INTELLIGENT SUPPORT FOR
KNOWLEDGE CAPTURE AND CONSTRUCTION

Ana Gabriela Maguitman

Submitted to the faculty of the Graduate School

in partial fulfillment of the requirements

for the degree

Doctor of Philosophy

in the Department of Computer Science,

Indiana University

December 2004

Accepted by the Graduate Faculty, Indiana University, in partial fulfillment of the requirements of the degree of Doctor of Philosophy.

Doctoral
Committee

David B. Leake, Ph.D.
(Principal Advisor)

Michael E. Gasser, Ph.D.

Filippo Menczer, Ph.D.

December 2, 2004

Lawrence S. Moss, Ph.D.

To Ignacio.

Acknowledgements

I want to express my gratitude to my advisor, David Leake, for his support and guidance throughout these years. David has constantly encouraged my work and I have learned a lot from his expertise as a researcher. Having the opportunity to collaborate with him has been a privilege and a very rewarding experience.

I would also like to thank the other members of my doctoral committee. Mike Gasser has been a great teacher and has given me helpful advice since my first days at IU. Filippo Menczer has been very involved in the technical aspects of my research. I am incredibly fortunate to have had the opportunity to discuss my work with him and to get his insightful feedback. Larry Moss has been one of the kindest and most open-minded professors I have ever met. I have truly enjoyed his logic classes and I have learned a great deal from him.

I am grateful to the faculty members of the Computer Science Department, especially to Randall Bramley, Andrew Hanson, Douglas Hofstadter, Daniel Leivant, Paul Purdom, Gregory Rawlins and Dirk Van Gucht for being outstanding professors; to the administrative staff, especially to Sherry Kay, Lucy Battersby, Jill Elaine Cowden, Deb Mart, Kelly Whitehead, Linda Barchet, Nancy Garrett, and Laura Reed for making everything easier; and to the systems staff, especially to Lynne Ann Crohn, Rob Henderson, Lloyd T. Jones, Bryce Himebaugh, and Shing-Shong Shei for being always ready to help and knowing how to.

I am very grateful to Alberto Cañas and the CmapTools team at the Institute for Human and Machine Cognition. Having the opportunity to collaborate with them made my research experience much richer and enjoyable. In particular, I am very thankful to Raúl Saavedra for his hospitality during my visits to Pensacola and for helping me to get familiar with the CmapTools project.

I wish to thank the members of the NaN group for their feedback on my work and for stimulating intellectual interchange. I am thankful to all the students at the Cranium lab for being excellent officemates. Very special thanks to my colleague and friend Thomas Reichherzer for a fruitful collaboration. I have enjoyed working with Thomas, and much of the work presented in this dissertation is the result of our long and constructive discussions. Sofia Brenes and Alejandro Valerio have been wonderful friends and it has been a pleasure to have them around with their contagious positive attitude.

A special thanks goes to Stella Kafka and Maricarmen Martínez for these five years of close friendship, and to many other colleagues and friends I had the chance to meet in Bloomington: Saleh Aliyari, Steve Bogaerts, Stefano Borgo, Anastasiya Chagrova, Hamid Ekbia, Sarah Ellis, Fulya Erdinc, Kate Holden, Tei Laine, Francisco Lara-Dammer, Hayoung Lee, Seunghwan Lee, Roussanka Loukanova, Waki Murayama, Dipanwita Sarkar, Shakila Shayan, Steve Simms, Aleksander Slominski, Raja Sooriamurthi, Nik Swoboda, and David Wilson.

I am thankful to the Computer Science Department and Mathematics Department at my home university, Universidad Nacional del Sur, for providing me with invaluable skills. I am especially grateful to my Argentinean advisor, Guillermo Simari, who initiated me into research, encouraged me to pursue a doctoral degree, and helped me in many ways to accomplish my goals. I also wish to thank Carlos Chesñevar and Fernando Tohme for their constant encouragement and friendship.

My parents have been my unconditional supporters in all my endeavors and have taught me by example the importance of perseverance and effort. My family and friends in Bahía Blanca have always been in touch, to cheer me up and to remind me that home will always be there. I am grateful to all of them.

More than anyone else, I thank Ignacio Viglizzo, my husband and best friend, for sharing all these years and for brightening my life.

This research was supported by the NASA Intelligent Systems Program under award No. NCC 2-1216. A teaching assistantship from the Computer Science Department and a Fulbright/FOMEC fellowship provided financial support during my first years at Indiana University. I gratefully acknowledge Google for the use of their Web API service.

Abstract

In traditional views of knowledge management, knowledge capture is seen as primarily knowledge acquisition, capturing knowledge that already exists within the expert. This thesis proposes an alternative approach, "knowledge extension," based on the premise that a knowledge model evolves from coordinated processes of knowledge acquisition and knowledge construction. In this view, it is crucial to support experts' construction of new knowledge as they extend existing knowledge models. This dissertation develops and evaluates artificial intelligence methods to facilitate knowledge extension, especially in the context of knowledge modeling via concept mapping. The problem of supporting knowledge extension raises two research questions: First, how can topic descriptors be algorithmically extracted from concept maps, and second, how to use these topic descriptors to identify candidate topics on the Web with the right balance of novelty and relevance. To address these questions, this thesis develops the theoretical framework required for a "topic suggester" to aid information search in the context of a knowledge model under construction. Finally, it describes and evaluates EXTENDER, an implemented support tool based on this framework. The proposed algorithms have been developed and tested within the framework of CmapTools, a widely-used system for supporting knowledge modeling using concept maps. However, their generality makes them applicable to a broad class of knowledge modeling systems, and to Web search in general.

Contents

List of Tables

List of Figures

1

Introduction

The topic of this dissertation research is intelligent support for human-centered knowledge modeling. Knowledge modeling is the process of representing a body of knowledge so that this knowledge can then be shared and exploited. Knowledge acquisition has long been considered to be a bottleneck in the development of knowledge-based systems [Hayes-Roth et al., 1983]. In recent years the knowledge acquisition bottleneck has been alleviated somewhat by the development of knowledge modeling tools that allow experts to enter descriptions of their expertise without the intervention of knowledge engineers (e.g., [Gil, 1994, Blythe et al., 2001, Aiken and Sleeman, 2003]), but it remains a fundamental problem.

The difficulty of encoding knowledge depends largely on the representation. For the most part, approaches to knowledge representation have followed the logicist tradition and have been based on rigorous specification languages. These languages are usually non-ambiguous and straightforward to process by algorithms but present a technical barrier for knowledge-modelers unfamiliar with these formalisms.

To capture data using these languages knowledge engineers or human programmers need to meditate between the expert and the system. This approach to capture knowledge gives rise to the famous “expert and knowledge engineer communication problem.” In addition, the expert is typically isolated from the knowledge modeling interface and only after the knowledge has been hand-crafted by the knowledge engineer is the representation ready to be manipulated by users and programs. Consequently, any direct interaction between expert and system as the model evolves is usually precluded. As we will study in more detail throughout this work, in-progress knowledge models can be used to characterize information requirements and to search for new useful material. Therefore, in order to benefit from suggestions that the system may be able to generate, it is highly desirable that the expert rather than the knowledge engineer be the one in charge of entering the information into the knowledge base.

In this dissertation we propose a “knowledge extension” approach to knowledge management, based on the premise that a knowledge model evolves from coordinated processes of knowledge acquisition and knowledge construction. In this view, it is crucial that the language used by the experts for entering their knowledge descriptions be one with which they feel comfortable. The use of natural languages may appear as a good choice for experts to directly enter descriptions of their knowledge [Iwanska and Shapiro, 2000]. However, automatic processing of knowledge models remains important because it is valuable for knowledge-acquisition tools to interact with users to reuse and adapt existing resources, rather than forcing them to build knowledge models from scratch. As a consequence, the use of natural language to encode knowledge would be impractical due to the fact that the extraction of concepts and relations from unstructured text is a very difficult process to be done automatically by a machine.

This dissertation research studies intelligent support methods to aid human-centered knowledge capture and reuse. *Our pragmatic goal is to develop effective mechanisms that unobtrusively assist the user in the knowledge modeling activity. Our research goal is to develop and study algorithms to make this possible.*

1.1 Concept Mapping for Knowledge Modeling

An intermediate approach to represent knowledge is to choose a method more structured than natural language but more flexible than a rigorous formalism. Concept maps, developed by Joseph D. Novak in the ’70s [Novak, 1977], are good candidates for providing a representation for knowledge models that is practical for experts to build. Concept maps are collections of simplified natural language sentences displayed as a two-dimensional, visually-based representation of concepts and their relationship. In concept maps, concepts are depicted as labeled nodes, and relations between concepts as labeled links. Figure 1.1 shows an example of a concept map. Concept mapping techniques have aided people of different ages to examine many fields of knowledge. They offer the flexibility of natural language but have the advantage of inducing their creators to organize their knowledge in a structured fashion, where concepts and their connections can be directly recognized. Because concept maps are rich in structure, they are more easily understood by other humans and more tractable for automated systems than plain text. In addition, electronic concept maps are elegant, browsable and sharable, making them an effective vehicle for aiding human understanding.

An initiative is now under way to support knowledge modeling by means of concept maps. CmapTools, developed by the Institute for Human and Machine Cognition (IHMC), is a suite of publicly-available software tools for knowledge acquisition, construction, and sharing [Cañas et al., 2004] based on concept maps. The CmapTools system is being used by people of all ages, from elementary school children to NASA Scientists. More important, experts are able to construct knowledge models of their domain without the need for a knowledge engineer’s intervention, or to actively participate in the knowledge elicitation if a knowledge engineer leads the process.

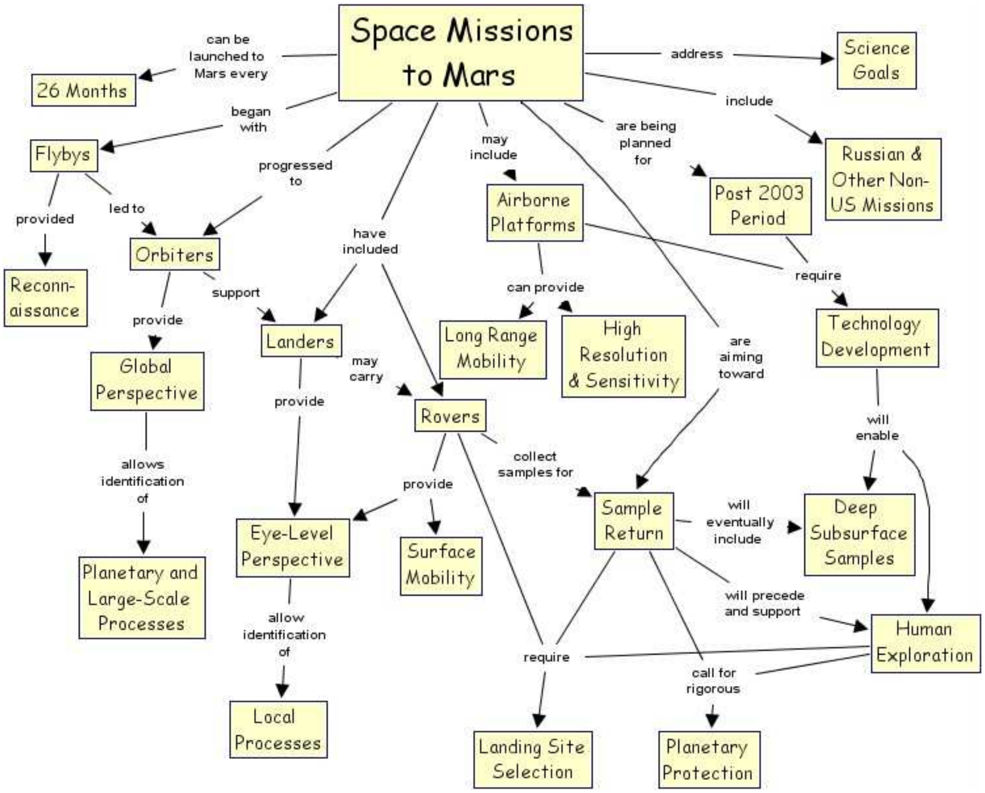


Figure 1.1: A concept map created by a NASA expert.

1.2 Issues and Research Questions

It has been noticed that when experts and ordinary users employ knowledge modeling tools, they often stop for significant amounts of time, wondering how to extend their models. In some cases, they search through existing libraries to discover previously captured knowledge and resources that can be integrated into their models. In other cases, they search through the Web looking for new material and ideas to enhance their in-progress representations. This search activity could be done more effectively if mechanisms for information access and delivery were included as part of the knowledge modeling tools.

To support knowledge modeling in CmapTools, we are developing a number of intelligent aides. These systems take as their starting point a concept map under construction, and propose information to aid the user's knowledge capture and knowledge construction by proactively suggesting relevant concept maps, propositions, resources, concepts and topics. These suggesters are described in detail in section 2.3 and in [Leake et al., 2003b].

In this dissertation we develop and study methods that use information automatically extracted from the current knowledge model to guide mining the Web to identify and suggest novel but relevant topics, for possible inclusion in the knowledge model. Topics are commonly defined as pieces of data that have been grouped together as a result of having a common theme. As opposed to manually constructed topics selected in light of a particular theme, the topics generated by our techniques result from automatic processes involving Web mining and clustering. Hence, we refer to them as *artificial topics*. Artificial topics are first presented to the user as suggestions consisting of a small collection of terms. These suggestions include, for each topic, a ranked list of constituent Web pages together with their descriptions and URLs. This method helps the user to extend the knowledge model beyond information that has already been captured. This approach is implemented by EXTENDER (EXtensive Topic Extender from New Data Exploring Relationships) within the framework of CmapTools [Leake et al., 2003a, Leake et al., 2003b].

Searching the Web to support knowledge extension presents new challenges unaddressed by classical IR techniques. As a consequence, intelligent support for knowledge extension needs to develop its own solutions to several issues. The design of the EXTENDER system gives rise to the following research questions:

Research Question One: How can topic descriptions be algorithmically extracted from non-standardized structured knowledge representations such as concept maps?

Research Question Two: How can those topic descriptors be used to characterize information requirements and to discover novel but relevant topics of potential interest that the user may want to include in the knowledge model?

This work addresses the above questions by formulating a number of associated hypotheses, developing algorithms based on those hypotheses and analyzing them empirically. The proposed algorithms have been developed within the framework of CmapTools. However, their generality makes them applicable to a broad class of knowledge modeling systems, and to Web search in general.

1.3 Overview of Proposed Techniques and Contributions

In the following we outline a collection of techniques proposed to address the research questions and we postulate the hypotheses investigated in our work.

Processing Non-Standardized Structured Representations

The first question we want to address in this work is how to extract topic descriptions from non-standardized representations like concept maps in such a way that we can take advantage of both the content and the structure of the maps.

From a data-processing perspective, concept maps present an important advantage over purely textual forms in at least two respects: (1) in concept maps, concepts and their relationships are readily available, and (2) concept maps are usually hierarchical and have a rich topology. Because concept maps are typically hierarchical and have a rich topology, we have examined the question of whether topological factors are useful to predict the descriptive power of a concept. We claim that topological analysis algorithms can be applied to the analysis of concept maps to describe the relative arrangements of their concepts, and the topological roles of concepts in the map can be usefully summarized according to a small set of dimensions [Cañas et al., 2001].

We developed and reported three candidate models for predicting the importance of concepts in concept maps [Leake et al., 2004a]. These models use the topology of concept maps to compute a weight predicting each concept's importance in describing the topic of a map. To determine which factors to include in the models, we first considered factors from the concept mapping literature. Novak proposes that meaningful learning is facilitated when new concepts or concept meanings are subsumed under broader, more inclusive concepts, which suggests that concept maps should have a hierarchical structure. The suggested models can reflect such a structure, with weightings reflecting that more descriptive concepts are at the top of the map, and less descriptive at the bottom. Our models associate with each concept a weight reflecting its descriptive power. Once these weights are computed they remain static unless the topology of the concept map changes.

The motivations for the topological analysis of concept maps can be summarized by the following hypotheses:

1. Concepts that are closer to the root of a concept map are considered better descriptors of the topic of the map.

2. Concepts with higher connectivity are considered better descriptors of the topic of the map.

By investigating these hypotheses we obtained empirical data to guide the design of techniques for the effective analysis of concept maps. Techniques based on topological analysis help to describe concept maps in terms of their most important concepts. These descriptions are applied in our implementation of mechanisms to search the Web for relevant topics. Section 3.3 describes the topological analysis models in detail.

Automatic Context-Based Topic Search

The process of searching for on-line data can be guided by diverse objectives. There are essential differences between searching for information to fulfill *consultation needs* and searching for material to support *knowledge extension*. Usually, the purpose of consultation is to find specific answers for specific questions. On the other hand, when searching for material to support knowledge extension, rather than a specific question there are usually several implicit questions and a task that to a certain extent is still open and needs to be completed.

Typical interfaces for querying electronic document collections have been designed to serve the purpose of fulfilling consultation needs (e.g., finding information with a Web search engine) rather than the purpose of supporting knowledge modeling. To search using these interfaces the user needs to know what to seek and has to be able to explicitly state his search request as a list of keywords. In some cases the list of keywords turns out to be too specific, resulting in very few results, while in others it is too general, resulting in extremely large amounts of unclassified information. In the former case, the user tends to reformulate the query, while in the latter the user typically browses through a good part of the results until the relevant information is finally reached.

Several studies have shown the benefits of having tools that provide assistance for query formulation (and reformulation) and for filtering results (e.g. [Greenberg, 1998, Chui, 2002]). A number of systems have been implemented to support query refinement (e.g. [Chen and Dhar, 1990, Vélez et al., 1997, Anick and Tipirneni, 1999, Oyama et al., 2001]) and several others that facilitate topic exploration by clustering search results into topically-coherent groups (e.g. [Cutting et al., 1992, Hearst and Pedersen, 1996, Anick and Vaithyanathan, 1997, Kaski et al., 1998, Zamir and Etzioni, 1999, Chen and Dumais, 2000]). These systems provide a browsing interface where the user intervention must be explicit.

The burden implied by the need to explicitly formulate search requests can be alleviated if queries are produced automatically [Rhodes and Starner, 1996, Budzik and Hammond, 1999]. In our task, the knowledge model under construction provides a rich body of contextual information that can be usefully exploited to guide retrievals. We are developing methods that take advantage of that information to produce queries that are presented to a Web search engine. Because conventional Web search engines limit queries to a small number of words, and knowledge models may contain numerous terms, selecting useful terms is crucial. While not all the information contained in a knowledge model can be summarized in a query, effective mechanisms can be designed to extract small sets of representative terms to construct queries. The returned results can be contrasted against the knowledge model under construction to filter noise and irrelevant data. In addition, the search context can be used to recognize which terms are the best descriptors of the topic at hand (i.e., which are the terms that best describe the present search context to a user) and which are the best discriminators (i.e., which are the most useful query-terms). We claim that topic descriptors can be obtained either by applying topological analysis directly to a concept map, or dynamically, by searching for terms that tend to occur *often* in documents similar to the map. On the other hand, topic discriminators can be extracted dynamically, by searching for terms that tend to occur *only* in documents similar to the topic at hand. To evaluate these claims, we tested the following two hypotheses:

3. Good topic descriptors can be found by looking for terms that occur often in documents similar to the given topic, i.e., human assessments of term importance in a topic are in good correspondence with this notion of term descriptive power.

4. Good topic discriminators can be found by looking for terms that occur only in documents similar to the given topic, i.e., queries constructed with terms dynamically selected in light of this notion result in better precision than the one achieved by static feature selection methods.

Techniques for the dynamic extraction of topic descriptors and discriminators are useful in the implementation of the EXTENDER system. Our basic approach is to use descriptors and discriminators automatically extracted from the topic of the current map to guide querying a Web search engine for relevant information. Differently from conventional approaches for querying the Web, search requests are not treated in isolation but in the context of a knowledge modeling task.

Another characteristic of classical information retrieval systems is that they attempt to match requests with the most similar documents. A few approaches take a different position [Budzik et al., 2000, Smyth and McClave, 2001] and postulate that in certain circumstances conventional notions of similarity may not be the best criteria for retrieval. In particular, when the purpose of the search is to bring material to extend knowledge coverage on certain domain, the criteria for determining usefulness should not be restricted to similarity. Since knowledge models are usually intended to include a rich variety of related topics, attaining novelty and diversity may be as important, or even more important, than attaining similarity.

We propose the use of an algorithm that starts from a knowledge model under construction and generates queries at incremental distances from the set of terms that originated the request. As a starting point, the search context is defined using the knowledge model under construction, and is progressively refreshed as the system moves its focus through a connected series of topics. Cohesive topics are generated by clustering the results returned by the Web search process. Irrelevant information is filtered by contrasting the search results with the search context. Our algorithm uses a temperature or curiosity mechanism to favor diversity at the beginning of the search and focus during the final stages. This mechanism has a fundamental role in defining the constraints imposed by the search context, as well as in the process of recombining old keywords with novel keywords to generate new topics. After a few iterations the process yields a final collection of topics, which the system presents as suggestions to the user. We claim that the implementation of this algorithm results in the retrieval of novel material, but still connected to the originating set of terms. An important question is how to evaluate a topic generation strategy. Traditional information retrieval schemes are evaluated by computing precision and recall on a pre-defined collection. We use global coherence and coverage (to be defined later), as generalizations of the notions of precision and recall. In addition, novelty will play an important role in our evaluations. Since we do not know how many relevant topics for a specific concept map exist on the Web we use a knowledge model consisting of concept maps on a particular domain as the collection of relevant topics. We investigated the following hypotheses:

5. Using the search context to maintain the relationship between the set of generated topics and the initial concept map helps to preserve global coherence, ensuring that the system maintains its focus on topics relevant to the initial concept map.

6. The use of a curiosity mechanism to incrementally search the Web increases novelty and coverage compared to a baseline mechanism that generate the same number of queries directly from the originating knowledge model.

