. and Frieder, O. (2003). *Information Retrieval: Algorithms and Heuristics*. Kluwer Academic Publishers. [Cited on page 13]

Grosz, B. J. and Sidner, C. L. (1986). Attention, intentions and the structure of discourse. *Computational Linguistics*, 12(3):175–204. [Cited on page 88]

Gruber, T. R. (1992). Ontolingua: A mechanism to support portable ontologies. Technical report, Knowledge Systems Laboratory, Stanford University. [Cited on page 33]

Gruber, T. R. (1993). A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199–220. [Cited on page 20]

Guarino, N. and Welty, C. (2000). Ontological analysis of taxonomic relationships. In Lender, A. and Storey, V., editors, *Proc. of International Conference on Conceptual Modelling*. Springer-Verlag LNCS. [Cited on page 32]

Guarino, N. and Welty, C. (2002a). Evaluating ontological decisions with Ontoclean. *Communications of the ACM*, 45(2):61–65. [Cited on page 32]

Guarino, N. and Welty, C. (2002b). Identity and subsumption. In Green, R., Bean, C. A., and Myaeng, S. H., editors, *The Semantics of Relationships. An Interdisciplinary Perspective*. Kluwer. [Cited on page 32]

Harmsze, F. (2000). *A Modular Structure for Scientific Articles in an Electronic Environment*. PhD thesis, University van Amsterdam. Available at: http://dare.uva.nl/record/78293. [Cited on pages 20, 21, and 119]

Hearst, M. A. (1994a). *Context and Structure in Automated Full-text Information Access*. PhD thesis, University of California at Berkeley. [Cited on pages 86 and 88]

Hearst, M. A. (1994b). Multi-paragraph segmentation of expository text. In *Proc. of the 32nd Meeting of the Association for Computational Linguistics, Los Cruces, NM*. [Cited on pages 85, 88, 89, 91, and 92]

Hearst, M. A. (1997). TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages. *Computational Linguistics*, 23(1):33–64. [Cited on page 84]

Hearst, M. A. and Plaunt, C. (1993). Subtopic structuring for full-lenght document access. In *Proc. of the 16th Annual International ACM SIGIR Conference on Research and Development in IR*, pages 59–68. [Cited on pages 87 and 88]

Hutchins, W. (1977). On the problem of "aboutness" in document analysis. *Journal of Informatics*, 1(1):17–35. [Cited on pages 83 and 104]

IEEE (1995). *IEEE Colloquium on Information Overload*. http://www.ieeexplore.ieee.org/xpl/tocresult.jsp?isNumber=10765. [Cited on page 25]

INEX (2005). Initiative for the evaluation of XML retrieval. http://inex.is.informatik.uni-duisburg.de/. [Cited on page 13]

Informatik, M. P. I. (2006). Spass. http://www.spass.mpi-sb.mpg.de/. [Cited on page 35]

Ingwersen, P. (1984). Psychological aspects of information retrieval. *Social Science Information Studies,*, 4:83–89. [Cited on page 1]

Ingwersen, P. (1992). *Informaton Retrieval Interaction*. Taylor Graham. Available at: http://www.db.dk/pi. [Cited on page 14]

Inhelder, B. and Piaget, J. (1964). *The Early Growth of Logic in the Child: Classification and Serialization*. Routledge & Kegan Paul. [Cited on page 33]

ISAVIZ (2004). Isaviz: A visual authoring tool for rdf. http://www.w3.org/2001/11/IsaViz/. [Cited on page 49]

ISO:2788 (1986). Documentation - guidelines for the establishment and development of monolingual thesauri. [Cited on page 18]

ISO:IND (1985). ISO 5963:1985. documentation - Methods for examining documents, determining their subjects, and selecting indexing terms. http://www.nlc-bnc.ca/iso/tc46sc9/standard/5963e.htm. [Cited on page 18]

ISO:MLT (1985). ISO 5964:1985. Documentation - Guidelines for the establishment and development of multilingual thesauri. http://www.nlc-bnc.ca/iso/tc46sc9/standard/5964e.htm. [Cited on page 18]

IWN (2005). Hindi and marathi wordnet. http://www.cfilt.iitb.ac.in/. [Cited on page 19]

Jeffries, R., Miller, J. R., Wharton, C., and Uyeda, K. (1991). User interface evaluation in the real world: A comparison of four techniques. In *Proc. of ACM CHI'91*, pages 119–124. ACM Press. [Cited on page 63]

Johnson, B. and Shneiderman, B. (1991). Treemaps: a space-filling approach to the visualization of hierarchical information structures. In *Proc. of the 2nd International IEEE Visualization Conference*, pages 284–291. Also in Sparks of Innovation in Human-Computer Interaction, Shneiderman, B., Ed., Ablex (June 1993) 309-322. Reprinted in Readings in Information Visualization: Using Vision to Think, by Stuart K. Card, Jock D. MacKinlay, Ben Shneiderman. Morgan Kaufmann; 1st edition (January 25, 1999). [Cited on page 52]

Joyce, T. and Needham, R. (1958). The thesaurus approach to information retrieval. *American Documentation*, 9:192–197. Reprinted in *Readings in Information Retrieval*. K. Sparck Jones and P. Willet (eds.). [Cited on page 18]

Kamps, J. and Marx, M. (2002). Words with attitude. In *In Proceedings of the 1st International Conference on Global WordNet*, pages 332–341. [Cited on page 19]

Karvounarakis, G., Alexaki, S., Christophides, V., Plexousakis, D., and Scholl, M. (2002). Rql: A Declarative Query Language for RDF. In *The Eleventh International World Wide Web Conference (WWW'02)*. [Cited on pages 46 and 133]

Kaszkiel, M. and Zobel, J. (1997). Passage retrieval revisited. In *Proc. of SIGIR 97*, pages 178–185. [Cited on pages 84, 85, and 87]

Kircz, J. and Harmsze, F. (2000). Modular scenarios in the electronic age. In van der Vet en P. de Bra (eds.), P., editor, *Proceedings Conferentie Informatiewetenschap 2000*, pages 31–43. http://www.kra.nl/Website/Artikelen/mod2k.html. [Cited on page 20]

Kircz, J. G. (1998). Modularity: the next form of scientific information presentation? *Journal of Documentation*, 54(2):210–235. [Cited on page 20]

Klavans, J. L., McKeown, K., Kan, M.-Y., and Lee, S. (1998). Resources for the evaluation of summarization techniques. In *Proc. of the 1st International Conference on Language Resources and Evaluation, Grenada, Spain*. [Cited on pages 91 and 92]

Knublauch, H., Fergerson, R., Noy, N. F., and Musen, M. A. (2004). The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. In *Proc. of ICSW04*. Springer. [Cited on page 45]

Lamping, J., Rao, R., and Pirolli, P. (1995). A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proc. ACM Conf. Human Factors in Computing Systems, CHI*, pages 401–408. ACM. [Cited on page 51]

Lee, G. G., Lee, S., Jung, H., Cho, B.-H., Lee, C., Kwak, B.-K., Cha, J., Kim, D., An, J., Seo, J., Kim, H., and Kim, K. (2001). SiteQ: Engineering high performance QA system using lexico-semantic pattern matching and shallow NLP. In *Proc of TREC 10*. [Cited on page 87]

Lesk, M. and Salton, G. (1969). Relevance assessment and retrieval system evaluation. *Information Storage and Retrieval*, 4:343–359. [Cited on page 92]

Lewis, J. (1994). Sample sizes for usability studies: Additional considerations. *Human Factors*, 36:368–378. [Cited on page 63]

Light, M., Mann, G., Riloff, E., and Breck, E. (2001). Analyses for elucidating current question answering technology. *Journal of Natural Language Engineering. Special issue on Question Answering*. [Cited on page 87]

Likert (1932). A technique for the measurement of attitudes. *Archive of Psychology*, 140:5–55. [Cited on page 62]

Liu, X. and Croft, W. (2002). Passage retrieval based on language models. In *Proc. 11th International Conference on Information and Knowledge ¡anagement (CIKM)*, pages 375–382. [Cited on page 84]

Llopis, F. and Vicedo, J. L. (2001). IRn, a passage retrieval system from University of Alicante, at Clef2001. In *Proc. of CLEF 2001*. [Cited on page 87]