The performance of our algorithm heavily relies on the selection of good parameters for setting permissible degrees of exploration and exploitation. By performing evaluations addressing the above hypotheses we gathered data for guiding the design of effective techniques as well as for assessing the competence of the EXTENDER system.

Contributions

This research provides the following contributions:

1. Methods for extracting topic descriptions from non-standardized structured representations such as concept maps.

2. Methods for dynamically extracting topic descriptors and discriminators from unstructured text-based data.

3. Methods that use in-progress knowledge models as a starting point to search the Web in order to discover novel but relevant topics.

4. Empirical data assessing the value of these methods.

5. A prototype tool to support human-centered knowledge extension built on these methods.

Overall our research contributes new perspectives and solutions to the problem of knowledge modeling via non-standardized structured representations and establishes a base for further studies of the topic.

1.4 Road Map

The roadmap for this thesis is as follows:

- This chapter states the problem addressed by this thesis. It formulates two research questions, postulates six associated hypotheses and outlines the contributions of this work.

- Chapter 2 discusses general perspectives on knowledge modeling, tracing the historical evolution of knowledge acquisition tools. It presents the CmapTools system and its accompanying knowledge elicitation methodology. It then reviews work on computer-mediated support systems, with special focus on intelligent aides and suggester systems. The chapter closes with an overview of a set of intelligent support tools implemented as part of CmapTools.

- Chapter 3 briefly discusses descriptive theories of human knowledge organization and reviews some existing schemes for externalizing knowledge. The chapter discusses the advantages of using concept maps as external representations of an individual cognitive structure and presents three models for assessing concept descriptive power in concept maps.

- Chapter 4 begins with a discussion of classical approaches to information retrieval and their limitations when applied to the problem of context-based topic search on the Web. It then reviews work on Web mining and topic extraction that relates to this thesis. After this review, it develops a theoretical framework addressing the problems of query formation and topic identification in the context of a knowledge model under construction.

- Chapter 5 describes how the framework developed in the previous chapter is applied in the implementation of the EXTENDER system.

- Chapter 6 focuses on the empirical analysis of the thesis' hypotheses. It describes three experimental studies for the evaluation of the methods and algorithms proposed in chapters 3, 4 and 5.

- Finally, chapter 7 concludes by summarizing the thesis' results, discussing the applicability of the proposed methods to a broader class of tasks, and outlining areas of future research work.

2

Knowledge Modeling Support

2.1 Perspectives on Knowledge Modeling

Knowledge modeling is the process of representing a body of knowledge to enable subsequent systematic access and sharing. Traditional methodologies to knowledge modeling are costly because they require time-consuming knowledge elicitation, with a knowledge engineer mediating between the expert and the system. The need for a knowledge engineer as an intermediary is in part due to the representation schemes used to model expert knowledge, which are usually inadequate to be used directly by experts.

There have been two major trends to represent knowledge, commonly typified as computer-centered or human-centered. The primary purposes of traditional knowledge acquisition tools have been to build expert systems and to facilitate knowledge sharing by software agents. As a consequence, classical approaches to knowledge representation have been computer-centered and followed the logicist tradition initiated by John McCarthy (1959) . Examples include semantic networks [Quillian, 1968], frame systems [Minksy, 1975], scripts [Schank and Abelson, 1977], conceptual graphs [Sowa, 1984], and description logic systems

[Brachman and Schmolze, 1985, Levesque and Brachman, 1987]. These representation languages are usually non-ambiguous and straightforward to process by algorithms but present a technical barrier for knowledge-modelers unfamiliar with these formalisms.

Some of the more recent work on knowledge modeling has marked a change in perspective, addressing the importance of creating knowledge bases that are natural to share and process by people rather than by software systems. Human-centered representation languages have been used with the purpose of allowing people to enter descriptions of their knowledge using a medium with which they feel comfortable. A few frameworks suggest the use of natural language not just as an interface but also as a knowledge representation medium (e.g., [Iwanska and Shapiro, 2000]). Others propose the use of sketching (e.g., [Forbus and Usher, 2002]), a human-centered knowledge modeling technique that results in visually and conceptually rich representations. Among the human-centered representation techniques is concept mapping [Novak, 1977, Novak and Gowin, 1984], the knowledge modeling scheme we have adopted.

A different dimension under which we can analyze the existing approaches to knowledge modeling is based on the procedure used for capturing knowledge. Knowledge acquisition has been recognized as the bottleneck in the activity of constructing knowledge-based systems [Hayes-Roth et al., 1983]. As a consequence, much of the knowledge modeling research has focused on the process rather than the result of knowledge modeling. This gave rise to different tools that facilitate knowledge editing, both for the situations when the resulting representation is computer-centered as well as when it is human-centered.

In the following section we trace the evolution of knowledge acquisition systems from the mid-70's, when we see the first efforts to enable domain experts to enter knowledge to a knowledge based system themselves, to the present, when knowledge acquisition tools are based on well-established methodologies stemming from the fields of social sciences, artificial intelligence and cognitive sciences. Then, in section 2.1 we discuss concept mapping as a vehicle for human-knowledge representation and the CmapTools system, which provides an easy-to-use interface for knowledge capture, extension, and examination.

The Evolution of Knowledge Acquisition Tools

Traditional approaches to knowledge acquisition involve knowledge engineers or human programmers as mediators between the expert and the system, resulting in many shortcomings, such as the "expert knowledge engineer communication problem." This communication problem is the result of a large gap between the expert and the knowledge engineer's views on the problem solving process and the absence of a common vocabulary. To overcome this problem, several tools were proposed for capturing knowledge directly from experts, without the knowledge engineer as an intermediary.

Instruction Systems

Efforts to enable experts to enter descriptions of their knowledge to the system themselves led to the development of knowledge acquisition tools known as instruction systems. During the early years these tools acted mostly as interfaces, where the users entered descriptions of their knowledge using statements in a restricted form of natural language. The instruction system was in charge of translating the statements into a formal internal representation. An example of this approach is illustrated by TEIRESIAS [Davis, 1979, Davis, 1982], a component of the MYCIN diagnostic expert system [Shortliffe, 1976]. TEIRESIAS employed meta-knowledge to formulate expectations about what other domain knowledge might be needed and used a dialog interface to elicit knowledge from the expert. Another example of an early instruction system is illustrated by KAS [Duda et al., 1979], the knowledge acquisition component of the PROSPECTOR geologist expert system. Starting from an initial representation of the domain based on a semantic network, KAS attempted to find errors, such as disconnected parts of the network, and generated questions to the expert with the purpose of completing the model. EXPERT [Weiss and Kulikowski, 1979] is another early instruction system where the user statements needed to be entered in the form of simple rules using a text editor.

Second-Generation Architectures

The first generation of instruction systems resulted in poorly structured knowledge-based systems. This was in part due to users not providing knowledge with a high degree of precision and the system's inability to distinguish the roles of different kinds of knowledge entered by the users. To overcome this problem subsequent systems incorporated knowledge about the world and became capable of *knowledge-level* communication. The knowledge-level [Newell, 1982] or epistemological-level provides a means to interact with a system at a level independent of underlying representation and implementation issues and to "rationalize" the behavior of the system. Programs that interacted with the expert at the knowledge-level engaged in highly structured dialogues with the purpose of constructing complete and coherent domain models. Examples of these systems are NEOMYCIN [Clancey, 1981, Clancey, 1983, Hasling et al., 1984], EES [Neches et al., 1985, Swartout et al., 1991], ROGET [Bennett, 1985], MOLE [Eshelman, 1988], OPAL [Musen et al., 1988] and SALT [Marcus and McDermott, 1989],

Support for Skeletal Model Construction

The main goal of the second generation of instruction systems was to facilitate model instantiation, compilation and refinement, but they were designed to work around predefined skeletal models, which imposed rigid requirements for the resulting representation. Later knowledge acquisition systems were able to provide greater flexibility by allowing users to construct skeletal models or customized ontologies. These tools supported this task by offering graphical editing facilities or libraries of components. An initial attempt to provide support for skeletal model construction is illustrated by PROTÉGÉ [Musen, 1989], the first of a generation of meta-tools developed by the Knowledge Modeling group at Stanford Informatics. The PROTÉGÉ system is an environment for knowledge-based systems that operates at the meta-level by generating domain specific knowledge acquisition applications.

Reusing Problem-Solving Knowledge

Several problem-solving methods (PSMs) were used repeatedly in a variety of knowledge-based system, offering opportunities to exploit reusability. The initiative for capturing PSMs as a special form of expertise knowledge and constructing libraries to reuse that kind of knowledge goes back to work on Heuristic Classification [Clancey, 1984, Clancey, 1985], and Generic Tasks [Chandrasekaran, 1983, Chandrasekaran, 1986]. PROTÉGÉ-II [Puerta et al., 1992] is an attempt to generalize PROTÉGÉ by providing facilities to deal with multiple PSMs. The trend to facilitate component reusability progressed with the development of several other knowledge acquisition frameworks. Among those aimed at reusing problem-solving knowledge we can distinguish EXPECT [Gil, 1994, Blythe et al., 2001]. EXPECT is a knowledge acquisition system that has the capability of storing the rationale for each piece of knowledge the system captures. Problem-solving knowledge is reused to generate "expectations" about the domain knowledge that needs to be entered. The system uses an internal representation language based on the description logic formalism but provides an interface that supports knowledge entry by non-programmer users.

Applying Methodologies to Support Knowledge Acquisition

Many principled methodologies derived from social sciences and psychological theories gave rise to a range of knowledge acquisition tools. Theories of situated actions [Suchman, 1987] and tool perspectives [Norman, 1991] gave rise to the development of the Human Interface Tool Suite [Terveen and Wroblewski, 1990] usually referred to as HITS. HITS incorporates a collaborative editor called HKE, which has been used as an interface to the CYC knowledge base [Lenat et al., 1990]. The editor requires users to be familiar with the basics of CYC terminology but incorporates AI technology, like rule-based critics and collaborative manipulation, to provide a human centered knowledge acquisition environment.

The psychological theory of personal constructs [Kelly, 1955] originated a knowledge acquisition methodology known as repertory grid. This methodology aims at gaining insight into the expert's mental model of the problem domain. It is implemented as an iterative process, where the expert is expected to name important objects in the domain and systematically identify characteristics of the objects and their importance. This data is captured in a grid, which the expert iteratively refines by adding or modifying entries. ETS [Boose, 1985], AQUINAS [Boose and Bradshaw, 1987] and KSS0 [Gaines and Shaw, 1993] are knowledge acquisition tools based on this methodology.

The Knowledge Analysis and Design Support (KADS) scheme [Schreiber and Wielinga, 1993] is a sophisticated methodology for reusing both domain knowledge and problem-solving knowledge. The KADS approach, usually promoted as Common KADS [de Hoog et al., 1993], suggests that the knowledge acquisition activity can be characterized in terms of multiple models, namely organization model, agent model, task model, expertise model, communication model, and design model. In the mean time, each model has a special structure, augmented with internal and external relations. Instances of tools that support KADS methodology are Shelley [Anjewierden et al., 1992], KADS Tool and Open KADS Tool [Kingston, 1995]. Another knowledge modeling methodology based on the ideas of reusing domain knowledge and problem-solving methods is COMMET [Steels, 1990], which stands for COMponential METhodology. COMMET is simpler than KADS and is supported by the KREST workbench, which provides a graphical environment to assist the reusability of components and the implementation of knowledge-based applications by non-programmers.

Ontologies

Frameworks aimed at reusing domain knowledge have centered mostly on the construction of standardized representations. The knowledge modeling community has long been concerned with devising ontologies as formal specifications that machines can understand and process [Gruber, 1993]. Recently, with the growing attention to the development of a Semantic Web [Berners-Lee, 1998, Berners-Lee et al., 2001], research on ontology design has become much more active.

Ontology construction is a tedious process; therefore systems have been built to expedite the design of ontologies and to facilitate sharing and integration of different frameworks. An example of a system that facilitates distributed, collaborative development of ontologies is the ONTOLINGUA server [Farquhar et al., 1997]. This system uses an extended version of the Ontolingua language [Gruber, 1992], which supports both semi-formal definitions and formal specifications. Others environments that facilitate ontology sharing include RiboWeb [Altman et al., 1999], Community Web Portals [Staab et al., 2000], and OntoShare [Davies et al., 2003].

A noteworthy work that includes support for ontology construction is illustrated by CODE4 [Skuce and Lethbridge, 1995], a graphical knowledge acquisition system that combines ideas from frame-based systems, object-oriented systems, and hypertext systems. A main assumption underlying CODE4's design is that "most users will want to represent largely informal knowledge and will rarely need or benefit from formal syntax and semantics, but these should be available if needed." Therefore, its main concern is to facilitate flexible knowledge representation. In particular, it provides support for certain natural language-related problems. Moreover, the system offers features for incrementally adding formal syntax and semantics.

Another instance of graphical knowledge browser and editor that facilitates the construction of ontologies is GKB [Paley et al., 1997]. The most salient feature of this system is its generality and portability across several frame knowledge representation systems. PROTÉGÉ-2000 [Noy et al., 2000] is another instance of the PROTÉGÉ family. It provides a graphical environment for ontology-development and knowledge acquisition. SHAKEN [Barker et al., 2001, Clark et al., 2001] is a human-centered tool for domain knowledge capture that represents the world in term of events, entities and relationships. Events and entities integrate a library of reusable components. Although components are stored as first-order logic descriptions, SHAKEN provides a graphical interface that can be manipulated by subject matter experts, without the mediation of knowledge engineers. Other tools for ontology edition are OILEd [Bechhofer et al., 2001], WebODE [Arpírez et al., 2001] and OntoEdit [Sure et al., 2002].

These research directions emphasize the need for human-centered knowledge modeling tools that facilitate knowledge construction, access, and re-application. In the next section we describe CmapTools, a human-centered knowledge modeling system that has received widespread use for knowledge modeling by experts and novices.

Concept Mapping and the IHMC CmapTools

Concept mapping, developed by Novak for use in education, was designed as a vehicle for making cognitive structures explicit by externalizing the concepts and propositions known to a person [Novak and Gowin, 1984], but the process of concept mapping is also viewed as a means to aid people in constructing meaningful knowledge, by organizing their knowledge and making relationships explicit.

CmapTools, developed by the Institute for Human and Machine Cognition (IHMC), is a suite of publicly-available[1] software tools for knowledge acquisition, construction, and sharing [Cañas et al., 2004] based on concept maps. The software, used in over 150 countries, facilitates construction and sharing of knowledge models based on concept maps, and also enables the use of concept maps to serve as the browsing interface to a domain of knowledge. The tools facilitate the linking of a concept to other concept maps, pictures, images, audio/video clips, text documents, Web pages, etc., enabling users to navigate to relevant resources by moving through concept maps.

Concept maps capture "informal" knowledge models: Although nodes and links can be seen as encoding propositions, they are not represented in a formal logic, and have no associated formal semantics. However, they provide a concise representation of information for human use, providing a form of representation between that of traditional representations—which are hard to capture and require intervention by knowledge engineers—and text—which may be hard to interpret. Concept maps are used by people of all ages, from elementary school children to NASA Scientists. More important, experts are able to construct knowledge models of their domain without the need for a knowledge engineer's intervention, or to actively participate in the knowledge elicitation if a knowledge engineer leads the process.

The CmapTools client provides a simple point-and-click interface to build new concept maps. Users can construct new concepts by double-clicking into a concept map window and entering the name of the concept into the appearing text field. They can then link two concepts by clicking on the arrow button of a selected concept and dragging the displayed arrow to a target concept or

[1] http://cmap.ihmc.us/

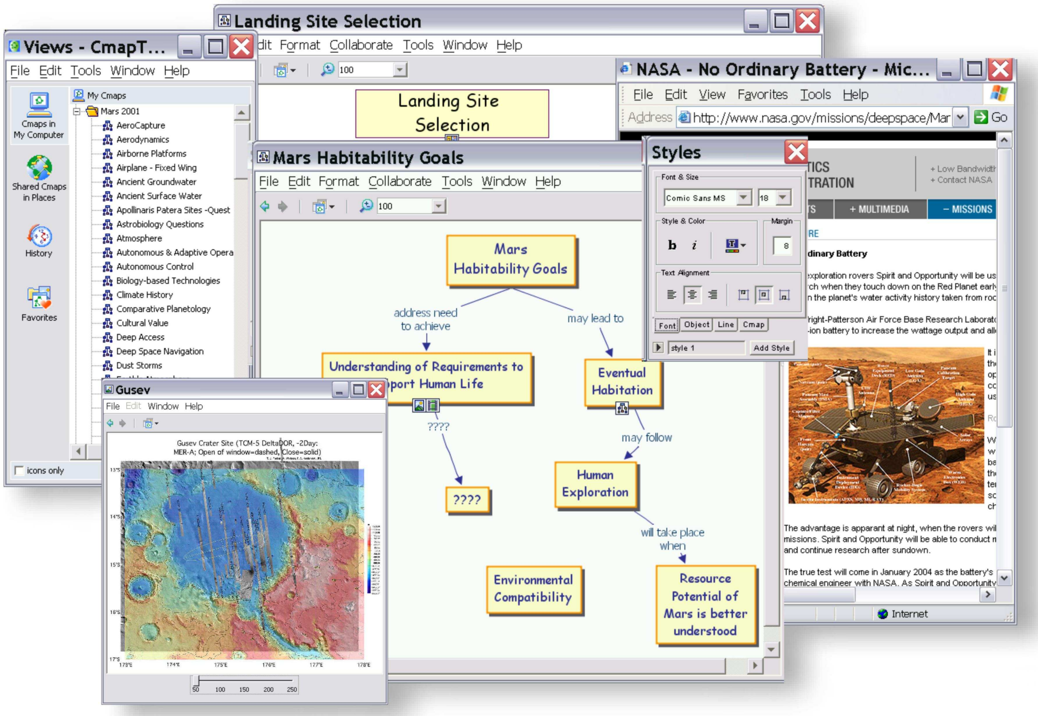


Figure 2.1: The IHMC CmapTools client.

the background of the concept map for creating a link to a new concept. When the link has been constructed, users can specify the label of the link. Users can link concept maps and other multimedia resources to concepts using menu options or a drag-and-drop interface. Figure 2.1 shows the CmapTools client interface displaying part of a knowledge model and a concept map under construction.

The CmapTools system and its accompanying knowledge elicitation methodology have been used successfully for capturing, representing and sharing expertise in a variety of domains. Applications include a nuclear cardiology expert system [Ford et al., 1996]; a prototype system to provide performance support and just-in-time training to fleet Naval electronics technicians [Cañas et al., 1998]; a knowledge preservation model on launch vehicle systems integration at

NASA [Coffey et al., 2002], a large-scale knowledge modeling effort to demonstrate the feasibility of eliciting and representing local meteorological knowledge undertaken at the Naval Training Meteorology and Oceanographic Facility at Pensacola Naval Air Station [Hoffman et al., 2001], and a large multimedia knowledge model on Mars [Briggs et al., 2004], constructed entirely by a NASA scientist, without the participation of knowledge engineers.[2]

2.2 Intelligent Support Systems

An important question in knowledge management is how to determine the information to capture. In traditional views, knowledge capture may be seen as primarily knowledge acquisition, capturing knowledge that already exists within the expert. In this dissertation research we study methods for supporting an alternative approach, "knowledge extension," based on the premise that a knowledge model evolves from coordinated processes of knowledge acquisition and knowledge construction. In this view, it is crucial to support experts' construction of new knowledge as they extend existing knowledge models.

Concept Mapping in CmapTools is facilitated by a family of intelligent suggesters that provide content-based support to users as they extend concept maps by adding concepts and propositions, and as they select topics for new maps. The goal is to provide scaffolding for experts as they build their own concept maps, link their maps to others', and decide how to extend their knowledge models. This family of intelligent support tools combines ideas from the research areas of intelligent aides and suggester systems. These areas are huge, interdisciplinary, and very dynamic. We present an illustrative—rather than exhaustive—review of the literature on these fields followed by an outline of the three systems developed to provide intelligent support for knowledge extension.

[2] http://www.cmex.arc.nasa.gov

Intelligent Aides

Aides are support tools that operate in association with the user to effectively accomplish a range of tasks. Some aides serve the purpose of expanding the user's natural capabilities, for example by acting as intelligence or memory augmentation mechanisms [Engelbart, 1962]. Some of these systems reduce the user's work by carrying out the routinizable tasks on the user's behalf. Others offer tips on how to refine or complete human generated products (such as electronic documents) by highlighting potential inaccuracies and proposing alternative solutions. Some aides "think ahead" to anticipate the next steps in a user's task providing the capability for the user to confirm the prediction and ask the system to complete the steps automatically. A popular kind of aides are those that come integrated into complex software systems and attempt to make users aware of the various features of the systems.

Many aides are based on the *intelligent agent metaphor* [Maes, 1994, Bradshaw, 1997, Negroponte, 1997, Laurel, 1997]. These aides operate as assistants with high degree of autonomy. Others adopt a *user-driven* approach and need to be initiated by commands or direct manipulation GUIs [Sutherland, 1963, Ziegler and Fahnrich, 1988, Shneiderman, 1992]. An intermediate group of aides reconciles both views, giving rise to *mixed-initiative user interfaces* [Horvitz, 1999]. While many kinds of interface tools can be regarded as aides, our interest here is in those that act in cooperation with people, complementing their abilities and augmenting their performance by offering proactive or on demand context-sensitive support.

Intelligence and Memory Augmentation

Joseph C. R. Licklider [Licklider, 1960] is usually referred to as the trailblazer in the area of cooperative aides. He proposed the notion of man-computer symbiosis as a subclass of man-machine systems. He envisioned human brains and computing machines coupled together very tightly, with the resulting partnership outperforming any human brain or known machine. A seminal work on *memory augmentation* aides is the Forget-me-not system [Lamming and Flynn, 1994]. Forget-me-not kept a record of a person's past activity, allowing retrieval of relevant information based on context. The system was expected to aid the user anytime and anywhere, therefore the system and its data resided on a small portable device called ParcTab. The context cues used to retrieve information included location, phone calls, and interaction between different people carrying the device.

A family of *Just-In-Time Information Retrieval* (JITIR) systems serving as memory augmentation aides has been implemented at the MIT Media Lab. JITIR systems are characterized for being proactive, unobtrusive and aware of the user's local context. A desktop version of a JITIR system, Remembrance Agent [Rhodes and Starner, 1996], is designed to run on the background of a computer, observing what the user types and reads on a text editor. Remembrance Agent uses that information to retrieve related documents and user's old emails, which become available through an unobtrusive interface. Wearable Remembrance Agent [Rhodes, 1997, Rhodes et al., 1999] is a portable, continuously running agent that uses the physical context to find information relevant to the user's situation. Another memory augmentation device developed at MIT media Lab is Memory Glasses [DeVaul and Pentland, 2002], a wearable aid that utilizes a context-awareness system based on sensors for vision and listening. Interaction is performed through buttons for user input into a light wearable computing core, while headphones and a clip-on display are used for information delivery.

Another example of memory augmentation aid is illustrated by the CybreMinder system [Dey and Abowd, 2000], a context-aware reminder application built using the Context Toolkit [Salber et al., 1999]. A salient feature of CybreMinder is its ample view of context, which includes location, time, activity, identity, physical/environmental conditions and potential co-located collaborators. An added important characteristic of CybreMinder is its support for customizing the way reminders are delivered. Based on this customization the systems employs the user's context to choose among different ways for delivering the reminders, including SMS on a mobile phone, e-mail, printing on a local printer or using a nearby display from a wearable, handheld, or static CRT.

Aides that Think Ahead

Aides that monitor the user's task to anticipate next steps and offer automatization of predicted actions are popular mostly in word processing and programming environments. Eager [Cypher, 1991] is an aid for HyperCard that monitors the user's activity and learns from examples. Eager draws on ideas of programming by example [Smith, 1977, Lieberman, 1987, Maulsby and Witten, 1989] to generalize user's repetitive patterns and anticipate what the user will do next. The system highlights menus and objects on the screen to indicate its predictions. If a correct anticipation has been generated the user can tell Eager to complete the task automatically.

Another text prediction aid is CIMA [Lieberman and Maulsby, 1996], an instructible agent that learns from conversational processes with the user, including examples and advice, and then suggests completions of sentences. Schlimmer and Hermens (1993) proposed and interactive note taker that uses finite state machines and decision trees to predict what the user is going to write and provides a default text that the user may select. OWL [Linton et al., 2000] is a writer's support tool that analyzes the sequence of commands typed with Microsoft Word to anticipate potentially useful next commands. OWL proposes commands to the user based on a repository containing the log of editing commands typed by different users. SMARTedit [Lau, 2001] is a programming by demonstration system that applies concept learning [Mitchell, 1982] to learn repetitive text-editing programs by example, automating repetitive tasks. Another tool, Writer's Aid [Babaian et al., 2002], is a collaborative interface that uses a planning system to support an author's writing efforts. While editing a research manuscript an author can insert a citation command followed by a few search parameters and then continue the writing task. Writer's Aid searches the user's preferred bibliographies and paper collections for reference to the particular citation command. Once the search is completed, the user can easily access a summary of the retrieved data, view any of the found articles, and ask the system to automatically insert certain bibliographic records on the bibliographic file as well as to place the pertinent citation keys in the text of the article.

Critics and Helpers

A different class of aides is illustrated by software assistants known as *critics* or *critiquing systems* [Silverman, 1992]. Critics are cooperative tools that observe the user interacting with a computer system and present reasoned opinions about a product under development. The goal of the critiquing systems is to discover and point out errors or suboptimal results that might otherwise remain unnoticed, and to help users to make the necessary repairs. Critics need a metric to evaluate the quality of a solution and usually generate their advice by using a specialized domain knowledge base. Most popular critiquing systems have been developed to assist word processing. These include spelling-, grammar-, and style-checkers [Kukich, 1992, Church and Rau, 1995, Bustamante et al., 1996].

Intelligent tutoring is another field for which critiquing systems provide natural support. A noteworthy instance is COACH [Selker, 1994], a proactive critiquing system for students learning the LISP programming language. COACH creates an adaptive model of the student by monitoring mistakes and then employs that model to provide advice. Critics have been implemented in many other applications, like diagnosis and decision-making [Miller, 1983], expertise-based design [Fischer et al., 1993], and knowledge acquisition system [Terveen and Wroblewski, 1990].

Microsoft Office Assistant (typically personified by Clippy) is certainly one of the best-known computer aides. It was developed in the framework of the Lumiere Project [Horvitz et al., 1998] and first distributed with Microsoft Office'97 product suite. The purpose of the Office Assistant is to provide support to Microsoft software users. It relies on Bayesian networks and influence diagram to model users' activity and predict their needs over time. The user can determine the level of obtrusiveness of the assistant and obtain help both proactively and on demand.

A general purpose and extensible framework for constructing context-aware assistants is provided by Suitor [Maglio et al., 2000, Maglio and Campbell, 2003]. Suitor is a collection of "attentive agents" that gather information from the users and the world and post that information on a centralized blackboard. A class of agents called investigators determine users' information needs by monitoring users' behavior and context, including eye gaze, keyword input, mouse movements, visited URLs and software applications on focus. On the meantime investigator agents watch Web pages and databases for updates. A second group of agents, the reflector agents, interact with the blackboard by prioritizing posted information and matching it with users' needs. Finally delivery agents display relevant information to users.

Aiding Knowledge Modeling

The support tools reviewed in this section address many of the needs of computer-users dealing with complex tasks. Knowledge modeling is a task that can greatly benefit from the use of intelligent aides.

CmapTools has been extended to aid the user in the creation of knowledge models. One of CmapTools' aides, "Joe in a Box", is a critiquing system that monitors the user's construction of a concept map, inspects several aspects of the map, and provides reasoned suggestions on how to improve the map. The suggestions provided by this aid are based on Joe Novak's guidelines

[Novak, 2002] on how to construct good concept maps. "Joe in a Box" warns the user of the existence of repeated labels or links containing too many words. It also points out potential problems related to the topology of the concept map. For instance, if the concept map is skewed to one side, or if it has no clear superordinate concept, "Joe in a Box" will detect the problem and provide advice.

Another component of CmapTools is a Word Sense Disambiguation aid [Cañas et al., 2003]. In order to resolve the correct sense of a word this aid uses topological information from the concept map to discover key concepts. Once these concepts are selected from the map the system uses the senses and semantic relations provided by WordNet [Fellbaum, 1998] to perform disambiguation.

CmapTools provides a search-enhancer tool [Carvalho et al., 2001], which takes queries generated by the users and searches the Web for information related to a map in progress or being browsed. When the user presents a query, a mobile agent is created that operates on top of one or more meta-search servers to query publicly available search engines. To filter and rank the results returned by the search engines, the agent uses contextual information extracted from the concept map at hand.

A family of aides integrated into CmapTools provides proactive and on-demand suggestions of concepts, propositions, multimedia resources, concept maps and topics to assist experts as they extend partial knowledge models. The implementation of a topic suggester system to aid knowledge modeling is the focus of this dissertation. In the next section we present a literature review of suggester systems, followed by an outline of CmapTools' suggesters.

Suggester Systems

Suggester systems, also known as recommender systems [Resnick and Varian, 1997], assist users in a plethora of computer-mediated tasks, by providing guidelines or hints. Most suggesters are aimed at helping users to deal with the problem of information overload by facilitating access to relevant items. Suggesters have emerged in diverse scenarios including science, education, entertainment, and commerce. Although suggesters may serve very different goals, they are all guided by a common principle: to collaborate with users by suggesting rather than acting. In that sense, suggesters provide the facts, links or tips but it is up to the user to decide how the suggestions are ultimately utilized.

Dimensions of Analysis

Suggesters adopt mainly two different views to help predict information needs, usually referred to as the *user-modeling* and *task-modeling* approaches. Suggesters based on the user-modeling schema attempt to create a profile or model of the users by observing users' behavior. These systems can be collaborative, building on similarities between users with respect to the objects they interact with, or content-based, building on similarities between potential recommendations and the objects that the user liked in the past. In both cases, the user's long-term interests need to be represented as an aggregation of objects or keywords. On the other hand, task-modeling schemas rely on the context in which the user is immersed at the time the request is made. The context may consist of an electronic document the user is editing, Web pages the user has recently visited or any other item representative of the user's current activity.

It is common to classify suggesters according to the personalization level they offer. User-modeling suggesters provide a persistent personalization level while task-modeling suggesters implement an ephemeral one. Another dimension of analysis is how to define similarities between users or contents. Many algorithms have been applied to compute these similarity measures, combining several methods coming either from the information retrieval or the machine learning areas. Commonly applied techniques are based on cluster analysis [Everitt, 1980], cosine similarity [Salton, 1989], K-nearest neighbors [Stanfill and Waltz, 1986], LSA [Deerwester et al., 1990], and Bayesian classifiers [Duda and Hart, 1973] among many others. Additional dimensions of analysis are the content of the suggestion (e.g., news, URLs, people, articles, text, products), the purpose of the suggestion (sales or information), the event that triggers the search for suggestions (user's demand or proactive), the way the system learns the user's interests (monitor user's behavior, receive feedback, engage in conversation with the user, or programmed), and the level of intrusiveness.

Collaborative Filtering

A common technique adopted by many suggester systems is collaborative filtering, which infers the preferences of individual users based on the behavior of multiple users. Collaborative filtering is based on the assumption that human preferences are correlated. Tapestry [Goldberg et al., 1992] is usually referred to as the first collaborative filtering system. It provided a mechanism for filtering email and news messages based both on the content of the messages and on implicit or explicit feedback from users. Feedback included manual annotations and automatically observed reactions (e.g., some user sent a reply to a message). Following Tapestry's initiative, a large number of suggester systems were developed and applied to diverse domains, providing different levels of personalization.

Web Recommender Systems

Given the huge amount of information existing on the Web it is not surprising that the great majority of the suggester systems have been built around content and resources available online. WebWatcher [Armstrong et al., 1995] is an early attempt to assist users locating information on the Web by highlighting hyperlinks in a page based on the declared preferences and browsing history of a user as well as information gathered from other users with similar interests. Personal WebWatcher [Mladenic, 1996] is a successor of WebWatcher that learns individual users' interests by observing their browsing behavior. Letizia [Lieberman, 1995] is a user interface agent that unobtrusively assists Web browsing. As the user navigates Web pages, Letizia performs a breadth-first search augmented by several heuristics to anticipate what items may be of interest to the user. Syskill & Webert [Pazzani et al., 1996] uses information retrieval techniques to process the content of a page rated by a user, and machine learning to acquire a model, that is utilized to predict which links on a Web page a user will find useful. SenseMaker [Baldonado and Winograd, 1997] is an interface that facilitates the navigation of information spaces by providing task specific support for consulting heterogeneous search services. The system helps users to examine their present context, move to new contexts or return to previous ones. SenseMaker presents the collection of suggested documents in bundles (their term for clusters), which can be progressively expanded.

Fab [Balabanović and Shoham, 1997] is a hybrid content-based, collaborative Web page recommender system that learns to browse the Web on behalf of a user. Fab generates recommendations by the use of a set of collection agents (that find pages for a particular topic) and selection agents (that find pages for a particular user). Users' explicit ratings of the recommended pages combined with several heuristics are used to update personal-agents' profiles, remove unsuccessful agents, and duplicate successful ones. Broadway [Jaczynski and Trousse, 1997] is a case-based reasoning system that monitors a user's browsing activity and provides advice by reusing navigational cases extracted from past browsing experiences of a group of users. Another Web navigation assistant is SiteSeer [Rucker and Polanco, 1997], which recommends pages collaboratively by looking at users' bookmarks. Alexa [Kahle and Gilliat, 1998] is a commercial Web search engine that augments Google search results by combining them with information like user reviews and ratings of the Web sites, traffic statistics and related links. Other Web suggester systems include LIRA [Balabanovic et al., 1995], BASAR [Thomas and Fischer, 1997], ifWeb [Asnicar and Tasso, 1997], SOAP [Voss and Kreifelts, 1997], Let's Browse [Lieberman et al., 1999], SurfLen [Fu et al., 2000], Margin Notes [Rhodes, 2000] and Quickstep [Middleton et al., 2001], among many others.

An example of hybrid news filtering system is NewsDude [Billsus and Pazzani, 1999], a learning agent that is trained by the user with a set of interesting news articles. A hybrid social chat recommender system is Butterfly [Van Dyke et al., 1999], a system that uses keywords to find interesting conversations in Usenet newsgroups. Collaborative news recommender systems include GroupLens [Resnick et al., 1994, Konstan et al., 1997] and PHOAKS [Terveen et al., 1997].

Task-Contextualized Suggesters

Several suggester systems exploit user interaction with computer applications to determine the user's current task and contextualize information needs. This gives rise to context-aware suggester systems. Some of the memory augmentation aides discussed in the previous section can be thought of as task-contextualized suggester systems.

The Watson system [Budzik and Hammond, 1999, Budzik and Hammond, 2000, Budzik et al., 2001] is a context-aware suggester that attempts to find relevant online resources. Watson is part of a family of programs known as *Information Management Assistants* (IMAs) developed at the InfoLab of Northwestern University. The purpose of the IMAs is to anticipate the user's needs and to provide proactive and on demand support for the user's current activity. In order to achieve this goal, IMAs generate a model of the user's task, access information retrieval systems on the user's behalf, and unobtrusively deliver useful material. IMAs provide an environment in which resources are retrieved proactively as well as mechanisms that augment users' explicit queries with keywords extracted from the current task. Point/Counterpoint [Budzik et al., 2000] is another IMA built on top of Watson. Instead of retrieving general information, Point/Counterpoint brings opposing arguments.

CALVIN [Leake et al., 2000, Bauer and Leake, 2001] is a case-based reasoning context-aware system that monitors the user's Web browsing activity to generate a model of the user's task. In addition it provides capabilities for users to manually enter information about a variety of resources, such as descriptions of books or articles, and data on useful contact people. The gathered material is stored as contextualized cases recording information users' consult during their decision-making. CALVIN provides an interface that proactively and unobtrusively suggests stored material when the user context is similar to the one associated with the stored cases.

Recommendations in Other Domains

There are several other domains in which suggester systems have proven to be useful. ReferralWeb [Kautz et al., 1997] aims at direct people to experts on a given topic. ExperFinder [Vivacqua, 1999] is a Java programmer's assistant that looks for other programmers using the same classes as the user. CiteSeer [Bollacker et al., 1998] is a Web-based research paper finder that uses metadata extracted from scientific publications and similarity measures among online available articles to suggest relevant material and facilitate access to it. Foxtrot [Middleton et al., 2003] is another research paper recommender that support content and collaborative filtering as well as ontological user profiling and profile visualization. ELFI [Nick et al., 1998] is a research funding recommender system that tracks user interaction with the system to suggest relevant unseen database entries. The Adaptive Place Advisor [Langley et al., 1999, Göker and Thompson, 2000] is a personalized conversational recommendation system that engages in dialogues to help users decide on a destination.

The entertainment and E-Commerce domains have also been the focus of many recommender system products. Ringo and Firefly [Shardanand and Maes, 1995] are influential instances of collaborative music recommender systems. Other examples of commercial recommender systems include Amazon.com$^{\text{TM}}$ (www.amazon.com), My CDNow$^{\text{TM}}$ (www.mycdnow.com) and MovieFinder (www.moviefinder.com), among a great number of others. Schafer et al. (1999) and Middleton [Middleton, 2003] outline some of these systems.

The suggester systems in CmapTools are task-contextualized; they take the user's current map as context to search for relevant material. Being able to access relevant material at the right times can facilitate the construction of high-quality knowledge models. In our view, the effectiveness of suggester tools depends on their ability to anticipate which material is relevant and make it easily accessible to the user in a unobtrusive manner. The next section outlines the suggesters implemented as part of the CmapTools system to aid users as they extend their knowledge models.

2.3 Aiding Knowledge Extension in CmapTools

The CmapTools effort includes a collaboration between researchers at IHMC and Indiana University to develop tools to aid the knowledge extension process. The tools are designed to address difficulties which have been observed arising during concept mapping. For example, users sometimes stop and wonder what concepts to add to a concept map; spend time trying to find the right word to use in a concept label or linking phrase; search for relevant concept maps to compare; and search the Web for additional material to enhance the concept map or to jog their memories for topics to include. Each of these has been addressed by a system to suggest relevant information, based on the context provided by the concept map. Each system starts from a concept map under construction, and proactively suggests relevant information such as concept maps, propositions, multimedia resources, concepts and topics.

The next three sections outline three approaches which start from a concept map under construction and mine related information—both from prior concept maps, and from the Web—to propose material to aid the user's knowledge capture and knowledge construction.

Suggester for Concepts

The goal of the *concept suggester*, developed at IHMC by Marco Carvalho and Marco Arguedas, is to facilitate concept map construction by proactively searching new concepts and suggesting them to the user [Cañas et al., 2002]. The concept suggester proposes collections of terms, each of them representing a concept that is novel (i.e., not contained in the current map) but potentially relevant. This can (1) help the user to remember familiar concepts that might otherwise be forgotten, and (2) give the user the opportunity to further explore and understand new and potentially relevant concepts.

To search for relevant concepts the system first mines the Web for documents related to the current map [Carvalho et al., 2001]. The collected documents are cached in a database for further analysis and for concept extraction. The state of the map under construction is continuously monitored for significant changes that could trigger a new search for concepts to add to the cache. Significant changes in the map are defined as any modifications of important nodes according to the topological analysis models we will discuss in section 3.3. Such modifications may affect the relevance of cached documents to the current context, thus requiring the system to launch a new search.

Figure 2.2 shows the process for searching new concepts. A search process starts with a request for concept suggestions sent from the CmapTools client to a search server. All the processing occurs at the server side, avoiding any additional processing load on the client or client use of additional network bandwidth. At the server side, the map is converted into a text query for a meta-search engine/crawler to retrieve additional documents that will be added to the database, and the database is searched for documents that are relevant to the context of the map. For performance reasons, this search process takes place in parallel, allowing for a timely response to the search request while still supporting database updates for future requests.

The subset of relevant documents retrieved from the database is then searched for potential concept suggestions. The current approach to extracting relevant concepts starts by searching the documents for concepts that are already in the map. Each time a concept is found in a document, all the neighboring words are saved in a temporary table as potential suggestions. Neighboring words are defined as the non-stop words in the document within a fixed distance threshold (currently 3 words) of the concept term. After searching for all the map's concepts in all the documents the system collects a large collection of terms that are, at some level, neighbors of the map's concepts in the text. A frequency analysis is then applied to rank these terms and determine the subset for the suggester to display. Preliminary experiments [Cañas et al., 2002] with the concept suggester show promising results.

Suggester for Propositions, Concept Maps and Multimedia Resources

Previously-built knowledge models, shared from other users, may be helpful to suggest propositions to consider and concept maps to consult while constructing a new concept map. To provide suggestions of propositions and concept maps, the proposition and resource suggester, developed at Indiana University [Leake et al., 2003a], applies techniques inspired by case-based reasoning

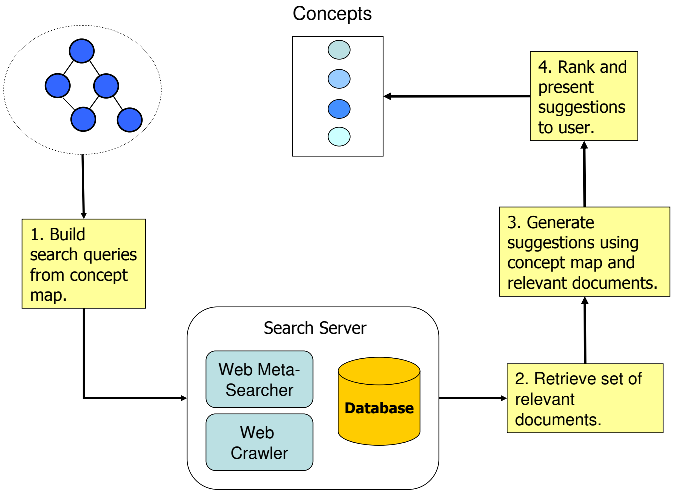

Concepts
4. Rank and present suggestions to user.
1. Build search queries from concept map.
3. Generate suggestions using concept map and relevant documents.
Search Server
Web Meta-Searcher
Web Crawler
Database
2. Retrieve set of relevant documents.

Figure 2.2: The process for searching new concepts.

[Kolodner, 1993, Leake, 1996].

The concept maps of various users are considered as case-bases of their concept-mapping activity, with each concept map considered to be a separate case. When a new user wants to "extend" a concept—add a new connected concept—the system views prior concept maps including the original concept as examples of how that concept was extended in the past.

In the current implementation, case libraries are compiled periodically from concept maps on the CmapTools servers and clients, generating case representations from raw concept maps and indexing new concept map cases. Each case stores information about a map's content, its structure, and links to other concept maps and resources that are attached to its nodes. This information is necessary to generate suggestions in the form of propositions, concept maps, and relevant multimedia resources that may be helpful in extending and annotating new concept maps.

Central to any case-based approach are techniques for indexing—characterizing when cases are likely to be useful in the future. The system guides retrieval based on a category index, implemented by Thomas Reichherzer. The index is computed from the concept map library and organizes concept maps into a hierarchical structure of categories, each containing a set of concept maps involving related concepts. More tightly coupled clusters of concept maps appear towards the bottom of the hierarchical structure, and more loosely coupled clusters towards the top. For each category, the index maintains references to the original concept maps and a cluster representative, generated from concept maps in the category to serve as a prototype. The cluster representative is used to determine if a new concept map is related to the maps in a category. Concept map similarity is computed from a vector representation of the concept maps. This representation is similar to the popular term-frequency vector with inverse-document frequency adjustment (TF-IDF) [Salton and Yang, 1973], but takes advantage of the structure of concept maps to adjust term weights, based on structural and topological clues to concept importance. The models developed for assessing concept importance are discussed in section 3.3.