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM journal of Research and Development*, 2(2):159–165. [Cited on page 15]

Mani, I. and Maybury, M. T., editors (1999). *Advances in Automatic Text Summarization*. MIT

Press. [Cited on page 85]

Mann, W. C. and Thompson, S. A. (1987). Rhetorical structure theory: A theory of text organization. Technical Report ISI/RS 87–190, ISI. [Cited on page 88]

Marchionini, G. (1995). *Information seeking in electronic environments*. Cambridge University Press. [Cited on page 1]

Maybury, M., editor (2004). *New Directions in Question Answering*. AAAI Press. [Cited on page 13]

McGuinness, D. L. (2002). Ontologies come of age. In Fense, D., Hendler, J., Lieberman, H., and Wahlster, W., editors, *Spinning the Semantic Web; Bringing the World Wide Web to Its Full Potential*. MIT Press. [Cited on page 20]

MESH (2007). Medical subject headings. `http://www.nlm.nih.gov/mesh/MBrowser.html`. [Cited on page 55]

Miller, G. (1995). Wordnet: A lexical database for English. *CACM*, 38(11):39–41. [Cited on page 19]

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63:81–97. [Cited on page 19]

Miller, N. (2005). Wikipedia and the disappearing "Author". *ETC: A Review of General Semantics*, 62(1):37–40. [Cited on page 120]

Mizzaro, S. (1997). Relevance: The whole history. *Journal of the American Society of Information Science*, 48(9):810–832. [Cited on page 104]

Monz, C. (2003). *From Document Retrieval to Question Answering*. PhD thesis, University of Amsterdam. [Cited on page 85]

Morton, T. (1999). Using coreference for question answering. In *ACL Workshop, Coreference and Its Applications*. [Cited on page 88]

MULTIWN (2005). Multiwordnet. `http://multiwordnet.itc.it/english/home.php`. [Cited on page 19]

Muskens, R., van Benthem, J., and Visser, A. (1997). *Handbook of Logic and Language*, chapter Dynamics. Elsevier. [Cited on pages 93 and 103]

MySQL (2007). Mysql. `http://www.mysql.com`. [Cited on page 58]

Nakatani, C., Grosz, B., Ahn, D., and Hirschberg, J. (1995). Instruction for annotating discourses. Technical Report TR-21-95, Center for Research in Computing Technology, Harvard University. [Cited on page 94]

Nielsen, J. (1989). Usability engineering at a discount. In *Proc. of the third International Conference on Human-Computer Interaction*, pages 1–8. [Cited on page 62]

Nielsen, J. (2000). Why you only need to test with 5 users. alertbox. `http://www.useit.com/alertbox/20000319.html`. [Cited on page 63]

Nielsen, J. and Landauer, T. K. (1993). A mathematical model of the finding of usability problems. In *Proceedings of ACM INTERCHI'93 Conference*, pages 206–213. [Cited on page 63]

Noy, N. and Musen, M. (2004). Specifying ontology views by traversal. In *In Proc. of the International Semantic Web Conference (ISWC)*. [Cited on page 121]

Noy, N. F., Fergerson, R. W., and Musen, M. A. (2000). The knowledge model of Protégé: Combining interoperability and flexibility. In *Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management*, pages 17–32. Springer-Verlag. [Cited on page 44]

Noy, N. F. and McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology. Technical report, Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880. [Cited on page 26]

Noy, N. F., Sintek, M., Decker, S., Crubźy, M., Fergerson, R. W., and Musen, M. A. (2001). Creating semantic web contents with Protégé-2000. *IEEE Intelligent Systems*, 16(2):60–71. [Cited on page 44]

Oberle, D., Volz, R., Motik, B., and Staab, S. (2004). An extensible ontology software environment. In Staab, S. and Studer, R., editors, *Handbook on Ontologies*, pages 311–333. Springer. [Cited on page 49]

O'Neill, M. and Denos, M. (1992). Practical approach to the stereo matching of urban imagery. *Image and Vision Computing*, 10(2):89–98. [Cited on page 90]

Owen, J. M. (2002). The new dissemination of knowledge: digital libraries and institutional roles in scholarly publishing. *Journal of economic methodology*, 9(3):275–288. [Not cited]

OWL (2004). Web ontology language (OWL). W3C Recommendation. http://www.w3.org/2004/OWL. [Cited on pages 20 and 134]

Passonneau, R. and Litman, D. J. (1993). Intention-based segmentation: human reliability and correlation with linguistic cues. In *Proc. of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 148–155. [Cited on page 92]

Peterson, T. (1994). *Introduction to the Art and Architecture Thesaurus*. Oxford University Press. http://shiva.pub.getty.edu. [Cited on page 18]

Plaisant, C., Maryland, U. O., Lukehart, C., Schiro, D., and Ryan, J. (2003). Using visualization tools to gain insight into your data. In *Society of Petroleum Engineers (SPE) Annual Conference and Exhibition*. [Cited on page 54]

Plumb Design (2004). Visual Thesaurus. http://www.visualthesaurus.com/. [Cited on page 52]

Ponte, J. M. and Croft, W. B. (1997). Text segmentation by topic. In *European Conference on Digital Libraries*, pages 113–125. [Cited on page 88]

Purchase, H. (1997). Which aesthetic has the greatest effect on human understanding? In Battista, G. D., editor, *Graph Drawing '97*, volume 1353 of *Lecture Notes in Computer Science*, pages 248–261. Springer-Verlag. [Cited on page 51]

Rabiner, L. W. and Schafer, R. W. (1978). *Digital processing of speech signals*. Prentice-Hall, Inc. [Cited on page 89]

Ragetli, J. (2001). Towards concept-based structuring of electronic information. Master's thesis, Master of Logic, University of Amsterdam. [Cited on pages 26 and 43]

RDF (1999). Rdf resource description framework. W3C Recommentations. http://www.w3.org/RDF/. [Cited on page 133]

RDFS (2004). RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendations. http://www.w3.org/TR/rdf-schema/. [Cited on pages 20, 58, and 133]

Reid, J., Lalmas, M., Finesilver, K., and Hertzum, M. (2006). Best entry points for structured document retrieval: Parts I & II. *Information Processing and Management*, 42:74–105. [Cited on page 12]

Reynar, J. C. (1998). *Topic Segmentation: Algorithms and Applications*. PhD thesis, University of Pennsylvania. [Cited on pages 88, 90, and 97]

Roark, B. and Charniak, E. (1998). Noun-phrase co-occurrence statistics for semiautomatic semantic lexicon construction. In *Proceedings of the 17th international conference on Com-*

putational linguistics, pages 1110–1116, Morristown, NJ, USA. Association for Computational Linguistics. [Cited on page 10]

Robertson, G., Mackinlay, J., and Card, S. (1991). Cone trees: Animated 3D visualizations of hierarchical information. In Robertson, S., Olson, G., and Olson, J., editors, *Proc. ACM Computer-Human Interaction '91 Conference on Human Factors in Computing Systems*, pages 189–194. ACM Press. [Cited on page 55]

Robertson, S. E. and Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 232–241. Springer-Verlag New York, Inc. [Cited on pages 109 and 132]

Robins, D. (2000). Interactive Information Retrieval: Context and basic notions. *Informing Science Journal*, 3(2):57–62. [Cited on page 14]

Roget, P. and Davidson, G., editors (2002). *Roget's Thesaurus of English Words and Phrases*. Penguin. First edition 1852. http://thesaurus.reference.com/roget/. [Cited on page 15]

Rosenstiehl, P., Allder, J.-A., and Carrington, D. (2001). User preference of graph layout aesthetics: A UML study. In Marks, J., editor, *Graph Drawing '00*, volume 1984 of *Lecture Notes in Computer Science*, pages 5–18. Springer-Verlag. [Cited on page 51]

RQL (2003). http://139.91.183.30:9090/RDF/RQL/. [Cited on page 133]

Salton, G. (1968). *Automatic Information Organization and Retrieval*. McGraw-Hill. [Cited on page 18]

Salton, G., Allan, J., and Buckley, C. (1993a). Approaches to Passage Retrieval in Full Text Information Systems. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–58. [Cited on page 13]

Salton, G., Allan, J., and Buckley, C. (1993b). Approaches to passage retrieval in full text information systems. In *Proc. of the 16th Annual International ACM/SIGIR Conference, Pittsburgh, USA*, pages 49–58. [Cited on pages 84 and 85]