Users can actively initiate search for new concepts or multi-media resources by selecting the concepts for which extensions are sought, or can rely on the system to monitor concepts being added to the concept map and proactively suggest propositions or annotations. Propositions in the map are encoded as concept-link-concept triples, where the link is outgoing from the first concept, and incoming to the second.

Figure 2.3 shows the process for searching propositions, concept maps and multimedia resources. Whether in user-driven or proactive mode, the suggester converts the map in progress to a term vector representation and extracts keywords from the concepts selected by the user or the suggester. The keywords of the selected concepts and the vector representation form a query, processed locally by the client and remotely by a designated index server. While the keywords are used to look up specific suggestions in a case, the term vector serves as a context in the search for suggestions. The vector is used to perform a binary search for the best-fitting category starting from the top of the relevant hierarchies in the combined category index and going towards the bottom.

By adjusting a slider, users can control how far the retrieval algorithm descends in the category index hierarchy tree to search for related concept maps. The further it descends, the fewer maps it finds, but those found are more closely related to the map in progress. This allows users to control how broad or narrow a search should be performed. Once a set of related maps has been identified, they are examined to find suggestions for propositions to extend the current map and to suggest resources linked to relevant nodes in the retrieved map.

Suggestions extracted from a case library are ranked by means of a keyword association factor, based on the distances between concepts within a concept map. The keyword association factor is discussed in detail in [Leake et al., 2002] and [Leake et al., 2003b]. Among all the potential suggestions only the n most relevant ones are displayed, sorted by their rank. The value of n can be changed by the user.

Initial evaluations of indexing performance and proposition suggestion show promising results. Details on preliminary experiments performed to evaluate the proposition suggester system can be found in [Leake et al., 2002, Leake et al., 2003a, Leake et al., 2003b].

Suggester for Relevant Topics

Suggestions from previous concept maps are useful for elaborating new maps, but cannot help to extend the knowledge model beyond information that has already been captured in the concept map libraries. Another suggester, EXTENDER (EXtensive Topic Extender from New Data Exploring Relationships), developed at Indiana University, identifies and suggests novel topics that the expert may wish to include in the knowledge model.

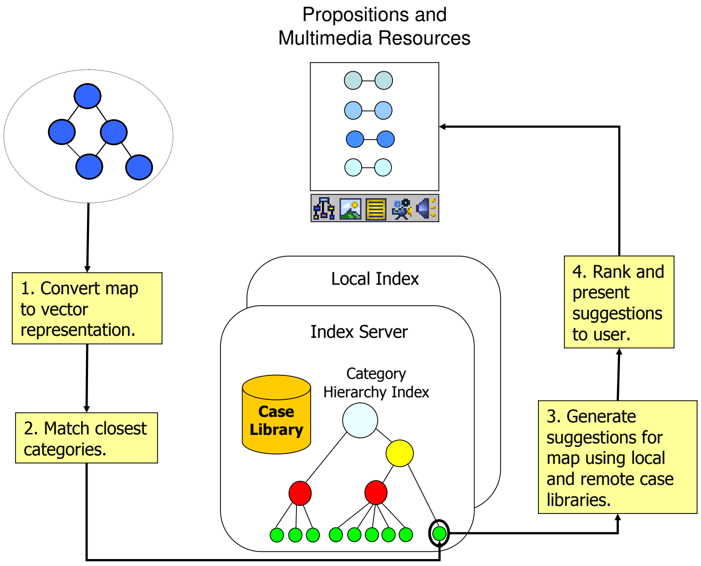

Propositions and
Multimedia Resources
1. Convert map to vector representation.
2. Match closest categories.
Local Index
Index Server
Case Library
Category Hierarchy Index
3. Generate suggestions for map using local and remote case libraries.
4. Rank and present suggestions to user.

Figure 2.3: The process for searching propositions, concept maps and multimedia resources.

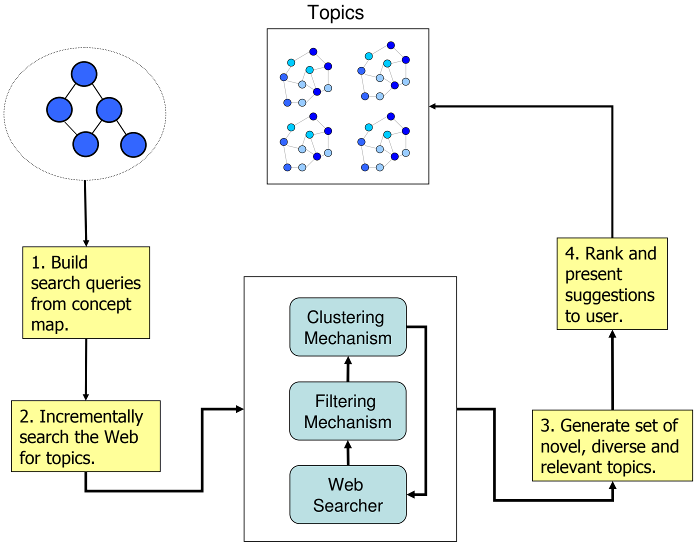

Topics
1. Build search queries from concept map.
2. Incrementally search the Web for topics.
Clustering Mechanism
Filtering Mechanism
Web Searcher
3. Generate set of novel, diverse and relevant topics.
4. Rank and present suggestions to user.

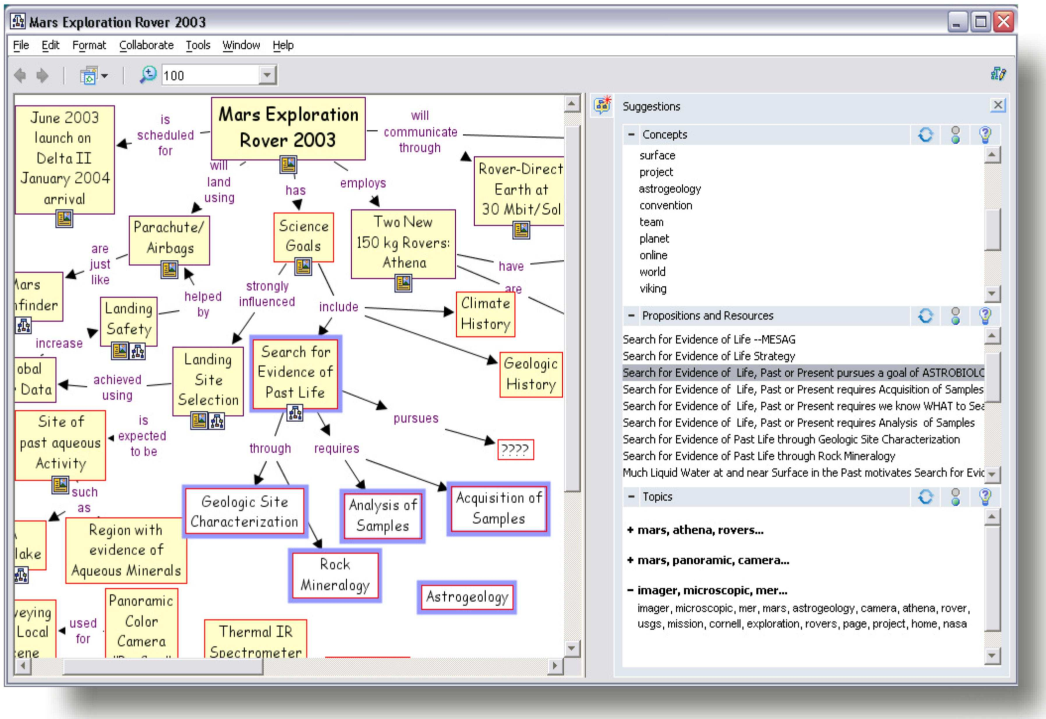
Figure 2.4: The process for searching new topics.

Topics are commonly defined as pieces of data that have been grouped together as a result of having a common theme. EXTENDER's process for searching new topics is outlined in figure 2.4. The system produces topics by an iterative process which takes a knowledge model as input and queries a Web search engine to find documents related to the initial model. At each step, the information found is clustered and incrementally used to guide further search, resulting in a sequence of generations of new topics. Irrelevant information is filtered by contrasting the search results with the search context, initially defined using the knowledge model under construction, and then progressively updated as the focus moves through a connected series of topics. Cohesive topics are generated by clustering the results returned by the Web search process. The system uses a curiosity mechanism to favor exploration during initial stages and exploitation at the end of the process. After a few iterations the process yields a final collection of topics which the system presents as suggestions to the user.

The design and evaluation of techniques to support knowledge extension by means of a topic suggester is the focus of this dissertation research. A framework for topic generation and a description of the methods applied in the implementation of EXTENDER are discussed in detail in chapters 4 and 5. The evaluation of the proposed methods is reported in chapter 6.

Integrated Suggestion Presentation

The three suggester systems address the challenges of proactively and unobtrusively providing the knowledge modeler with suggestions to extend the model under construction. To integrate the material collected by the three suggesters and present them in a convenient form, we use a suggestion panel implemented for CmapTools by Sofia Brenes. The panel is attached to the side of a concept map and becomes visible only when the user decides to open it; otherwise, an unobtrusive signal lets users know when suggestions have arrived if the panel is closed. Figure 2.5 depicts a



Concepts

surface
project
astrogeology
convention
team
planet
online
world
viking

Propositions and Resources

Search for Evidence of Life --MESAG
Search for Evidence of Life Strategy
Search for Evidence of Life, Past or Present pursues a goal of ASTROBIOLC
Search for Evidence of Life, Past or Present requires Acquisition of Samples
Search for Evidence of Life, Past or Present requires we know WHAT to Sea
Search for Evidence of Life, Past or Present requires Analysis of Samples
Search for Evidence of Past Life through Geologic Site Characterization
Search for Evidence of Past Life through Rock Mineralogy
Much Liquid Water at and near Surface in the Past motivates Search for Evic

Topics

+ **mars, athena, rovers...**

+ **mars, panoramic, camera...**

− **imager, microscopic, mer...**
imager, microscopic, mer, mars, astrogeology, camera, athena, rover, usgs, mission, cornell, exploration, rovers, page, project, home, nasa

Figure 2.5: Concept map under construction with associated Suggestions.

map under construction and the side panel with associated suggestions. Controls allow users to enable or disable particular suggesters, to request an update on the presented suggestions, and to request additional suggestions of a given type.

3

Modeling Concepts and their Descriptive Power

Concept mapping was developed in an educational setting by Joseph Novak, in an effort to design better teaching and learning activities [Novak and Gowin, 1984]. Novak based the approach on Ausubel's cognitive learning theory [Ausubel, 1963, Ausubel, 1968], which proposes that meaningful learning is a process in which new information is related to an existing relevant aspect of an individual *cognitive structure*.

3.1 Concepts and Cognitive Structure

Cognitive structure is a central construct of Ausubel's theory of meaningful learning. This theory emphasizes the importance of a clear, stable and suitably organized structure, forming connections between pieces of knowledge, to facilitate new learning and retention. The process of learning requires deliberate effort by the learner to connect new concepts to relevant preexisting concepts and propositions in the learner's own cognitive structure. Concept mapping was designed to support the learner's effort by externalizing concepts and propositions known to the student, making them visually apparent to facilitate their connection with newly acquired concepts.

Most theories of knowledge organization emphasize the importance of concepts and their associations. According to the *classical view of concepts*, which dates back to the philosophical works of Plato and Aristotle, a concept meaning can be characterized by a conjunctive lists of properties. All properties used in defining a concept must be necessary and sufficient to identify what is and what is not an instance of the concept. The classical view has shown to be limited—some of its predictions are highly questionable or have been shown to be untrue [Medin and Smith, 1984]. Other lines of research (e.g., [Rosch et al., 1976, Tversky, 1977, Schank and Abelson, 1977, Schank et al., 1986, Wisniewski and Medin, 1991, Gädenfors, 2000]) have examined alternative frameworks to account for certain aspects of complex knowledge organization.

For the purpose of this study, we adhere to Novak's definition of concepts. Novak defines *concepts* as "perceived regularities in events or objects, or records of events or objects, designated by a label." According to Collins and Quillian (1969) concepts are formed to promote *cognitive economy*. In other words, humans consider certain elements of the world as instances or members of a class to decrease the amount of information to perceive, learn and retain. In addition, as has been suggested by Ausubel, the arrangement of concepts in an individual cognitive structure tells us about an individual's organization of knowledge in a particular subject-matter field at any given time.

Empirical studies provide support for the usefulness of concept maps for assessing cognitive structure [Aidman and Egan, 1998, Michael, 1994]. As a consequence, concept maps are particularly useful in education and cognitive psychology as a medium for examining the organization of knowledge in human memory [West et al., 2002].

3.2 Knowledge Representation in Memory

Descriptive Theories of Knowledge Organization

The problem of knowledge representation in memory has been a central issue in the study of human cognition. The first theories developed for explaining the organization of knowledge can be traced back to Kenneth Craik's work on the early 40's where he introduced the notion of mental model as small-scale symbolic representations of reality [Craik, 1943]. Most subsequent descriptive theories of knowledge organization in memory have followed Craik's principle that the mind is a symbolic system. Ausubel's cognitive structure is an example of such theories. Other symbolic theories of knowledge organization include semantic memory [Quillian, 1968, Tulving, 1972], frames [Minksy, 1975], and scripts [Schank and Abelson, 1977]. Each of these theories presents unique ideas about the structure of knowledge in memory. For example, the essential organizing principle of cognitive structures is that of "hierarchy". Semantic memory also assumes that human memory is arranged in a hierarchical fashion but focuses mainly on "semantic relatedness" or "semantic distance" between concepts. The frame and script theories, alternatively, claim that knowledge is organized around "expectation" and can be modeled using slot-fillers, pointers between frames or scripts, and instantiation procedures. Despite the many differences among theories of knowledge organization in memory, all of them share the fundamental premise that models of knowledge are built in terms of components, and that these components are organized.

The nature of the components that represent knowledge varies among representations of words, concepts and events. The organization of these components refers to the specific relations among them, where these relations can be definitional, instances, temporal, causal, or class inclusions, among others. Schemes based on graphs or networks are commonly used as models of human memory organization, to account for phenomena such as similarity judgments or hierarchical category structure. Proposals for non graph-based representations to model concepts and their relationships include formal concept analysis [Ganter and Wille, 1999], which models the organization of concepts in terms of lattice theory, and the geometric structure of conceptual spaces [Gädenfors, 2000].

Externalizing Knowledge

Associated with many descriptive theories is a scheme for externalizing the way knowledge is represented in the human. An important aspect of an external representation is that it allows us to reach conclusions by looking only at features of the representation. Successful knowledge management largely depends on the ability to elucidate the experts' understanding of a domain, to represent that understanding in a form that supports effective examination by others, and to make the encoded knowledge accessible when needed. A central question is how to externalize the needed knowledge.

Most representational systems that have been developed are propositionally based, which means that knowledge is represented as a set of discrete symbols or propositions. Computer scientists looked at formalisms developed by mathematicians and logicians to use as representational structures, and formalisms such as predicate calculus led the way to many AI developments. The predicate calculus approach to knowledge representation has the advantage of providing a powerful and simple representational mechanism with a well understood semantics and inferential component. However, as has been discussed in a number of sources, the language of predicate calculus is not natural to model some of the most salient psychological aspects of knowledge such its associative nature [Rumelhart and Norman, 1988]. This resulted in the development of alternative representational schemes, both formal and informal, in which knowledge pieces are connected to each others to form graphs or networks.

Semantic networks [Quillian, 1968] are formal representation schemes used to model semantic memory. Quillian's semantic network was introduced as a graph-based means of representing concepts in memory, where nodes stand for concepts and relations are associations among sets of concepts. In this way, the meaning of a concept is given by the patterns of relationships in which the concept participates. Some nodes in a semantic network may correspond to words in natural languages, others represent concepts with no natural language equivalent, and others are tokens that represent instances of more general concepts.

The work on semantic networks was followed by other formal approaches to graph-based representations such as KL-ONE [Brachman and Schmolze, 1985] and conceptual graphs [Sowa, 1984]. These representational schemes are closely related to the formalism of predicate calculus and attempt to provide a representation suitable for machine processing. The externalization of knowledge using formal representational schemes maximizes the usefulness of captured knowledge for automated processing but, as discussed earlier, requires considerable involvement by knowledge engineers to mediate knowledge modeling.

Our approach for externalizing knowledge held by an individual builds on concept mapping. By externalizing a cognitive structure as a concept map, individuals can display the organization of their knowledge about certain topics. These externalizations enable knowledge sharing by others.

Electronic concept maps are valuable from a computational perspective because they are machine-readable representations of an individual understanding of a particular topic. From a data-processing view, electronic concepts maps have many advantages over other knowledge externalization forms, such as purely textual forms in at least two respects:

1. in concept maps, concepts and their relationships are readily available, and

2. concept maps are usually hierarchical and have a rich topology.

However, in order to take advantage of this structural information it is fundamental to (1) gain understanding of the different topological roles of concepts in a map, and (2) develop methods for usefully summarizing and applying this information. The next section discusses our approach for assessing the importance of concepts in concept maps and how we use such assessments to build concise and informative summaries of concepts maps.

3.3 Assessing Concept Descriptive Power in Concepts Maps

There has been little study of what affects subjects' judgments of the topic of a concept map, how to determine topic similarity from concept maps, and the types of representations that may support computer models of concept map retrieval. In previous studies using similar types of representations, topological information about graphs has been used to define measures of graph similarity [Goldsmith and Davenport, 1990, Goldsmith et al., 1991] and for concept clustering [Esposito, 1990]. These frameworks are based on the premise that the closer the relationship of two concepts—the "closer" they are in cognitive structure—the closer they will be in the graph representation. This has been used to induce concept proximity or relatedness. In order to assess the importance of concepts in concept maps we investigate a complementary question, the influence of other structural factors, such as the numbers of incoming and outgoing links.

Applying Topological Analysis to Concept Maps

How graph topology affects assessments of concept importance is central to understanding the information conveyed by concept map structure. We developed three candidate models of the influence of structural characteristics on human expectations for the importance of particular concepts to the topic of concept maps. These models have been introduced in [Leake et al., 2004a], and portions of the following are adapted from that work.

In the models, concepts are represented as nodes in the concept map graph and the topology of the concept map is used to compute a weight predicting each concept's importance in describing the topic of the map. To determine which factors to include in the models, we first considered factors from the concept mapping literature. Novak proposed that meaningful learning is facilitated when new concepts or concept meanings are subsumed under broader, more inclusive concepts, which suggests that concept maps should have a hierarchical structure. Our models can reflect such a structure, with weightings reflecting that important concepts are at the top of the map, and less important at the bottom.

We also considered the applicability of topological analysis methods from other domains, in particular, Kleinberg's HITS algorithm [Kleinberg, 1999] for topological analysis of graphs, used to identify important nodes in a hyperlinked environment. Kleinberg's work characterized nodes on the World Wide Web as "hubs" and "authorities" based on their interconnections. When applied to concept maps, we expected hub and authority concepts to be especially important to determining the topic of concept maps.

The hypotheses underlying our use of topological analysis to assess concept descriptive power are the following:

1. Concepts that are closer to the root of a concept map are considered better descriptors of the topic of the map.

2. Concepts with higher connectivity are considered better descriptors of the topic of the map.

The models presented in the rest of this chapter provide the theoretical basis for answering the first of our research questions:

Research Question One: *How can topic descriptions be algorithmically extracted from non-standardized structured knowledge representations such as concept maps?*

Two of the proposed models are parameterized so that the actual contribution of hierarchical structure and connectivity—if any—can be determined empirically. In the following we present the three models. The evaluation of these models is presented in section 6.1, where we report on the results from an experiment conducted at Indiana University to study the fit of our models with human-subjects data.

Connectivity Root-Distance Model (CRD)

The connectivity root-distance model is based on two observations. First, concepts that participate in more than one proposition, as indicated by their connectivity—the number of incoming and outgoing connections—may be more important in defining a map's content than concepts with lower connectivity. Second, Novak argues that concept maps are best constructed if a "focus question" or a single root concept guides the selection of concepts and their hierarchical organization in the map. In his description on how to construct "good concept maps" Novak suggests that once a focus question has been formulated, the next step is to identify the key concepts that apply to the particular situation. "These could be listed, and then from this list a rank order should be established from the most general, most inclusive concept, for this particular problem or situation, to the most specific, least general concept." This suggests that the root concept, located at the top of a map, may be the most general and inclusive concept and that concept importance may increase with proximity to the root concept.

The CRD model calculates root proximity as the minimum number of direct links between the

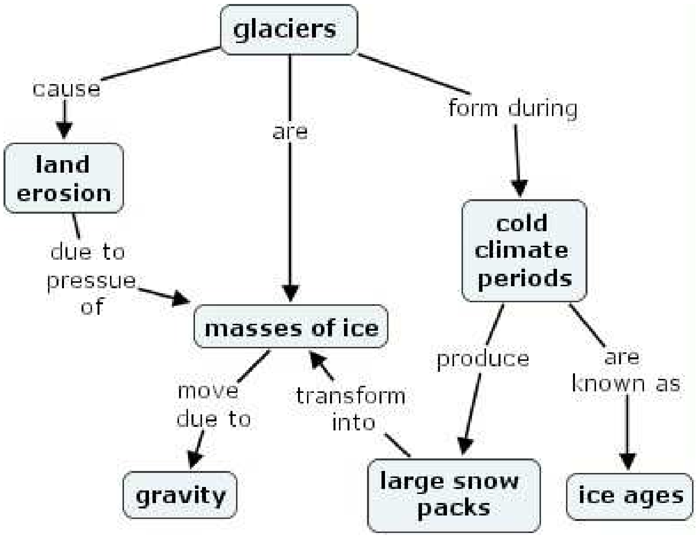


Figure 3.1: A simple concept map about glaciers.

map's root concept and a given concept. In addition, it determines the connectivity of each concept, by counting both the number of outgoing and incoming links. For example, in figure 3.1, the concept "masses of ice" has a connectivity of four (one outgoing and three incoming links) and a distance of one to the root concept "glaciers". If concept k in a map has o outgoing and i incoming connections to other concepts and is d steps distant from the root concept of the map, then the weight assigned to k by the CRD model is

$$W(k) = (\alpha \cdot o(k) + \beta \cdot i(k)) \cdot (1/(d(k)+1))^{1/\delta}$$

The model parameters α, β, and δ determine influence of the outgoing connections, incoming connections, and distance to the root concept. The formula implies that the higher a concept's connectivity and the shorter its distance to the root concept, the larger its weight and therefore relevance in the topic of the map.