Salton, G., Allan, J., and Singhal, A. (1996a). Automatic text decomposition and structuring. *Information Processing and Management*, 32(2):127–138. [Cited on page 88]

Salton, G. and Buckley, C. (1988). Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24:513–523. Reprinted in *Readings in Information Retrieval*, K.Sparck Jones and P.Willet (editors). [Cited on page 132]

Salton, G. and Buckley, C. (1991). Global text matching for information retrieval. *Science*, 253:1012–1015. [Cited on page 86]

Salton, G., Singhal, A., Buckley, C., and Mitra, M. (1996b). Automatic text decomposition using text segments and text themes. In *Proc. of the 7th ACM Conference on Hypertext, Washington D.C., USA*. [Cited on page 88]

Salustri, F. A. (1998). Ontological commitments in knowledge-based design software: A progress report. In Finger, S., Tomiyama, T., and Mantyla, M., editors, *Proc. 3rd IFIP TC5 WG5.2 Workshop on Knowledge Intensive CAD*, pages 41–72. Kluwer Academic Publishers. [Cited on page 33]

Saracevic, T. (1975). Relevance: A review of and a framework for the thinking on the notion in information science. *Journal of the American Society for Information Science*, 26:321–343. [Cited on page 104]

Sayyadian, M., Shakery, A., Doan, A., and Zhai, C. (2004). Toward entity retrieval over structured and text data. In *SIGIR 2004 Workshop on the Integration of Information Retrieval and Databases (WIRD'04)*. [Cited on page 13]

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proc. of International Conference on New Methods in Language Processing*. [Cited on page 109]

Schreiber, G., Akkermans, H., Anjewierden, A., deHoog, R., Shadbolt, N., VandeVelde, W., and Wielinga, B. (2000). *Knowledge Engineering and Management: The CommonKADS Methodology*. MIT Press. [Cited on page 26]

Seidenberg, J. and Rector, A. (2006). Web ontology segmentation: Analysis, classification and use. In *In Proc. 15th of the World Wide Web Conference (WWW)*. [Cited on page 121]

SemWiki (2007). Semantic Wiki. http://www.semwiki.org/. [Cited on page 120]

SeRQL (2005). http://www.openrdf.org/doc/users/ch06.html. [Cited on pages 46 and 133]

SESAME (2005). http://openrdf.org. [Cited on pages 46, 58, and 133]

Simons, P., editor (1987). *Parts: a Study in Ontology*. Clarendon Press, Oxford. [Cited on page 33]

Sintek, M. (2003). Ontoviz tab: Visualizing protg ontologies. http://protege.stanford.edu/plugins/ontoviz/ontoviz.html. [Cited on page 49]

Skorochod'ko, E. (1972). Adaptive method of automatic abstracting and indexing. *Information Processing*, 71:1179–1182. [Cited on page 88]

Smolensky, P., Bell, B., Fox, B., King, R., and Lewis, C. (1987). Constraint-based hypertext for argumentation. In *Proc. of Hypertext '87*, pages 215–245. ACM Press. [Cited on page 119]

Snow, R., Jurafsky, D., and Ng, A. Y. (2005). Learning syntactic patterns for automatic hypernym discovery. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *NIPS 17*. [Cited on page 10]

Sowa, J. F. (1991). *Principles of Semantic Networks*. Morgan Kaufmann Publishers, Inc. [Cited on page 18]

Spärck Jones, K. (2004). What's new about the Semantic Web? some questions. *SIGIR Forum*, 38(2). [Cited on page 27]

Srzednicki, J. T., Rickey, V. F., and Czelakowski, J., editors (1984). *Lesniewski's Systems: Ontology and Mereology*. Nijhoff International Philosophy Series 13. Martinus Nijhoff Publishers, The Hague. [Cited on page 33]

Stanley, A. and Clipsham, P. (1997). Information overload - myth or reality? In *IEE Colloquium Digest(340). Proc. of the 1997 IEE Colloquium on Information Technology Strategies for Information Overload*. [Cited on page 25]

Stokes, N., Carthy, J., and Smeaton, A. F. (2002). Segmenting broadcast news streams using lexical chaining. In Vidal, T. and Liberatore, P., editors, *Proc. of STAIRS 2002*, volume 1, pages 145–154. IOS Press. [Cited on page 87]

Storey, M., Noy, N., Musen, M., Best, C., and Fergerson, R. (2002). Jambalaya: an interactive environment for exploring ontologies. In *Proc. of International Conference on Intelligent User Interfaces*, page 239. [Cited on page 49]

Storey, V. (1993). Understanding semantic relationships. *Very Large Databases Journal*, 2(4):455–488. [Cited on pages 25, 32, and 78]

Stuckenschmidt, J. and Klein, M. (2004). Structure-based partitioning of of large concept hierarchies. In *Proc. of the Third International Semantic Web Conference (ISWC)*. [Cited on page 121]

TOPICMAPS (2006). Xml topic maps specifications. http://www.topicmaps.org/xtm/1.0/. [Cited on page 3]

Trigg, R. (1983). *A network approach to text handling for the online scientifc community*. PhD thesis, University of Maryland. [Cited on page 119]

Trigg, R. H. and Weiser, M. (1986). Textnet: a network-based approach to text handling. *ACM Trans. Inf. Syst.*, 4(1):1–23. [Cited on page 119]

Tufte, E. R. (1988). *The Visual Display of Quantitative Information*. Graphics Press. [Cited on page 61]

Tutte, W. T. (1963). How to draw a graph. In *Proc. London Math. Society*, volume 13, pages 743–768. [Cited on page 51]

Tversky, B. (1990). Where partonomies and taxonomies meet. In *Meaning and Prototypes: Studies in Linguistic Categorization*, pages 334–344. Routledge. [Cited on page 33]

van Benthem, J. and ter Meulen, A., editors (1997). *Handbook of Logic and Language*. Elsevier. [Cited on pages 3 and 115]

van Eijck, J. and Kamp, H. (1997). *Handbook of Logic and Language*, chapter Representing Discourse in Context. Elsevier. [Cited on pages 93 and 103]

van Hage, W. R. (2004). Living on the Edge. Master's thesis, University of Amsterdam. [Cited on pages 22, 46, 57, and 109]

van Harmelen, F. and Antoniou, G. (2004). *A Semantic Web Primer*. MIT Press. [Cited on page 20]

van Schie, M. (2003). Using semantic web technologies for electronic publishing. Master's thesis, Free University Amsterdam. [Cited on page 44]

Vickery, B. C. (1997). Ontologies. *Journal of Information Sciences*, 23(4):277–286. [Cited on page 20]

Virzi, R. A. (1992). Refining the test phase of usability evaluation: how many subjects is enough? *Hum. Factors*, 34(4):457–468. [Cited on page 63]

Voorhees, E. M. (1994). Query expansion using lexical-semantic relations. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 61–69. Springer-Verlag New York, Inc. [Cited on pages 18 and 19]

Vossen, P., Hofmann, K., de Rijke, M., Tjong Kim Sang, E., and Deschacht, K. (2007). The Cornetto database: Architecture and user-scenarios. In Moens, M.-F., Tuytelaars, T., and de Vries, A., editors, *Proceedings DIR 2007*, pages 89–96. [Cited on page 19]

W3C (2001). MathML. http://www.w3.org/Math/. [Cited on page 133]

W3C (2006). W3c on the semantic web. http://www.w3.org/2001/sw/. [Cited on page 20]

Webber, B. (2004). D-LTAG: extending lexicalized tag to discourse. *Cognitive Science*, 28:751–779. [Cited on page 88]

Wiegand, W. A. (1998). The "Amherst Method": the Origins of the Dewey Decimal Classification Scheme. *Libraries & Cultures*. [Cited on page 15]

Wilkinson, R. (1994). Effective retrieval of structured documents. In *Proc. of SIGIR 1994*, pages 311–317. [Cited on pages 14, 84, and 86]

Wilson, P. (1995). Unused relevant information in research and development. *Journal of the American Society for Information Science*, 46(1):45–51. [Cited on page 25]