An important characteristic of the CRD model is that each concept's connectivity weight can

```
PROCEDURE Compute-CRD-Weights
INPUT:
 G = (V,E): a concept map graph
 $\alpha$: the influence of outgoing connections
 $\beta$: the influence of incoming connections
 $\delta$: the influence of distance to the root
OUTPUT:
 w: a vector such that w[v] represents v's CRD weight
BEGIN
 r = Root(G)  % Return the root concept (we assume the map has a root)
 d = Minimum-Root-Distance(G,r)
 for each vertex v ∈ V[G]
 do
     i = Incoming(v)  % Return the number of edges (u,v) incoming to v
     o = Outgoing(v)  % Return the number of edges (v,u) outgong from u
     w[v] = ($\alpha$ * o + $\beta$ * i) * (1/(d[v]+1)) ^ (1/$\delta$)
 return w
END
```

Table 3.1: Pseudocode of the algorithm for computing the CRD weights.

be computed independently of the weights of other concepts in the map. As a consequence, these weights are based on local topology only, with positive computational cost effects. The procedures used to compute the CRD weights for a concept map are shown in table 3.1 and 3.2.

The most expensive part of this algorithm is the computation of each concept's minimum distance to the root, implemented by the procedure Minimum-Root-Distance. This procedure is an adaptation of Dijkstra's algorithm [Cormen et al., 1990], which solves the single-source shortest path problem on a graph in $O(n^2)$, where n is the number of vertices in the graph.

Hub Authority and Root-Distance Model (HARD)

The Hub Authority and Root-Distance Model also explores the importance of the root node and the hierarchical organization of concepts in maps. However, while CRD performs a local analysis, only taking immediate neighbors into account for computing a concept's connectivity,

```
PROCEDURE MINIMUM-ROOT-DISTANCE
INPUT:
  G = (V,E): a concept map graph
  r: the vertex representing the root of the concept map
OUTPUT:
  d: a vector such that d[v]=k if k is the minimum distance between r and v
BEGIN
  for each vertex $v \in V[G]$
  do
      $d[v] = \infty$
  $d[r] = 0$
  $Q = V[G]$
  while $Q \neq \emptyset$
  do
      u = EXTRACT-MIN(Q)  % delete and return the vertex from Q whose index is minimum
      for each edge$(u,v) \in E[G]$  % edges outgoing from u
      do
          if $d[v] > d[u] + 1$
          then
              $d[v] = d[u] + 1$
  return d
END
```

Table 3.2: Adaptation of Dijkstra's algorithm for computing each concept's minimum distance to the root.

HARD performs a global analysis on the influences of the concepts on each other. Its analysis centers on three different types of concepts that may be found in a concept map:

- *Hubs* are concepts that have multiple outgoing connections to authority nodes.
- *Authorities* are concepts that have multiple incoming connections from hub nodes.
- *Upper* nodes include the root concept and concepts closest to the root concept.

To determine a node's role as a hub or authority, we adapted Kleinberg's algorithm [Kleinberg, 1999] for analyzing hyperlinked graphs to concept maps. Our algorithm associates each concept with three weights between 0 and 1, each reflecting the concept's role as a hub, authority, or upper node. A given concept may simultaneously have properties of all three, but in

```
PROCEDURE COMPUTE-HARD-WEIGHTS
INPUT:
 G = (V,E): a concept map graph
 $\alpha$: the influence of hub weights
 $\beta$: the influence of authority weights
 $\gamma$: the influence of proximity to the root
OUTPUT:
 w: a vector such that w[v] represents v's HARD weight
BEGIN
 r = ROOT(G)  % Return the root concept (we assume the map has a root)
 [h,a] = HUBS-AUTHORITIES(G)
 d = MINIMUM-ROOT-DISTANCE(G,r)  % defined in the CRD algorithm
  for each vertex v ∈ V[G]
  do
     u = 1/ (d[v] + 1)
     w[v] = $\alpha$ * h(v) + $\beta$ * a(v) + $\gamma$ * u
  return w
END
```

Table 3.3: Pseudocode of the algorithm for computing the HARD weights.

Figure 3.1, “glaciers” is primarily a hub concept, due to the number of outgoing connections, and “masses of ice” is primarily an authority, due to its mostly incoming connections. Among the three concepts with outgoing links to the concept “masses of ice”, “glaciers” is the one with the greatest influence in making “masses of ice” an authority node, because of the comparative strength of “glaciers” as a hub.

In the HARD model, the three weights of a selected concept k are combined into a single weight as follows:

$$W(k) = (\alpha \cdot h(k) + \beta \cdot a(k) + \gamma \cdot u(k))$$

In the above formula h, a, and u are the corresponding hub, authority, and upper node weights of a concept in a map and α, β, and γ are the model parameters. As above, the parameters reflect the influences of the different roles that a concept may play. The procedures used to compute the HARD weights in a concept map graph are outlined in tables 3.3, 3.4 and 3.5.

```
PROCEDURE HUBS-AUTHORITIES
INPUT:
  G = (V,E): a concept map graph
OUTPUT:
  h: a vector with hub-weight values
  a: a vector with authority-weight values
BEGIN
  for each vertex v ∈ V[G]
  do
      h[v] = 1
      a[v] = 1
      $h_0[v] = 0$
      $a_0[v] = 0$
  while ($h_0 \neq h$) or ($a_0 \neq a$)
  do
      $a_0 = a$
      $h_0 = h$
      a = SUM-IN(G,$h_0$); h = SUM-OUT(G, $a_0$)
      NORMALIZE(a)  % normalize vector a so that $\sum_v a(v)^2 = 1$
      NORMALIZE(h)  % normalize vector h so that $\sum_v h(v)^2 = 1$
  return h, a
END
```

Table 3.4: Adaptation of Kleinberg's algorithm for computing Hubs and Authories.

The most costly part of this algorithm is the HUBS-AUTHORITIES procedure. The iterative method used to compute the hub and authority weights is guaranteed to converge in at most n steps, where n is the number of vertices in the graph representation of the concept map [Kleinberg, 1999]. This fact, combined with the doubly nested loop structure of the SUM-IN and SUM-OUT procedures yields an $O(n^3)$ upper bound on the worst-case running time of this algorithm.

Path Frequency Model (PF)

The Path Frequency Model, like the CRD model, reflects the expectation that concepts participating in more propositions will tend to be more important to the topic of a map. However, instead of considering only a concept node's immediate connectivity, like the CRD model, the PF model

```
PROCEDURE Sum-In
INPUT:
  G = (V,E): a concept map graph
  h: a vector with in-progress hub-weight values
OUTPUT:
  a: a vector with new computed authority-weight values
BEGIN
  for each vertex $v \in V[G]$
  do
      $a[v] = 0$
      for each edge $(u,v) \in E[G]$  % edges incoming to v
      do
          $a[v] = a[v] + h[u]$
  return a
END

PROCEDURE Sum-Out
INPUT:
  G = (V,E): a concept map graph
  a: a vector with in-progress authority-weight values
OUTPUT:
  h: a vector with new computed hub-weight values
BEGIN
  for each vertex $v \in V[G]$
  do
      $h[v] = 0$
      for each edge $(v,u) \in E[G]$  % edges outgoing from v
      do
          $h[v] = h[v] + a[u]$
  return h
END
```

Table 3.5: Auxiliary procedures for computing Hubs and Authorities.

considers indirect relationships as well. It counts all possible paths, starting from the root concept, that contain the concept in question and either (1) end on a concept with no outgoing connections, or (2) end on a concept that has already been visited in that path.

The weight $W(k)$ of a concept k in a map is the number of paths crossing k. Unlike the previous two models, this model considers only a single influence on concept weight, and consequently requires no parameters.

We note that if a concept has high connectivity (which allows for many paths to form in the

```
PROCEDURE Compute-PF-Weights
INPUT:
 G = (V,E): a concept map graph
OUTPUT:
 w: a vector such that w[v] represents v's PF weight
BEGIN
 r = Root(G)  % Return the root concept (we assume the map has a root)
 for each vertex v ∈ V[G]
 do
     w[v] = 0
     visited   w = Find-Paths(G,r,visited,w)
 return w
END
```

Table 3.6: Pseudocode of the algorithm for computing the PF weights.

map), then the number of paths crossing a concept also increases for concepts indirectly linked to the high-connectivity concept. For example, the PF value for the concept "gravity" in figure 3.1 is three, because there are three paths extending from the root concept to "gravity," due to "masses of ice" which is well connected in the map.

Due to the hierarchical structure of concept maps, concepts that are closer to the root tend to participate in more paths. In particular, the root concept participates in all possible paths in a map and as a consequence it receives the highest PF weight. The procedures used to compute the PF weights of a concept map are presented in tables 3.6 and 3.7.

The theoretical upper bound on PF time complexity is $O(n!)$, where n is the number of vertices in the concept map graph. In practice, however, due to the sparse nature of graphs representing concept maps, the cost of computing the PF weights is usually much smaller than this upper bound.

Each of the three models presented in this section applies distinguishing mechanisms to model concept importance in concept maps, but nonetheless they all share the central idea that topology

```
PROCEDURE FIND-PATHS
INPUT:
  G = (V,E): a concept map graph
  v : a concept from which the search for paths begins
  visited: a vector such that visited[v] =1 if v has been visited, and visited[v] =0 otherwise
  w: a vector such that w[v] represents the number of paths in which v participates so far
OUTPUT:
  w: a vector such that w[v] represents the number of paths in which v participates so far
BEGIN
  if visited[v] = 1
  then % A cycle was found, update and finish
      w= w + visited
  else % The vertex v has not been visited
      visited[v] = 1
      if there is no edge (v,u) $\in$ E(G)
      then % v has no outgoing connections, update and finish
          w = w + visited
      else % v has outgoing connections, continue searching for paths
          for each edge (v,u) $\in$ E[G]  % edges outgoing from v
          do
              w = FIND-PATHS (G,u,visited,w)
  return w
END
```

Table 3.7: Procedure for counting how many paths cross each concept in a concept map

is important to assess concept descriptive power. In particular, they are all based on the premises that (1) concepts that are closer to the root of a concept map are better descriptors of the topic of the map, and (2) concepts with higher connectivity are better descriptors of the topic of the map.

In section 6.1 we will provide empirical evidence, supporting the effectiveness of our topological models in predicting human's judgments of concept importance in concept maps.

4

Context-Based Topic Search

The extraction of descriptors from a concept map, which was the focus of chapter 3, is important because a small set of terms with high descriptive power can convey the topic of the map to a human. However, the task of identifying good descriptors is distinct from identifying good query terms for retrieving related information. When providing support for knowledge extension, other terms may be effective cues for retrieving topic-relevant documents, but they may not be good descriptors or may not even be present in the map.

This chapter develops a framework for the dynamic identification of "good query terms" to aid topic search in the context of a knowledge model under construction. We begin by discussing classical approaches to information retrieval and their limitations when applied to the problem of context-based topic search on the Web. Then, we review work on Web mining and topic extraction that relates to our work. After this review, we describe our theoretical framework for addressing the query formation and topic identification problems.

4.1 Information Retrieval and Web Search

The World Wide Web provides a rich source of information on potential new topics to include in a knowledge model. To access relevant information, appropriate queries must be formed. In text-based Web search, users' information needs and candidate text resources are typically characterized by terms.

Substantial experimental evidence supports the effectiveness of using weights to reflect relative term importance for traditional information retrieval (IR) [Salton and Yang, 1973, Salton and Buckley, 1988]. The main purpose of a term weighting system is the enhancement of retrieval effectiveness.

Recall and Precision

Effective retrieval depends on retrieving those items that are likely to be relevant to the user's needs, but also on filtering irrelevant material. In order to assess the ability of a system to retrieve relevant items and reject the irrelevant ones, the IR community normally uses two measures, known as *recall* and *precision*.

Given an information request and its set of relevant documents R, assume that a given retrieval strategy generates a document answer set A. The recall and precision measures are defined as follows [Baeza-Yates and Ribeiro-Neto, 1999]:

- **Recall** is the fraction of relevant documents (the set R) which has been retrieved, i.e.,

$$\textbf{Recall} = \frac{|R \cap A|}{|R|}$$

- **Precision** is the fraction of retrieved documents (the set A) which is relevant, i.e.

$$\textbf{Precision} = \frac{|R \cap A|}{|A|}$$

The recall measure, as defined above, assumes that we have access to $|R|$, the number of relevant documents. For a large and dynamic corpus, such as the Web, it is impossible to determine this number. Approximations for the recall and precision measures for the Web domain have been proposed in a number of studies (e.g, [Saracevic, 1995, Chu and Rosenthal, 1996, Wishard, 1998, Srinivasan et al., 2004]).

In principle, a system is preferred that produces both high recall and high precision. To serve recall and precision, conventional IR scheme use composite term weighting factors that contain both recall- and precision-enhancing components. However, as has been discussed by a number of sources, issues arise when attempting to apply conventional IR schemes for measuring term importance to systems for searching Web data [Kobayashi and Takeda, 2000, Belkin, 2000]. One difficulty is that methods for automatic query formation for Web search do not have access to a full predefined collection of documents, raising questions about the suitability of classical IR schemes for measuring term importance when searching the Web. A central question addressed in our work is how to formulate topic descriptors and discriminators to guide context-based topic search on the Web.

The Classical View of Descriptors and Discriminators

The IR community has investigated the roles of terms as descriptors and discriminators for several decades. Since Sparck Jones' seminal work on the statistical interpretation of term specificity [Jones, 1972], term discriminating power has often been interpreted statistically, as a function of term use. Similarly, the importance of terms as content descriptors has been traditionally estimated by measuring the frequency of a term in a document.

The combination of descriptors and discriminators gives rise to schemes for measuring term relevance such as the familiar *term frequency inverse document frequency* (TF-IDF) weighting model [Salton and Yang, 1973]. TF-IDF is a simple way to measure the relevance of a term for a document relative to a collection. Relevance according to the TF-IDF scheme is determined by two quantities:

- **Term Frequency**. Given a document d and a term t, the *term frequency* is simply measured as the number of times term t occurs in document d:

  $$\mathit{TF(d,t)} = n(d,t)$$

- **Inverse Document Frequency**. Given a term t and a collection D of documents, the *inverse document frequency* measure varies inversely with the number of documents to which a term is assigned. In its common form, *inverse document frequency* is defined as follows [Salton and Yang, 1973]:

  $$\mathit{IDF(t)} = \log \frac{1 + |D|}{|D_t|}$$

  where $|D_t|$ represents the number of documents in D containing term t.

Term frequency factors help to achieve high recall. However, term frequency alone cannot insure acceptable precision because high frequency terms may also occur in irrelevant documents.

Hence inverse document frequency performs the function of penalizing those terms that lack discriminating power. TF and IDF are combined to form the TF-IDF measure as follows:

$$\textit{TF-IDF(d,t)} = \textit{TF(d,t)} \times \textit{IDF(t)}$$

New Challenges for Information Retrieval

The TF-IDF scheme is a reasonable measure of term importance but is insufficient for the task domain for our research. Searching the Web to support knowledge extension presents new challenges for formulation of descriptors and discriminators. Specifically, making full use of the information available in knowledge models requires:

- **Search methods that can reflect extensive contextual information** (instead of attempting to summarize context in a small number of weighted terms). For knowledge model extension, the knowledge model under construction provides a rich context that can be exploited for information filtering, term-weight reinforcement, and query refinement. Because search engines may restrict queries to a small number of terms (e.g., the 10-term limit for Google), incremental approaches may be needed to fully reflect search context.

- **Methods for topic search** (instead of document search). Users selecting topics to include in a knowledge model will be aided by search methods which directly generate characterizations of possible topics—which may span individual documents—rather than simply presenting sets of documents. In traditional IR approaches, term discriminating power is based on the overall rarity of a term in a document collection, rather than on term distribution across different topics. For example, the term discrimination value under the TF-IDF model expresses the goodness of a term in discriminating a *document*, as opposed to discriminating the *topic* of the document. Mining topics requires new measures for term discrimination.

- **Methods for searching open collections of documents** (instead of a pre-defined and pre-analyzed collection). In Web-based knowledge extension tasks, the search space is the full Web, and analysis must be limited to a small collection of documents—incremental retrievals—that is built up over time and changes dynamically. Unlike traditional IR schemes, which analyze a predefined collection of documents and search that collection, Web-based knowledge extension must rely on methods that use limited information to assess the importance of documents and to manage decisions about which documents to retain for further analysis, which ones to discard, and which additional queries to generate.

Before introducing our framework for context-based topic search on the Web, we present a brief review on the most relevant work in the areas of Web mining and topic extraction.

Web Mining and Topic Extraction

Web mining is the process of extracting knowledge and patterns from the Word Wide Web. The Web is massive, dynamic and diverse, presenting interesting challenges for developing systems aimed at exploiting the rich information sources it provides. Despite the fact that extracting useful information from the Web is to a great extent more complex than dealing with standardized information sources, such as databases, important advances have been made based on the *Structured Web Hypothesis* [Etzioni, 1996b], which states that "information on the Web is sufficiently structured to facilitate effective Web mining."

Numerous Web agents have been developed to facilitate Web mining and topic extraction. Some of these agents, such as the SoftBots [Etzioni and Weld, 1994, Etzioni, 1996a] operate on top of Internet tools and services, with the purpose of abstracting away the technology underlying the accessed resources. The kind of Web agents known as Web crawlers [Pant et al., 2004] exploit the graph structure of the Web to follow hyperlinks, discover resources, and map them into searchable index structures. Some Web crawlers are exhaustive, and perform an extensive exploration of the resources available online, independently of a pre-defined set of topics. Other Web crawlers are topical or focused [Chakrabarti et al., 1999c, Menczer et al., 2004], in which case the mining process is guided not only by following existing links but also by considering content to focus on pages relevant to a specific theme.

Web mining is divided in three main categories [Kosala and Blockeel, 2000] identified as Web content mining, Web structure mining, and Web usage mining. The third category, Web usage mining, deals with the extraction of Web navigational trends and patterns with the purpose of predicting user behavior. The extracted data can be used to reduce response time in the Web environment as well as to improve Web site design and navigation opportunities. Overviews of research on this area can be found in Borges et al. (1999), Srivastava et al. (2000), Cooley's PhD thesis (2000), and Eirinaki and Vazirgianni (2003).

Web Content Mining

Our work on context-based topic search relates to work on Web content mining and Web structure mining. Much of the existing work on Web content mining builds on long-established areas of research, including information retrieval, natural language processing, databases, and machine learning. Web content mining usually combines *text-mining* and *intra-document structure mining* techniques.

Text-mining is performed by looking at document's text-data to identify salient features, which are extracted and employed to create indices, or to fill in data structures (e.g., vector representations) or databases. Text-mining algorithms draw on a range of methods such as automatic text-learning [Mladenic, 1999], text categorization [Sebastiani, 2003], clustering [Everitt, 1980] and latent semantic indexing [Deerwester et al., 1990], among many others.

Instead of merely exploiting text-data, intra-document structure mining approaches also take advantage of the additional structural information (e.g., tags and hyperlinks) existing in semi-structured data. Semi-structured data, sometimes called self-describing data, has a series of distinguishing characteristics [Abiteboul et al., 1997, Abiteboul et al., 2000]. In a different way from rigidly structured data that is normally constrained by an a-priori schema, semi-structured data is only bond to an a-posteriori data guide, which provides indication of an implicit, partial and irregular structure. Currently, HTML documents are the most highly disseminated forms of semi-structured data. HTML is a document markup language that uses predefined tags for presentation purposes and not to convey semantics. In spite of that, various approaches have demonstrated that HTML tags can be usefully exploited to extract meaningful content [Doorenbos et al., 1997] and to develop wrappers [Ashish and Knoblock, 1997, Kushmerick et al., 1997], which are programs that provide database like interfaces to HTML sources. A proposal worth noticing is the Web KB system [Craven et al., 2000], which, guided by an ontology and relations of interest, is trained to mine HTML pages and extract symbolic information that is added to a large knowledge base.

HTML has been extended in different ways, to support automatic extraction of information from semi-structured data. XML, in a different way from HTML, is a data interchange format where the tags describe meta-information, commonly used to supply semantics. This facilitates the extraction of content but introduces a number of issues due to the fact that XML only provides a data format for documents, without a predefined vocabulary, data types or data interpretation. Document Type Definition (DTD), XML Schemas, and Ontology Languages such as RDF and its extension DAML+OIL have been introduced with the purpose of addressing some of these issues [Klein, 2001] and to contribute to the realization of a Semantic Web. The content of a Semantic Web is expected to be meaningful and tractable not only by Web mining agents but also by reasoning engines.

Web Structure Mining

The second Web mining category, Web structure mining, deals with the structure of the hyperlinks within the Web, hence with the inter-document structure. Modeling the Web as a huge graph, where the pages represent nodes and the hyperlinks edges, admits the implementation of mathematically clean connectivity analysis methods. The main premise behind the application of connectivity analysis on the Web graph is that authoritativeness, in addition to relevance is desired in search results. Popularity has been taken as the principal emissary of authoritativeness; hence, techniques borrowed from social network and citation analysis theory have been used to discover authority sites (most popular pages) and hubs (access point to good authority pages).

One of the goals of the EXTENDER system is to provide topics that facilitate access to authoritative sites. EXTENDER produces topics associated with authoritative Web pages as a by-product of our use of Google Web API service to search the Web. In order to estimate the importance of Web sites, the Google search engine uses PageRank [Brin and Page, 1998] as a component of its search-result ranking mechanisms.