Wilson, T. D. (1994). Information needs and uses: fifty years of progress? In Vickery, B., editor, *Fifty yeaars of information progress: a Journal of Documentation Review*, pages 15–51. Aslib, London. [Cited on page 1]

Wilson, T. D. (1997). Information behaviour: an interdisciplinary perspective. *Inf. Process. Manage.*, 33(4):551–572. [Cited on page 1]

Wilson, T. D. (1999). Models in information behaviour research. *Journal of Documentation*, 55(3):249–270. [Cited on page 2]

Wilson, T. D. (2000). Recent trends in user studies: action research and qualitative methods. *Information Research*, 5(3). http://informationr.net/ir/5-3/paper76.html. [Cited on page 61]

XML (1998). EXtensible Markup Language (XML). W3C Recommendation. http://www.w3.org/XML/. Third Edition, 2004. [Cited on pages 43, 131, and 133]

XML (2001). XML Linking Language (XLink). Version 1.0. http://www.w3.org/TR/xlink/. [Cited on page 44]

XMLS (2001). XML Schema. W3C Recommendations. http://www.w3.org/XML/Schema. [Cited on page 134]

Yaari, Y. (1997). Segmentation of expository text by hierarchical agglomerative clustering. In *Proc. of the Conference on Recent Advances in Natural Language Processing*, pages 59–65. [Cited on page 88]

Samenvatting

"Handbooks" zijn dikke delen, die door verschillend auteurs worden volgeschreven met tientallen pagina's lange hoofdstukken. Het doel is een omvattend overzicht te geven van een wetenschappelijke vakgebied. Helaas ontbreekt er vaak een duidelijke gestandaardiseerde structuur. In de internationale wetenschappelijke literatuur zijn deze *Handbooks* niet zomaar gelijk te stellen aan het meer encyclopedische Nederlandse handboek. Gebruikers van een wetenschappelijke *Handbooks* kunnen een specifieke informatiebehoefte hebben, wat hen er toe brengt een specifiek onderwerp of zelfs een specifiek aspect daarvan te willen lezen. In het laatste geval wil de lezer liever niet door tientallen pagina's bladeren, maar zou graag direct naar die tekstdelen van het boek "springen," die het gezochte aspect behandelen. Dit noemen we gefocusseerde toegang tot een tekst. In andere gevallen kunnen *Handbook* gebruikers vagere informatiebehoeften hebben die meer gerelateerd zijn aan de noodzaak of behoefte om een bredere en algemener overzicht van het vakgebied te krijgen.[2]

Vanuit de behoefte aan gefocusseerde toegang is het begrijpelijk om te denken aan een overzichtskaart van het vakgebied. Een afzoekbare kaart met zowel de verschillende onderwerpen en hun relaties, alsmede de relevante plaatsen in de tekst waar een en ander behandeld wordt. De gebruiker kan dan door zoeken of door navigatie op een plek in de kaart aankomen en vandaar inzoomen op het onderwerp dat het beste toegesneden is op haar informatiebehoefte. Daarom stellen we ons een kaart voor van het vakgebied dat niet alleen referenties bevat naar de passende locaties in de tekst maar ook de relaties weergeeft tussen de verschillende informatie elementen onderling—we noemen deze *topics*. Onze vraagstelling is nu: Wat kunnen we winnen voor het toegankelijk maken van een *Handbook* als wij een kaart aanbieden die verwijzingen, links, bevat naar het boekwerk? Dit proefschrift heeft tot doel om deze kwestie te beantwoorden. Daartoe splitsen we de vraagstelling in kleinere onderzoeksvragen.

[2]Op deze plek beschouwen we niet het geval van het volgordelijk doorlezen van het hele boek of een hoofdstuk daaruit.

In Hoofdstuk 1 introduceren wij het onderwerp en definiëren wij de onderzoeksvragen.

In Hoofdstuk 2 ontwikkelen wij onze visie op gefocusseerde toegang tot wetenschappelijke *Handbooks*. Wij nemen ons voor om een afzoekbare kaart te maken, die de belangrijke termen en hun relaties weergeeft alsmede een passende interface is naar een elektronisch in plaats an papieren Handbooks. Ook presenteren wij hier de algemene achtergrond van het zoeken naar en binnen boeken en over de structuren die voor indexering- en classificatiedoelen nodig zijn.

In Hoofdstuk 3 benoemen wij onze eerste onderzoeksvraag. Dit is de vraag naar de vereisten waaraan een kaart moet voldoen die geschikt is voor menselijk bladeren en tevens een geraamte vormt voor gefocusseerde toegang tot een wetenschappelijk Handbook. Op basis van deze vereisten, passen wij het zo verkregen model toe op ons onderzoekscorpus: het domein van logica en taal. Wij presenteren dan onze kaart die wij de *Logic and Language Links*, of afgekort LoLaLi kaart noemen. Wij gaan in hoe zij tot stand is gekomen en wat haar onderdelen zijn, inclusief de hiërarchische en niet-hiërarchische relaties tussen topics. Als corpus voor ons onderzoek nemen wij het *Handbook of Logic & Language* onder redactie van Van Benthem en Ter Meulen.

In Hoofdstuk 4 behandelen we de tweede onderzoeksvraag: hoe presenteren we deze kaart aan lezers? Om te beginnen geven wij een overzicht van de bekende visualisatietechnieken voor semantische structuren zoals de onze. Daarna beschrijven wij de interface die wij ontwikkeld hebben om het eindgebruikers mogelijk te maken om binnen de LoLaLi kaart te navigeren en te zoeken. Wij sluiten af met een rapportage van gebruikersstudies die wij onder twee groepen studenten uitvoerden.

In Hoofdstuk 5 gaan we verder met onderzoeksvraag drie. Wat zijn de passende tekstdoelen in het *Handbook* om daadwerkelijk gefocusseerd en topic bepaald koppelingen te maken tussen de kaart en de tekst? Wij concentreren ons op de selectie van tekstexcerpten die door middel van twee soorten geautomatiseerde tekst segmentatietechnieken geselecteerd worden. De ene manier is structuur, de andere semantisch georiënteerd. De resultaten vergelijken wij met een handmatige annotatie van het onderhavige *Handbook*.

Gegeven de verzameling tekstexcerpten die wij in het vorige hoofdstuk hebben verkregen gaan wij in Hoofdstuk 6 na in hoeverre deze tekstdelen passen bij de topics van de kaart. Dit is onze vierde onderzoeksvraag. Wij bespreken de evaluatie van dit proces en komen met voorstellen hoe zoiets aangepakt moet worden.

Uiteindelijk trekken wij in Hoofdstuk 7 onze conclusies en belichten richtingen voor verder onderzoek. Ons werk laat zien dat als wij dit *Handbook* voorzien met een domeinkaart, wij een alternatief pad om de tekst te benaderen hebben vormgegeven die door onze doelgroep gewaardeerd wordt.

Wij hebben geleerd van dit onderzoek dat een domeinkaart een informatief stuk gereedschap kan zijn en dat benoemde relaties zoals wij die gebruiken zowel sterke als zwakke kanten hebben. Aan de ene kant kunnen zij niet-expert gebruikers door de tekst en het vakgebied leiden, aan de andere kant hangt het begrip af van actuele kennis van de gebruiker wat kan leiden tot lastige interpretaties. De gebruikersstudies leerden ons dat sommige kenmerken van de interface verbeterd moeten worden en dat sommige soorten gebruikers meer profijt hebben van de LoLaLi omgeving dan anderen.

Onze conclusie is dat een *topics*-kaart de basis kan vormen voor een elektronische omgeving waar een variëteit van informatiezoekgedrag ondersteund wordt en topic gestuurde toegang tot een tekst mogelijk wordt gemaakt. Naar onze mening zal echter het idee van een alles omvattende domeinkaart los moeten worden gelaten ten bate van beperktere kaarten die hetzij handmatig, hetzij (semi)automatisch geëxtraheerd worden uit de onderhavige tekst. Deze kaarten zouden specifiek toegang tot de tekst moeten bewerkstelligen en het onderwerpsdomein representeren volgens de interpretatie van de auteur(s). Deze kleiner kaarten kunnen dan op een dusdanige wijze aan elkaar verbonden worden dat gebruikers op een eenduidige manier kunnen navigeren, d.w.z. bladeren en zoeken, in een verzameling gerelateerde, maar onderling onafhankelijke bronnen.