PageRank and HITS

PageRank provides an objective measure of the popularity of Web pages based on the probability that an idealized Web surfer jumps to a Web page as the result of a random walk on the Internet graph. The PageRank measure is estimated by means of a recursive formula based on the amount of incoming hyperlinks to a page, while recurrently considering the rank of the pages from which the links come. This rank is assigned to Web pages based solely on connectivity analysis, and is independent from the content of the pages. A search on Google returns pages sequentially ordered in terms of a measure that combines content relevance (between query and page) and the pre-computed PageRank score.

Another prominent algorithm that uses connectivity analysis to estimate the importance of a Web site for a particular query is HITS [Kleinberg, 1999] (which was briefly discussed in section 3.3, in connection to the problem of finding important concepts in a concept maps). Instead of pre-processing the whole Web graph structure, HITS operates on focused subgraphs that result from extending the outcome of a query presented to a search engine. One of the motivations underlying the HITS algorithm was the observation that, at the time HITS was proposed, a typical search on the Internet might not return the most authoritative pages relevant to a query. However, a search was likely to return at least one result with a link to some authoritative page. The algorithm, therefore, expands the results returned by a search engine by adding pages containing links that enter or leave any of the pages in the initial set. This is followed by the application of an iterative algorithm aimed at identifying the authoritative pages in the expanded graph of pages. The algorithm associates with each page i two weights h_i and a_i, standing for hub weight and authority weight of the page. Important authorities are those that have links from important hub, whereas important hubs are expected to have links to multiple relevant authorities. Hub and authorities reinforce each other, and by means of a convergent cross-recursive algorithm it is possible to compute the h_i and a_i weights for each node i in a graph. HITS algorithm generates a graph expansion and performs connectivity analysis after the query is presented and therefore is slower than Google, which utilizes pre-computed ranks.

Combining Content and Link Information

In hyperlinked environments, keywords non-local to a document extracted from text associated with incoming links have been used to augment the description of resources and to improve retrieval [Salton, 1963, Kwok, 1985, Croft and Turtle, 1989, Frei and Stieger, 1992]. These ideas and some variants have been exploited more recently in work on automatic resource categorization [Chakrabarti et al., 1998a, Chakrabarti et al., 1998b] and on indexing digital libraries based on reference [Bradshaw et al., 2000]. The Clever system [Chakrabarti et al., 1999a, Chakrabarti et al., 1999b] incorporates heuristics that combine content extracted from anchor text with link information, resulting in an improvement on the HITS algorithms. To avoid topic contamination or drift, Clever computes a matching measure between the anchor text and the target query and uses that measure to weight the edges of the extended graph.

Bharat and Henzinger (1998) also address the problem of topic contamination by proposing a collection of algorithms that improve on the results delivered by HITS. Their algorithms implement content analysis of online documents with the purpose of pruning the graph to be distilled. The pruning of the graph is carried out by discarding those nodes whose similarity to the pages directly retrieved from the search engine is below a certain threshold.

In order to implement good quality and efficient connectivity analysis methods, it is of primary importance to have effective access to the Web graph structure. Many proposals have addressed this issue, providing services and tools for storing and manipulating sets of URLs and portions of the Web graph [Bharat et al., 1998, Randall et al., 2001, Suel and Yuan, 2001, Guillaume and Latapy, 2002, Boldi and Vigna, 2003, Raghavan and Garcia-Molina, 2003].

Topic Identification and Extraction

It has long been recognized that the hyperlink structure of the Web can help to discover Web communities, which often lead to the extraction of topically coherent subgraphs. Many algorithms based solely on link information have been proposed to partition hypertext environments [Hara and Kasahara, 1990, Bernstein et al., 1991, Hara et al., 1991, Botafogo and Shneiderman, 1991, Botafogo et al., 1992, Botafogo, 1993, Pitkow and Pirolli, 1997] and to identify and examine the structure of topics on the Web [Gibson et al., 1998, Dill et al., 2001, Chakrabarti et al., 2002]. Other algorithms, such as Companion and Cocitation, use the hyperlink structure of the Web to find related pages [Dean and Henzinger, 1999].

While "link-only" approaches often provide a good indication of relatedness, the incorporation of textual signals can considerably improve methods for grouping similar resources and discovering relevant sites. HyPursuit [Weiss et al., 1996] is an early example of a system that combines link and content structure to cluster hypertext. Pirolli et al. (1996) exploit usage statistics and page meta-information to associate types with Web sites according to their role and purpose (e.g., head organizational home page, head personal home page, index, reference, etc.) and for enhancing clustering and relevance assessments.

Marchiori (1997) discusses the idea of hyper search engines as systems that combine textual and hyper-information content to increase the precision of current search engines. Chen (1997) presents an approach called Generalized Similarity Analysis (GSA) that combines hypertext linkage, content similarity and usage patterns to define proximity relations. Proximity data underlying patterns are represented spatially using Pathfinder Networks [McDonald et al., 1990]. Modha and Spangler (2000) introduce the toric k-means algorithm as a geometric hypertext clustering algorithm where similarity between documents is defined in terms of features extracted from the document textual content, out-links and in-links.

Organizing Search Results into Meaningful Groups

Our research on topic extraction also shares insights and motivations with proposals aimed at clustering search results (e.g., [Cutting et al., 1992, Hearst and Pedersen, 1996, Anick and Vaithyanathan, 1997, Kaski et al., 1998, Zamir and Etzioni, 1999, Chen and Dumais, 2000]) and refining queries (e.g., [Chen and Dhar, 1990, Vélez et al., 1997, Anick and Tipirneni, 1999, Oyama et al., 2001]). However, differently from our proposals, these systems provide browsing interfaces in which the user's intervention must be explicit. In addition, their goal is to help users to focus on specific information and to remove alternatives rather than to discover novel but related material.

In the remainder of this chapter we discuss the theoretical framework we have developed for topic generation. The application of the framework in the implementation of the EXTENDER system will be discussed in chapter 5.

4.2 A Framework for Topic Generation

Topics group documents related by a common theme. One way to represent topics is implicitly, as sets of related documents. Alternatively, a topic can be represented as a set of cohesive terms summarizing the topic content. Some terms may have strong descriptive power, enabling a small set to convey the topic to a human. As we have discussed in earlier sections, some terms may be effective cues for retrieving topic-relevant documents, but may not be good descriptors. Consider for example a topic involving exploration of Mars, described by the following set of terms occurring in documents related to Mars exploration:

Mars	Exploration	Rover	Landing	Site
Selection	Opportunity	Spirit	Images	Global
Surveyor	Orbiter	Camera	MGS	MOC

The terms *Mars* and *Exploration* are good descriptors of the topic for a general audience. Terms such as *MGS* and *MOC*—which stand for "Mars Global Surveyor" and "Mars Orbiter Camera"—may not be good descriptors of the topic for that audience, but are effective in bringing information similar to the topic when presented in a query.

This suggests that the importance of a given term depends on the task at hand; the notion of term importance has different nuances depending on whether the term is needed for query construction, index generation, document summarization or similarity assessment. For example, a term which is a useful descriptor for the content of a document, and therefore useful in similarity judgments, may lack discriminating power, rendering it ineffective as a query term, due to low precision of search results, unless it is combined with other terms which can discriminate between good and bad results.

Intuitively, we can characterize topic descriptors and discriminators as follows:

- Terms are *good topic descriptors* if they answer the question "What is this topic about?"
- Terms are *good topic discriminators* if they answer the question "What are good query terms to access similar information?"

In this section, we develop a framework for addressing the second of our research questions:

Research Question Two: *How can knowledge models be used to characterize information requirements and to discover novel but relevant topics of potential interest that the user may want to include in the knowledge model?*

Our hypothesis, evaluated in section 6.2, is that terms that tend to occur frequently in the context of a given topic tend to be good topic descriptors. Thus a possible strategy for finding good topic descriptors is to (1) find documents that are similar to other documents already known to have that topic, and (2) select from those documents the terms that occur often.

On the other hand, a term is a good discriminator for a topic if most documents that contain that term are topically related. Thus finding good topic discriminators requires finding terms that tend to occur only in the context of the given topic.

Both topic descriptors and discriminators are important as query terms. Because topic descriptors occur often in relevant pages, using them as query terms may improve recall. Because good topic discriminators occur primarily in relevant pages, using discriminators as query terms may improve precision. The following sections transform the above informal characterizations of topic descriptors and discriminators into precise definitions and apply them to the task of mining the Web for context-related topics.

Using Hypergraph Representations for Documents and Terms

Determining topic discriminators and descriptors requires analyzing the interplay between terms, documents and topics. We propose hypergraphs [Berge, 1973] as a natural way to represent such relationships. A hypergraph is a generalization of a graph, in which each edge (hyperedge) is represented as a multiset of nodes.

If we disregard the structure of text documents, we can view any collection of documents as a hypergraph $H = (T, \mathcal{D})$, where each node $t \in T$ corresponds to a term and each hyperedge $d \in \mathcal{D}$ corresponds to a document. A hyperedge d is a multiset with elements in T, representing the abstraction of a document as a bag of terms. We call this a *document-centered hypergraph*. As a dual to this view, we can think of a term as a multiset whose elements are those documents in which the term occurs. Therefore, for each document-centered hypergraph $H = (T, \mathcal{D})$, there corresponds a *term-centered hypergraph* $H^* = (D, \mathcal{T})$ whose nodes correspond to documents and whose hyperedges correspond to terms, represented as multisets of documents. Hypergraph H^* is called the dual hypergraph of H. Figures 4.1(a) and 4.1(b) illustrate a hypergraph representation for a collection of three documents, A, B, and C, each represented as a multiset, containing some of the terms 1, 2, 3 and 4. This collection can be represented by the document-centered hypergraph $H = (\{1,2,3,4\}, \{A,B,C\})$ (with $A = \{1,1,2\}, B = \{2,2\}$ and $C = \{2,3,4\}$) or by its dual $H^* = (\{A,B,C\}, \{1,2,3,4\})$ (with $1 = \{A,A\}, 2 = \{A,B,B,C\}, 3 = \{C\}$ and $4 = \{C\}$). In figures 4.1(a) and 4.1(b), circles represent hyperedges and triangles represent nodes. The value associated with the connection between a node and a hyperedge stands for the number of occurrences of the node in the hyperedge. For example, the value 2 associated with the connection between node 1 and hyperedge A in figure 4.1(a) denotes that term 1 occurs twice in document A.

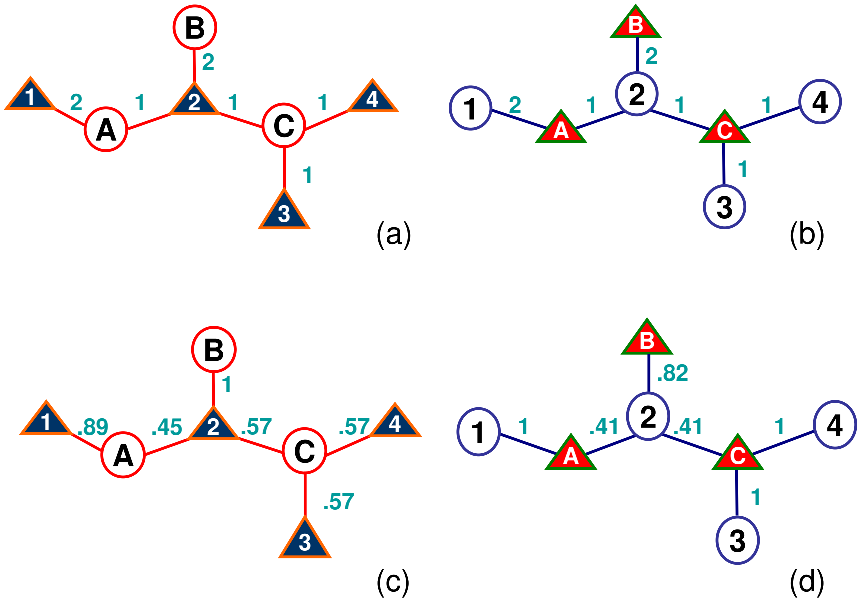


Figure 4.1: (a) hypergraph H; (b) hypergraph H^*; (c) and (d) the hypergraphs' weighted version.

The incidence matrix of a document-centered hypergraph $H = (T, \mathcal{D})$ for a collection of m documents and n terms is a matrix $\mathbf{H}$ with m rows that represent the documents (hyperedges of H) and n columns corresponding to the terms (nodes of H) such that

$$\mathbf{H}[i, j] = k$$

where k is the number of occurrences of t_j in d_i. Note that the incidence matrix of the dual hypergraph H^* is the transpose of the incidence matrix of hypergraph H.

Representing the relationships between terms and documents using hypergraphs forms the basis for our analysis of a series of dual notions. These dualities arise at various levels, and can be interpreted as reflecting interesting properties of terms and documents leading to our characterization of topic descriptors and discriminators.

Document Descriptors and Discriminators

We use the adjacency matrix $\mathbf{H}$ of a document-centered hypergraph to define functions corresponding to the notions of term descriptive power and term discriminating power in a document. Term descriptive power in a document is modeled by a function $\lambda : \{d_0, \ldots, d_{m-1}\} \times \{t_0, \ldots, t_{n-1}\} \to [0, 1]$ that maps a document-term pair into a value in the unit interval. It is defined as follows:

$$\lambda(d_i, t_j) = \frac{\mathbf{H}[i, j]}{\sqrt{\sum_{k=0}^{n-1} (\mathbf{H}[i, k])^2}}.$$

Function λ can be used to construct a *document-centered weighted hypergraph* (which we will call a d-hypergraph) in which the descriptive power of term t_j in document d_i is used as the weight of node t_j in hyperedge d_i. In figure 4.1(c) we can see a d-hypergraph in which terms have different descriptive power for their associated documents. In particular, document B is entirely described by term 2.

The second function $\delta : \{t_0, \ldots, t_{n-1}\} \times \{d_0, \ldots, d_{m-1}\} \to [0, 1]$ is used to model discriminating power of a term in a document. If we define $\mathrm{s}(k)$, to return 1 if $k > 0$ and 0 if $k = 0$, we define δ as follows:

$$\delta(t_i, d_j) = \frac{\mathrm{s}(\mathbf{H^T}[i, j])}{\sqrt{\sum_{k=0}^{m-1} \mathrm{s}(\mathbf{H^T}[i, k])}}.$$

Function δ maps a term-document pair into a value in the unit interval. If term t_i does not occur in document d_j then $\delta(t_i, d_j) = 0$. On the other extreme, if term t_i occurs in no document other than d_j, then $\delta(t_i, d_j) = 1$ and we say that t_i fully discriminates d_j.

Discriminating power of a term in a document is independent of the number of occurrences of the term in the document. If k represents the number of occurrences of a term in a document, function δ will only consider $\mathrm{s}(k)$, disregarding the total number of occurrences and considering only whether or not a term is in a document.

Function δ can be used to construct a *term-centered weighted hypergraph* (t-hypergraph) where the discriminating power of term t_i in document d_j is the weight of node d_j in hyperedge t_i. In figure 4.1(d), term 1 fully discriminates document A.

Both for d-hypergraphs and t-hypergraphs, the square of the weights associated with each hyperedge sum to 1, i.e.,

$$\sum_j (\lambda(d_i, t_j))^2 = 1 \quad \text{and} \quad \sum_j (\delta(t_i, d_j))^2 = 1.$$

It is easy to verify that the weighted hypergraphs will continue to be duals structurally, but in general they will not preserve the numerical duality. Consequently, the new associated incidence matrices will not be transposes of each other.

As is the case with other IR characterizations of descriptors and discriminators, the notions discussed above only allow discovering terms that are good descriptors or discriminators of a *document*, as opposed to good descriptors or discriminators of the *topic of a document*. The IR community has been aware of this limitation and introduced different heuristic to tackle the problem. A simple heuristic is to eliminate terms that are too rare or too common [Kira and Rendell, 1992].

While useful to a certain extent, these heuristics have been criticized because they do not exhibit well substantiated theoretical properties and they depend on artificially defined thresholds for term exclusion. In the next sections, we build on the notions of document descriptors and discriminators to identify higher-order relationships between documents and terms and to provide new definitions of descriptors and discriminators. These new definitions make the notions of descriptors and discriminators topic-dependent.

Similarity and Co-occurrence

To address the problem of identifying terms that are good descriptors or good discriminators of a topic, we first need to characterize the notion of *topic*. We treat topics as defined by either a collection of similar documents or a collection of terms that tend to co-occur. Thus the notions of document similarity and term co-occurrence play important roles in identifying topics.

The similarity between documents d_i and d_j can be computed using the well-known cosine measure as follows:

$$\begin{aligned}\sigma(d_i, d_j) &= \frac{\sum_{k=0}^{n-1}(\lambda(d_i,t_k)\cdot\lambda(d_j,t_k))}{\sqrt{\sum_{k=0}^{n-1}(\lambda(d_i,t_k))^2\cdot\sum_{k=0}^{n-1}(\lambda(d_j,t_k))^2}} \\ &= \sum_{k=0}^{n-1}(\lambda(d_i,t_k)\cdot\lambda(d_j,t_k)).\end{aligned}$$

The idea of *term co-occurrence* captures a relation between terms that is dual to the notion of document similarity. A measure of co-occurrence for terms t_i and t_j can be obtained as follows:

$$\kappa(t_i, t_j) = \frac{\sum_{k=0}^{m-1}(\delta(t_i,d_k)\cdot\delta(t_j,d_k))}{\sqrt{\sum_{k=0}^{m-1}(\delta(t_i,d_k))^2\cdot\sum_{k=0}^{m-1}(\delta(t_j,d_k))^2}}$$

$$= \sum_{k=0}^{m-1}(\delta(t_i, d_k) \cdot \delta(t_j, d_k)).$$

Figure 4.2(a) presents a simple illustration of the notion of document similarity by means of a d-hypergraph. In this example we can see that documents D and E are similar. Figure 4.2(b) shows the corresponding t-hypergraph in which it is easy to see that terms 3 and 4 co-occur.

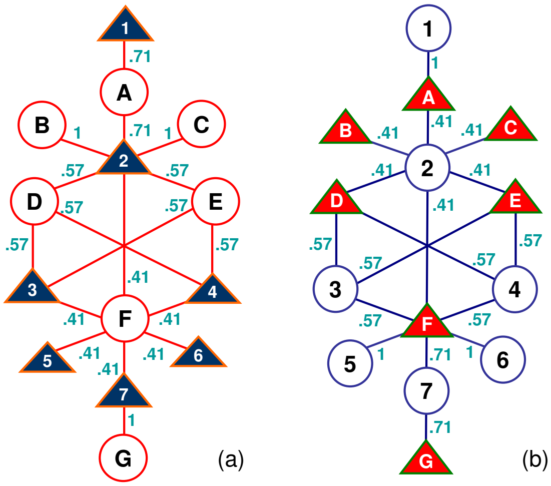


Figure 4.2: Weighted hypergraphs illustrating a series of dual notions: document similarity, term co-occurrence, topic discriminators, topic focus, topic descriptors and topic exhaustivity.

Topic Discriminators and Topic Focus

By examining document-term duality, we can develop higher-order notions useful for identifying good topic descriptors and discriminators. A term is a *good discriminator of a document's topic* if those documents discriminated by the term are similar to the given document. This intuition can be formally expressed using the function $\Delta : \{t_0, \ldots, t_{n-1}\} \times \{d_0, \ldots, d_{m-1}\} \rightarrow [0, 1]$ defined as follows:

$$\Delta(t_i, d_j) = \sum_{\substack{k=0 \\ k \neq j}}^{m-1} (\delta(t_i, d_k)^2 \cdot \sigma(d_k, d_j)).$$

We can think of the discriminating power of term t_i for the topic of document d_j as the average of the similarity to d_j of other documents discriminated by t_i. Note that even in the case when d_j does not contain t_i, the value of the function $\Delta(t_i, d_j)$ will not necessarily be 0. On the other hand, if no other document similar to d_j contains t_i, i.e., $\sigma(d_k, d_j) = 0$ or $\delta(t_i, d_k) = 0$ for all documents d_k containing t_i with $k \neq j$, then t_i has no discriminating power over the topic of d_j and as a consequence $\Delta(t_i, d_j) = 0$.

We have previously discussed the dual notions of document similarity and term co-occurrence. At this stage we might ask what would be the dual notion to "term discriminating power in a topic." This would be a function comparable to Δ but applicable to documents rather than terms. We can think of *document focus* as a property of documents that plays a role dual to that of *term discriminating power*. A document is focused on the topics associated with a term if the terms describing the document tend to co-occur with the given term. Formally, we can compute the degree of focus of a document on the topic identified by a term as a function $\Phi : \{d_0, \ldots, d_{m-1}\} \times \{t_0, \ldots, t_{n-1}\} \rightarrow [0, 1]$ defined as follows:

$$\Phi(d_i, t_j) = \sum_{\substack{k=0 \\ k \neq j}}^{n-1} (\lambda(d_i, t_k)^2 \cdot \kappa(t_k, t_j)).$$

Note that we have defined the higher-order dual notions of topic discriminators and topic focus by means of more basic dual notions. Term discriminating power in a topic has been defined using the notions of term discriminating power in a document and document similarity. Analogously, the measure of document focus on a topic has been defined via term descriptive power in a document and term co-occurrence.

Topic Descriptors and Topic Exhaustivity

The notion of *topic descriptors* was informally defined earlier as terms that occur *often* in the context of a topic. The *descriptive power* of a term in a topic is a measure that can be computed using the previously defined measures of document similarity and term descriptive power in documents. We measure *term descriptive power in the topic of a document* as a function $\Lambda : \{d_0, \ldots, d_{m-1}\} \times \{t_0, \ldots, t_{n-1}\} \rightarrow [0, 1]$:

$$\Lambda(d_i, t_j) = \begin{cases} 0 & \text{if } \sum_{\substack{k=0 \\ k \neq i}}^{m-1} \sigma(d_i, d_k) = 0 \\ \frac{\sum_{\substack{k=0 \\ k \neq i}}^{m-1} (\sigma(d_i,d_k) \cdot \lambda(d_k,t_j)^2)}{\sum_{\substack{k=0 \\ k \neq i}}^{m-1} \sigma(d_i,d_k)} & \text{otherwise.} \end{cases}$$

Descriptive power of a term t_j in the topic of a document d_i is a measure of the quality of t_j as a descriptor of documents similar to d_i. If no other document is similar to d_i or t_j does not occur in other documents similar to d_i then the descriptive power of t_j in the topic of d_i is equal to 0.

The last property we define is *document exhaustivity* with regard to a topic. A document is exhaustive (or comprehensive) with regard to the topic identified by a term if most terms that co-occur with the given term tend to discriminate that document; exhaustivity of a document can be thought of as the dual property of descriptive power of a term. We propose a measure of document exhaustivity as a function $\Xi : \{t_0, \ldots, t_{n-1}\} \times \{d_0, \ldots, d_{m-1}\} \to [0, 1]$:

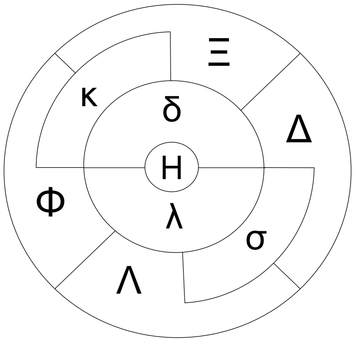
$$\Xi(t_i, d_j) = \begin{cases} 0 & \text{if } \sum_{\substack{k=0 \\ k \neq i}}^{n-1} \kappa(t_i, t_k) = 0 \\ \frac{\sum_{\substack{k=0 \\ k \neq i}}^{n-1} (\kappa(t_i, t_k) \cdot \delta(t_k, d_j)^2)}{\sum_{\substack{k=0 \\ k \neq i}}^{n-1} \kappa(t_i, t_k)} & \text{otherwise.} \end{cases}$$

By the definition of $\Xi(t_i, d_j)$, if term t_i does not co-occur with any other term or d_j does not contain any term that co-occurs with t_i then the exhaustivity of d_j with regard to the topic of t_i is 0.

In the hypergraphs of figure 4.2 terms 2, 3 and 4 are all good descriptors in the topic of documents D, E and F. However, while terms 3 and 4 are good discriminators in that topic, term 2 is not—term 2 occurs often in that topic but not only in that topic. Note also that in this example documents D, E and F are exhaustive on the topic of terms 2, 3 and 4. Among these three documents, only D and E are focused on the topic. For example, document F contains most terms that co-occur in that topic but not only terms from that topic. The diagram of figure 4.3 summarizes the notions discussed in this section. It starts with the hypergraph incidence matrix $\mathbf{H}$ in the center of the diagram, where $\mathbf{H}[i, j]$ represents the *number of occurrences of term t_j in document d_i*, and shows how the higher-level notions are built upon the more basic ones. Dual notions (e.g., similarity and co-occurrence) appear on opposite sides of the diagram.

A Summary of How EXTENDER Applies the Dual Notions

The higher-order notions of discriminating power, descriptive power, focus and exhaustivity are useful for identifying and characterizing topics. Topic descriptors and discriminators are useful as query terms to favor recall and precision respectively. We have applied discriminating power and focus in the implementation of a clustering algorithm to produce cohesive topics. Because descriptors describe the subject of a topic, they are good terms to use as the topic's label, when



$\mathbf{H}[i,j]$: number of occurrences of t_j in d_i
$\delta(t_i,d_j)$: discriminating power of t_i in d_j
$\lambda(d_i,t_j)$: descriptive power of t_j in d_i;
$\sigma(d_i,d_j)$: similarity of documents d_i and d_j.
$\kappa(t_i,t_j)$: co-occurrence of terms t_i and t_j.
$\Delta(t_i,d_j)$: discriminating power of t_i in d_j's topic.
$\Lambda(d_i,t_j)$: descriptive power of t_j in d_j's topic.
$\Phi(d_i,t_j)$: focus of d_i on t_j's topic.
$\Xi(t_i,d_j)$: exhaustivity of d_j on t_i's topic.

Figure 4.3: The different levels of the document-term duality

the topic is presented to the user. A combination of focus and exhaustivity can be used to rank documents in a topic. The next chapter discusses the the use of the notions developed in this chapter in the implementation of the EXTENDER system.

5

The EXTENDER System

Our pragmatic goal is to develop competent mechanisms to search the Web for topics that the user may find useful for inclusion in a knowledge model. The framework developed in the previous chapter has been applied to this task in the implementation of the EXTENDER system. This chapter takes a closer look at EXTENDER, discussing its goals, methods and algorithms in detail.

5.1 EXTENDER's Processing Cycle

Starting from a concept map, EXTENDER identifies and suggests sets of terms characterizing novel but related topics, as candidate new topics for inclusion in a knowledge model. As opposed to manually constructed topics, EXTENDER's topics are the result of automatic processes of querying a Web search engine, filtering, and clustering, therefore, we refer to them as *artificial topics*.

EXTENDER is a *human-in-the-loop* system: It automates part of the knowledge extension process, by searching for useful material, but relies on the user to carry out the knowledge-modeling task. Figure 5.1 outlines EXTENDER's processing cycle. The system starts from a concept map

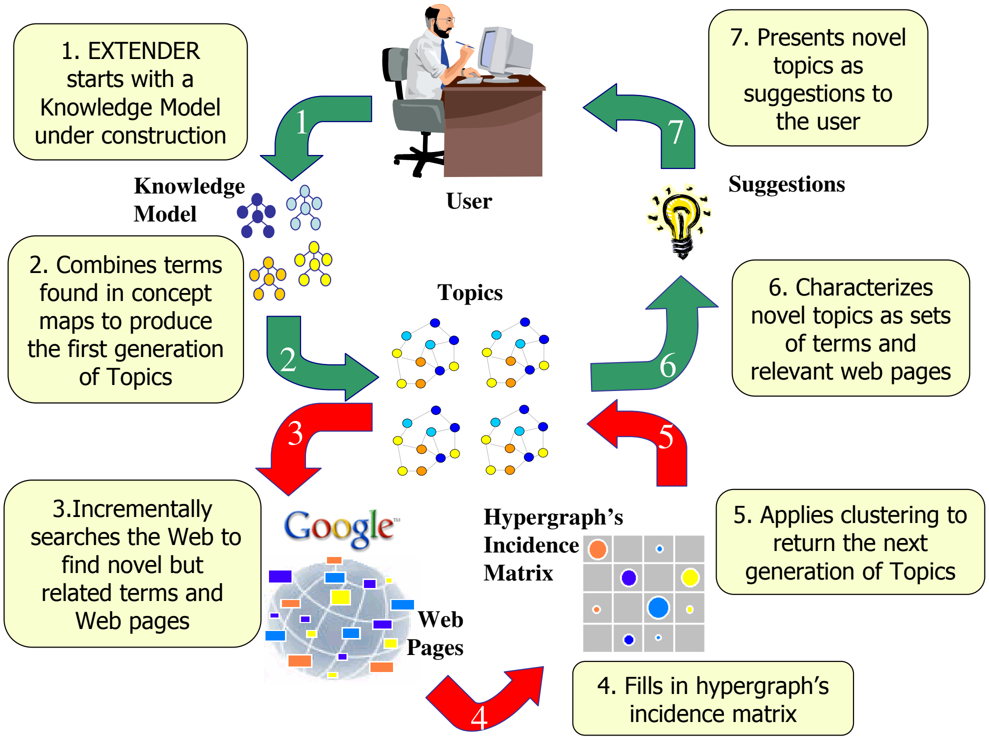

1. EXTENDER starts with a Knowledge Model under construction
User
7. Presents novel topics as suggestions to the user
1
7
Knowledge Model
Suggestions
2. Combines terms found in concept maps to produce the first generation of Topics
Topics
6. Characterizes novel topics as sets of terms and relevant web pages
2
6
3
5
3.Incrementally searches the Web to find novel but related terms and Web pages
Google
Hypergraph's Incidence Matrix
5. Applies clustering to return the next generation of Topics
Web Pages
4
4. Fills in hypergraph's incidence matrix

Figure 5.1: EXTENDER's Cycle.

and iteratively searches the Web for novel information, which is clustered to produce topics that are related to the initial concept map. The user can highlight a concept or set of concepts from the starting concept map in order to bias the system toward the search for topics related to the highlighted concepts. Alternatively, the search can be initiated from the full map, without introducing any additional bias.

At each iteration, the system's goal is to extend the current topics, an operation that requires searching the Web for related novel material. The collected material is represented by means of hypergraphs' adjacency matrices, clustering is applied to identify topics in the collection, and unimportant material is discarded. This process is repeated a number of times, with the stopping criterion depending on a user-selected limit on iterations. Once EXTENDER completes its iterations, it presents the generated topics as suggestions to the user. In addition, it organizes the Web pages that gave rise to those topics according to topic, to facilitate access to topic-relevant information. A generated topic can be easily imported as a set of concepts, from which the user can start the mapping process.

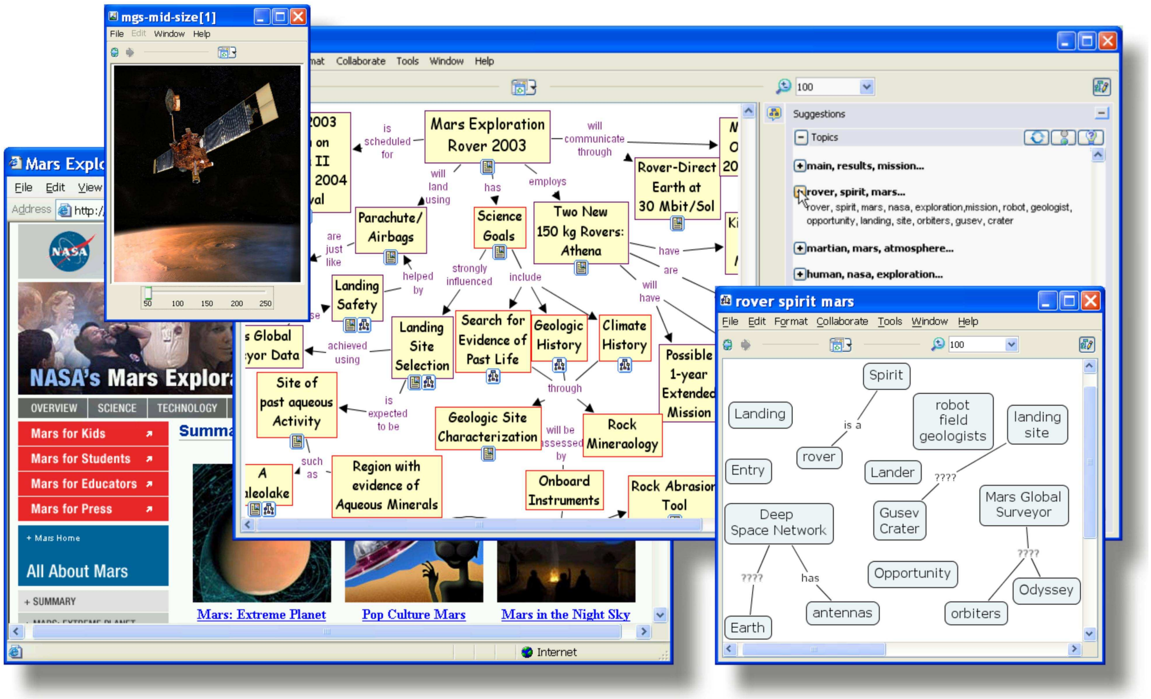


Figure 5.2: Portion of a Knowledge Model with EXTENDER suggesting new topics.

Figure 5.2 shows a part of a knowledge model with EXTENDER's suggestion window for new topics at the upper right. The in-progress concept map in the bottom right contains some concepts that the user selected from a topic suggested by EXTENDER.

5.2 Goals for EXTENDER's Topic-Generation Strategy

EXTENDER's task is an instance of a more general one: suggesting novel topics related to a user's focus. For example, topic suggestions could be useful to a researcher (e.g., to provide related but distinct areas to consider for connections and synergies or to help assure that relevant areas have been considered).

The effectiveness of a topic-generation strategy is hard to assess because the usefulness of topic suggestion is highly subjective. However, to increase the likelihood that the proposed topics are useful to the user task, it is desirable for the topics to satisfy a number of properties:

- **Local quality**. Each generated topic must be of high quality according to the criteria for the domain. Such criteria might include measures for conciseness (that the topic is summarized in a few terms, for easy user comprehension), term coherence (that each topic description is constituted of tightly related terms and documents), etc.
- **Global Coherence**. The system must be able to maintain its focus within relevant topics. To achieve global coherence, the generated topics must be related to the originating knowledge model.
- **Coverage**. A good topic-generation strategy should be able to generate a sufficient subset of the topics considered to be relevant.
- **Novelty**. Some generated topics must go beyond previously captured information.
- **Diversity**. The system should generate a rich set of topics. These topics must be sufficiently diverse from each other for additional topics to be useful.

Summary of How the Goals Interact

EXTENDER's strategy for preserving global coherence is to use a *search context* for filtering irrelevant information and to identify good topic descriptors and discriminators for guiding query formation and subsequent retrievals. To attain coverage, novelty and diversity EXTENDER generates queries at incremental distances from the set of terms that originated the request. The system uses a *curiosity mechanism* to diversify during initial stages and focus towards the end. Finally, to produce cohesive topics EXTENDER applies a clustering algorithm that relies on the dual notions of description, discrimination, exhaustivity and focus presented in chapter 4. The next sections discuss these methods and algorithms in detail.

5.3 Searching for Novel but Related Material

EXTENDER's artificial topics are produced by combining terms and documents from Web searches. The terms and documents collected by the system should be relevant to the knowledge model under construction but should help to extend the knowledge beyond the information that is already captured. For that reason, attaining novelty and diversity may be as important, or even more important, than attaining similarity. Therefore, methods are needed to produce topics with the right balance of novelty and relevance.

Search Context

Search engines restrict queries to a small number of terms (e.g., the 10-term limit for Google). As a result, a single query cannot reflect extensive contextual information. For knowledge model extension, the knowledge model under construction provides a rich context that can be exploited to preserve global coherence. In order to reflect full context, incremental approaches are needed. In an incremental approach to topic search, contextual information plays a fundamental role in guiding the exploration and discovery of related material. During its cycle, EXTENDER maintains the relationship between candidate topic terms and the initial concept map in three ways:

- **Term-weight reinforcement.** Terms collected during EXTENDER's retrievals are associated with weights summarizing the terms' descriptive and discriminating power. During the first cycle, a term's descriptive power is obtained directly from the topology of the source concept map—possibly adjusted by some bias introduced by the user's selection of certain concepts from the map. For subsequent iterations, contextual information is used for term-weight reinforcement, favoring the weights of terms that have proven to be good descriptors or discriminators for the topic represented by the search context.

- **Information filtering.** For a document's terms to be considered candidates for inclusion as part of a new topic, the document has to survive a selection process that requires a minimum similarity between the document and the search context. Novel terms that are not good descriptors or discriminators of the topic reflected by the search context are also discarded.

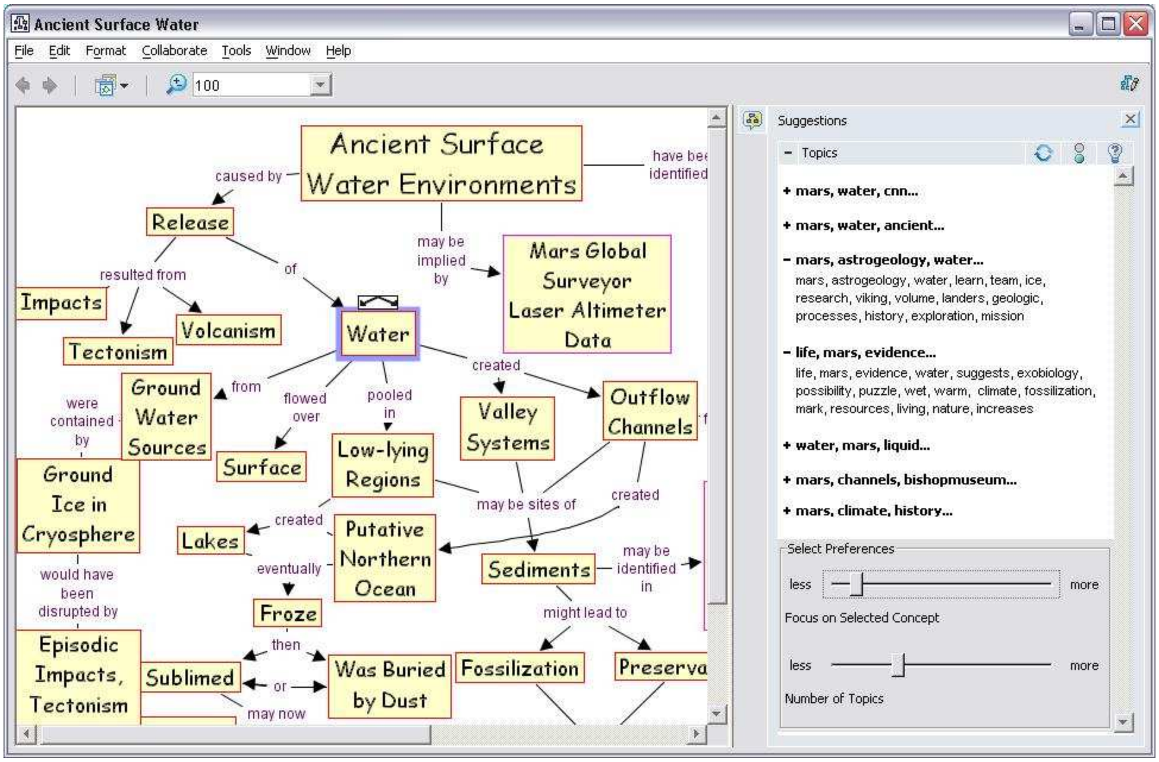
- **Query refinement.** The first query terms generated for a Web search may not provide the definitive results. However, initial search results can help to automatically refine subsequent queries. Terms that occur often in documents similar to the search context help to achieve good recall when used in a query. On the other hand, terms that tend to occur only in similar documents are useful for achieving high precision. Consequently, the generation of second-round and subsequent queries can significantly benefit from contrasting previous search results against the search context.

EXTENDER's search context is initially defined using the knowledge model under construction, and it is then progressively updated as the focus shifts though a connected series of topics. Figures 5.3 and 5.4 illustrate the importance of exploiting the search context to keep global coherence. The first figure presents a concept map from a knowledge model on *Mars*, describing the topic *Ancient Surface Water Environments*. The second figure, on the other hand, presents a concept map on the topic of *Rivers*. In both examples the user highlighted the concept *Water* to initiate the search. However, the topics produces for each map are different, reflecting the context of the corresponding maps. The two sliders at the bottom right of EXTENDER's suggestion window allow the users to control the focus on the selected concept and the maximum number of topics the system will return. The first slider has an effect on the weightings given by the system to the highlighted concepts. The second slider directly affects how many times the system will iterate before returning the final set of topics and the number of topics produced after each iteration (ramification factor).

Curiosity Mechanism

EXTENDER uses a "curiosity mechanism" to diversify during initial processing stages and to focus towards the end. The application of EXTENDER's curiosity mechanism is in the spirit of searching and learning techniques (e.g., simulated annealing and reinforcement learning) in which a temperature factor is used to favor exploration at the beginning and exploitation during the final stages.

Throughout the system's iterations, while attempting to extend a given topic $\mathbf{T}$, new-found terms are collected. Because the number of collected terms grows rapidly, novel terms are only preserved if they survive a selection process regulated by the curiosity mechanism. For each term t, the system tracks both the goodness of t in describing the topic $\mathbf{T}$ and the goodness of t in discriminating $\mathbf{T}$. To do so, it considers $\mathbf{T}$ as a multiset of terms and computes functions $\Lambda(\mathbf{T}, t)$



+ mars, water, cnn...

+ mars, water, ancient...

− mars, astrogeology, water...
mars, astrogeology, water, learn, team, ice, research, viking, volume, landers, geologic, processes, history, exploration, mission

− life, mars, evidence...
life, mars, evidence, water, suggests, exobiology, possibility, puzzle, wet, warm, climate, fossilization, mark, resources, living, nature, increases

+ water, mars, liquid...

+ mars, channels, bishopmuseum...

+ mars, climate, history...

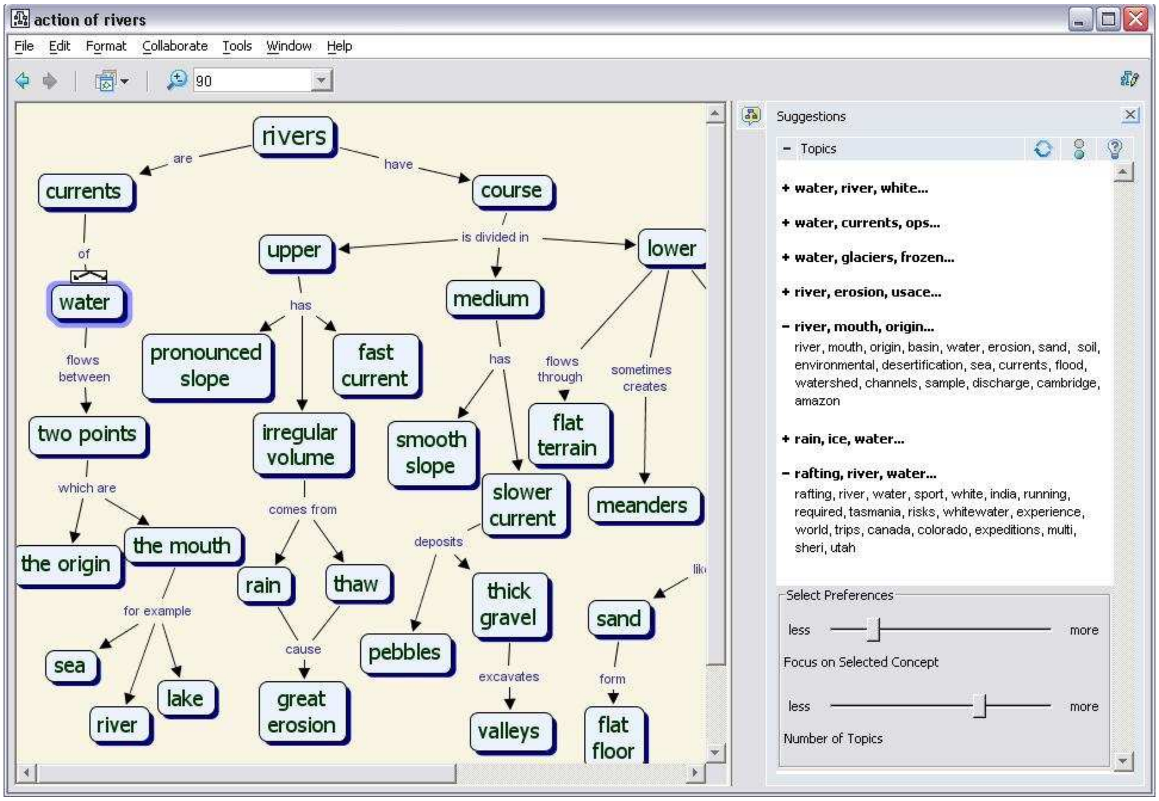
Figure 5.3: EXTENDER suggesting topics for the concept *Water* in the context of *Mars' Ancient Surface Water Environments.*

and $\Delta(t, \mathbf{T})$, respectively.

The curiosity mechanism imposes a threshold for the survival of descriptors and discriminators. For iteration I, the threshold for the survival of descriptors is computed by means of a function $\mathcal{T}_\Lambda : \{0, \ldots, s-1\} \to [a, b]$

$$\mathcal{T}_\Lambda(\text{I}) = (b-a) \cdot \left(\frac{\text{I}}{s-1}\right)^c + a,$$

where a stands for the "starting threshold" parameter, b for the "stopping threshold" parameter, c is a curiosity decay parameter, and s is the total number of iterations. The parameter a (resp. b) reflects the initial (final) stage of exploration (exploitation), when many (few) new terms are collected. The threshold for discriminators, $\mathcal{T}_\Delta$, is defined similarly.



+ water, river, white...

+ water, currents, ops...

+ water, glaciers, frozen...

+ river, erosion, usace...

− river, mouth, origin...
river, mouth, origin, basin, water, erosion, sand, soil, environmental, desertification, sea, currents, flood, watershed, channels, sample, discharge, cambridge, amazon

+ rain, ice, water...

− rafting, river, water...
rafting, river, water, sport, white, india, running, required, tasmania, risks, whitewater, experience, world, trips, canada, colorado, expeditions, multi, sheri, utah

Figure 5.4: EXTENDER suggesting topics for the concept *Water* in the context of *Rivers*.

Another curiosity threshold is used by EXTENDER to filter irrelevant documents according to the search context. This is implemented by a similarity threshold function τ_σ defined analogously to the definition of the other curiosity mechanism functions.

Because the curiosity threshold increases with the number of iterations, novel terms and documents are seldom collected during the final stages. As a consequence, the exploitation phase primarily reinforces the weights associated with particular material that has already been added to the collection.

5.4 Generating Cohesive Topics

Clustering is the unsupervised classification of items into groups (clusters). Basically, we want to form these groups in such a way that items in the same group are similar to each other, whereas items in different groups are dissimilar. Grouping similar items together while keeping dissimilar ones appart is usually an expensive task but necessary for attaining **local coherence**. In the following, we present an overview of the major clustering methods and after that, we address the problem of generating cohesive topics by proposing a clustering algorithm tailored for EXTENDER.

Clustering Algorithms

There are many dimensions that can be selected to classify clustering algorithms [Jain et al., 1999, Berkhin, 2002]. Traditional approaches to clustering can be broadly classified into *hierarchical* and *partitioning*.

Hierarchical Clustering

Hierarchical clustering algorithms (e.g., [Sibson, 1973, Defays, 1977]) build a tree of clusters, also known as dendrogram, reflecting the nested groupings of data at different levels of granularity. Hierarchical clustering methods are usually classified into *agglomerative* and *divisive*. In order to produce a nested series of partitions, an agglomerative approach starts by assigning each document to a singleton group and progressively merges groups according to some measure of similarity, until a stopping criterion is satisfied. A divisive method, in contrast to an agglomerative method, begins with a single cluster containing all of the data, and proceeds by splitting the single cluster up into smaller sized clusters. Agglomerative clustering builds the tree of clusters from the bottom to the top, while divisive clustering operates from the top to the bottom, hence these two hierarchical approaches to clustering are also known as *bottom-up* and *top-down* respectively.

Hierarchical clustering facilitates the exploration of data at different levels of granularity and is robust to variations of cluster size and shape. The most commonly cited disadvantage of hierarchical approaches is their computational cost. Hierarchal clustering takes quadratic time on the number of documents and therefore is too costly to be performed on large collections. Another problem is the large IO cost and space needed to build a tree of clusters.

Partitioning Clustering

A Partitioning approach to clustering, in contrast to a hierarchical approach, obtains a single partition of the collection. Partitioning clustering algorithms are typically more time and space effective than hierarchical approaches but require the user to stipulate the number k of desired clusters [Dubes, 1987].

Searching for the optimal partition by checking all possible partitions is too expensive from a computational perspective. Therefore, a number of greedy heuristics have been developed to produce an approximation of the optimal partition. An iterative optimization approach to partitioning clustering starts from k clusters and iteratively reassigns points between these clusters until no point is reassigned to a different cluster. To guide the point relocation process, a common approach is to define an objective function based on intra-cluster similarity and inter-cluster dissimilarity [Zhao and Karypis, 2001]. The pair-wise computation of similarities between all items in a collection is too expensive. To lessen this cost, a common approach is to take a centroid or a small set of points representing each cluster and to compute the objective function using the clusters' representatives instead of all the clusters' elements. The *k-means* algorithm [Hartigan and Wong, 1979] is a popular centroid-based partitioning algorithm. Because centroids are typically computed as the weighted average of points within a cluster, they have a clear geometric interpretation but tend to be expensive to calculate because they have to be recomputed for each newly assembled group.

Other partitioning algorithms, such as *expectation maximization* (EM) [Dempster et al., 1977], identify each cluster with a certain probabilistic model whose unknown distribution parameters (e.g., mean and variance) have to be found. Each point x_i in a collection is assumed to belong to one cluster C_j and the probability $\Pr(x_i|C_j)$ of such assignment is estimated on the basis of the guessed parameters. The initial guess is iteratively refined to maximize an objective function. The maximization of the objective function guarantees the maximum likelihood estimate of the missing parameters.

Hard vs. Soft Clustering

Traditional clustering approaches produce partitions: Each item belongs to exactly one cluster. These methods are sometimes said to produce a hard clustering, because they result in an inflexible assignment of items to clusters. Soft clustering [Ruspini, 1969] relaxes this requirement by associating each item with every cluster using a membership function. Hence, the same item may be part of more than one cluster, where the item membership coefficient for each cluster can be specified by means of a fuzzy value in $[0, 1]$.

Soft clustering can be integrated both with hierarchical and partitioning methods. The design of a fuzzy membership function and techniques for efficiently updating this function as the clusters are recomputed are typically the most important problems associated with soft clustering approaches.

Clustering algorithms have been used in a large variety of applications, including data-mining, data compression, image segmentation, object recognition, and information retrieval. Depending on the application, several design choices for the implementation of clustering algorithms can be made. Due to the application dependant nature of clustering, the design choices are not always guided by the same considerations. In the next section we present EXTENDER's clustering problem, followed by a discussion of the clustering algorithm we propose to address the specific problem of topic identification in the context of a knowledge extension task.

EXTENDER's Clustering Problem

EXTENDER's artificial topics are the product of searching the Web for material similar to the user's context, filtering irrelevant material, and clustering the remaining collection of search results. The problem of clustering a collection of short text excerpts from highly related documents to identify cohesive topics makes this task different from other clustering scenarios. EXTENDER's clustering problem is characterized by:

- **The topic generation task**. In traditional views, clustering algorithms have been suggested in the context of index construction, for reasons of efficiency. They have been developed in response to the clustering hypothesis, which states that closely associated documents tend to be relevant to the same requests [Rijsbergen, 1979]. However, EXTENDER's clustering problem is not aimed at indexing documents for efficient retrieval but at dynamically generating sample topics that will serve as hints to the knowledge modeler.
- **Short descriptions of documents**. Each document is represented only by the information that is readily available from the search results (e.g., title, "snippet" of text, url, Open Directory Project summary). Unlike most document clustering problems, in which documents are represented by their complete text, our clustering technique must rely on methods that use

limited information to identify the topic of the documents.

- **Highly related material**. Because EXTENDER attempts to preserve global coherence, most documents in the collection are highly related, i.e., they share a common general theme. The identification of more specific topics within a collection of documents with a common theme requires the identification of items (terms and documents) that are good at discriminating topics at a fine level of granularity.
- **Small topic-specific lexicon**. As EXTENDER iterates, only a selection of terms is preserved—those terms surviving a filtering process regulated by the curiosity mechanism discussed in section 5.3. Consequently, only terms that play a reasonably important role as descriptors or discriminators of the topic at hand are part of the dynamically generated lexicon. This is in contrast to most clustering situations, where the number of terms involved is usually very large and may correspond to very diverse topics.
- **Overlapping topics**. Documents collected by EXTENDER may belong to more than one thematic category. Instead of producing a partition of the document collection, EXTENDER's clustering mechanism must combine similar material together, with the resulting groupings representing topics with overlapping content and fuzzy boundaries. This calls for the application of soft clustering techniques.

Clustering Around Medoids

A common approach in clustering is to use one or a small set of points as cluster representatives. For example, the k-means algorithm uses a *centroid*, which is the weighted average of points within a cluster. An alternative approach is to use a *medoid* instead of a centroid. A medoid is the most appropriate point within a cluster that represents it. Assuming the set of medoids is given, then the clustering problem reduces to selecting the subsets of items "close" to the respective medoids. In particular, for a soft clustering approach each cluster C_i can be represented by a membership function. Once medoids are selected, the grouping of points for forming each cluster can be easily done using this membership function (e.g., by using a threshold on the number of items in a cluster or a threshold on the minimum similarity allowed). While this grouping phase is simple, selecting a set of good medoids is a more complex task.

An early clustering approach proposing a technique for medoid identification is Partitioning Around Medoids (PAM) [Kaufman and Rousseeuw, 1989]. The PAM algorithm starts from an arbitrary initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves an objective function based on intra-cluster similarity. A problem accompanying this technique is that it requires pre-specifying number of output clusters, which has to be presented as an input to the algorithm. A second problem is the need to re-compute intra-cluster similarity each time points are re-nominated as potential medoids, which is obviously very costly. In addition, the initial selection of candidate medoids is arbitrary, and the algorithm does not apply an efficient heuristic to expedite the search for the best set of medoids. In EXTENDER's clustering problem the number of output clusters is not known in advance and efficiency is an important factor. Thus, a different technique is needed to search for cluster representatives.

In section 4.2 we proposed a framework for analyzing terms, documents and topics in the light of a series of dual notions. Among the studied notions were the notions of term descriptive power, term discriminating power, document exhaustivity and document focus. Terms with high descriptive and discriminating power are good representatives of the topics in which they are included because they tend to occur often in the context of that topic, but not in other topics. Likewise, document that are both exhaustive and focused are also good topic representatives because they provide thorough information specific to the topic rather than general unfocused data. Consequently, terms with high descriptive and discriminating power, as well as documents with high focus and exhaustivity coefficient could be used as topic medoids in a clustering around medoids approach.

A Co-Clustering Algorithm Tailored for EXTENDER

An important decision in the design of clustering algorithms for topic identification is whether the grouping is applied to documents or terms. Terms may be clustered on the basis of the documents in which they co-occur. Term clustering has typically been applied in automatic construction of thesauri (e.g., [Crouch, 1988]) and it has also proved to be useful in reducing feature dimensionality for more effective document classification (e.g., [Baker and McCallum, 1998]). However, most of the traditional clustering approaches cluster documents rather than terms, using their similarity as the basis for grouping them.

When full access to documents' text is available for topic generation, document clustering is generally preferred over term clustering. This is because in most real data collections documents are better topic representatives than terms, giving the clustering algorithm greater discerning power to identify topics. However, when documents are represented by a small number of terms (as is the case for the text excerpts collected by EXTENDER), and the collection under analysis consists of material that shares a common general theme (which is a consequence of EXTENDER's attempt to preserve global coherence), then an unusual clustering situation arises. In this new clustering scenario terms may be as informative as documents for identifying topics within the collection.

With a few exceptions (e.g., [Dhillon, 2001]) most existing clustering algorithms apply single purpose clustering—they cluster documents and terms separately. In this section, we propose a new clustering method that identifies topic representatives to cluster documents and terms simultaneously. In order to identify the best topic representatives, we need a mechanism to quantify the "representation value" of a term or a document in a topic. Given a term i and a document j, we measure the representation value of term i in the topic of document j by means of a function $\rho^{[\Lambda\Delta]}(i,j)$ defined in terms of the descriptive and discriminating power functions:

$$\rho^{[\Lambda\Delta]}(i,j) = \Lambda(j,i) \times \Delta(i,j).$$

Similarly, we define the representation value of document j in the topic of term i by means of function $\rho^{[\Xi\Phi]}(j,i)$ defined in terms of the exhaustivity and focus functions:

$$\rho^{[\Xi\Phi]}(j,i) = \Xi(i,j) \times \Phi(j,i).$$

Using these functions, we developed an algorithm to co-cluster documents and terms.

Our clustering algorithm takes as input two matrices codifying functions δ and λ (defined in section 4.2) for a collection of terms and documents. The algorithm computes the similarity, co-occurrence, discrimination, description, focus and exhaustivity matrices using the techniques described in section 4.2. After these matrices are computed, the co-clustering procedure is invoked.

In order to co-cluster terms and documents, the algorithm starts by assuming that every document in the collection is a good topic representative, i.e., it assumes that all documents are candidate medoids. The computation continues with a loop that, once terminated, returns a small set of terms and documents that play the role of medoids, representing different topics in the collection. This is done by alternating two processes:

- FIND-MEDOID-TERMS: search for terms with the highest $\rho^{[\Lambda\Delta]}$ value in the topics associated with the candidate document-medoids, and

- FIND-MEDOID-DOCUMENTS: search for the documents with the highest $\rho^{[\Xi\Phi]}$ value in the topics associated with the candidate term-medoids.

These two recurrent processes are repeated until any of the termination conditions (to be discussed in section 5.4) is satisfied.

Finally, each term-medoid and document-medoid is applied in the definition of a membership function for other terms and other documents in the collection. Suppose m_i^t is the term-medoid representing cluster C_i, then μ_i^D, the document membership function for cluster C_i is defined as:

$$\mu_i^D(d_j) = \rho^{[\Xi\Phi]}(d_j, m_i^t).$$

The term membership function $\mu_i^T(t_j)$ for cluster C_i is defined as follows:

$$\mu_i^T(t_j) = \begin{cases} 0 & \text{if } t_j \text{ occurs only once in the collection} \\ \rho^{[\Lambda\Delta]}(t_j, m_i^d) & \text{otherwise} \end{cases}$$

where m_i^d is the document-medoid representing cluster C_i. The general algorithm and the procedures used for generating cohesive topics are outlined in tables 5.1, 5.2 and 5.3.

A Note on Convergence and Time Complexity

To find the best topic representatives, our algorithm alternates the FIND-MEDOID-TERMS and FIND-MEDOID-DOCUMENTS procedures until (1) two consecutive iterations produce the same set of medoids, or (2) the same result is detected for two non-consecutive iterations.

```
PROCEDURE GENERATE-COHESIVE-TOPICS
INPUT:
  L: matrix codifying term descriptive power in a document % L[i,j]= $\lambda(d_i, t_j)$
  D: matrix codifying term discriminating power in a document % D[i,j]= $\delta(t_i, d_j)$
OUTPUT:
  DC: a matrix such that DC[i,j] contains the membership value of document j in cluster i
  TC: a matrix such that TC[i,j] contains the membership value of term j in cluster i
BEGIN
  Similarity = COMPUTE-SIMILARITY(L)
  Co-occurrence = COMPUTE-CO-OCURRENCE(D)
  Discrimination = COMPUTE-DISCRIMINATION(D,Similarity)
  Description = COMPUTE-DESCRIPTION(L,Similarity)
  Focus = COMPUTE-FOCUS(L,Coocurrence)
  Exhaustivity COMPUTE-EXHAUSTIVITY(D,Coocurrence)
  discriminatingTerms = CO-CLUSTERING(Description, Discrimination, Exhaustivity, Focus)
  i = 0
  for each term j such that discriminatingTerms[j] $\neq$ 0
  do % define the membership values for a new topic
    i = i + 1
    k = discriminatingTerms[j] % select document-medoid for topic j
    for each document l
    do
      DC[i,l] = Focus[l,j] * Exhaustivity[j,l] % membership value of document l in topic i
    for each term l
    do
      if term l occurs only once
      then
        TC[i,l]=0
      else
        TC[i,l] = Discrimination[l,k] * Description[k,l] % membership value of term l in topic i
END
```

Table 5.1: Pseudocode of the algorithm for generating cohesive topics.

For any collection of terms and documents, the procedures FIND-MEDOID-TERMS and FIND-MEDOID-DOCUMENTS return a set of terms and a set of documents containing the candidate medoids selected after each iteration. It is easy to verify that after each iteration the sizes of the sets containing medoid-term and medoid-document decrease or remain the same, so each cluster will converge to a unique medoid (case 1) or it will fluctuate among a finite number of candidate medoids (case 2). Case 1 implies that the algorithm has found single representatives for each identified topics. On the other hand, case 2 occurs when some of the identified topics

```
PROCEDURE Co-Clustering
INPUT:
  Description: a matrix codifying descriptive power in a topic
  Discrimination: a matrix codifying discriminating power in a topic
  Exhaustivity: a matrix codifying exhaustivity
  Focus: a matrix codifying focus
OUTPUT:
  medoidTerms: a vector such that medoidTerms[i] = j ($j \neq 0$) if i is a medoid of j's topic
BEGIN
  for each document i
  do
      medoidDocuments[i] = 1 % assume all documents are medoids of an arbitrary term
  Update-States(States,focusedDocuments) % we keep track of the system state
  while not done
  do
      medoidTerms = Find-Medoid-Terms(medoidDocuments,Description,Discrimination)
      medoidDocuments = Find-Medoid-Documents(medoidTerms,Exhausitivy, Focus)
      Update-States(States, medoidDocuments)
      done = Check-Termination(States) % check for convergence or for repetitive sequences
  return medoidTerms
END
```

Table 5.2: Co-Clustering procedure.

have multiple representatives. The second case is uncommon in our experience, taking place in situations when the algorithm's selection of a topic representative fluctuates between two or more terms (documents).

Once the loop terminates, only one term-medoid (document-medoid) is selected as a representative of each topic. This selection is straightforward for case 1—the topic medoid is the term (document) to which the cluster converges. In case 2, those clusters for which the algorithm diverges are represented by a term (document) arbitrarily selected from the terms (documents) involved in the repetitive sequence.

The time complexity for the procedures Find-Medoid-Terms and Find-Medoid-Documents is $O(m \times n)$, where m is the number of documents and n is the number of terms in the collection. These procedures are invoked at most k times, where k is the minimum value of m and n. Because the matrices Description, Discrimination, Exhaustivity and Focus

```
PROCEDURE FIND-MEDOID-DOCUMENTS
INPUT:
 medoidTerms: a vector codifying potential term-medoids
 Exhaustivity: a matrix codifying exhaustivity
 Focus: a matrix codifying focus
OUTPUT:
 medoidDocuments: a vector codifying potential document-medoids
BEGIN
 for each document i
 do
   medoidDocuments[i] = 0
 for each term j such that medoidTerms[j] ≠ 0
 do
   mostExhaustiveAndFocusedDocumentForJ = 0
   mostExhaustiveAndFocusedValueForJ = 0.0
   for each document i
   do
     v = Exhaustivity[j,i]*Focus[i,j]
     if v > mostExhaustiveAndFocusedValueForJ
     then
       mostExhaustiveAndFocusedDocumentForJ = i
       mostExhaustiveAndFocusedValueForJ = v
   medoidDocuments[mostExhaustiveAndFocusedDocumentForJ] = j
END
PROCEDURE FIND-MEDOID-TERMS
INPUT:
 medoidDocuments: a vector codifying potential document-mendoids
 Description: a matrix codifying descriptive power
 Discrimination: a matrix codifying discriminating power
OUTPUT:
 medoidTerms: a vector codifying potential term-medoids
BEGIN
 for each term i
 do
   medoidTerms[i] = 0
 for each document j such that medoidDocuments[j] ≠ 0
 do
   mostDescriptiveAndDiscriminatingTermForJ = 0
   mostDescriptiveAndDiscriminatingValueForJ = 0.0
   for each term i
   do
     v = Description[j,i] * Discrimination[i,j]
     if v > mostDescriptiveAndDiscrimatingValueForJ
     then
       mostDescriptiveAndDiscriminatingTermForJ = i
       mostDescriptiveAndDiscriminatingValueForJ = v
   medoidTerms[mostDescriptiveAndDiscriminatingTermForJ] = j
END
```

Table 5.3: Procedures for finding document- and term-medoids.

are fixed, the termination condition is guaranteed to take place in no more than k iterations. In practice, however, we noticed that the FIND-MEDOID-TERMS and FIND-MEDOID-DOCUMENTS procedures are usually not invoked more than 3 or 4 times, which indicates that for real data, the algorithm finds the best medoids after very few iterations. Another costly component of the algorithm is the computation of the Similarity ($O(m^2)$), Co-occurrence ($O(n^2)$), Discrimination ($O(m^2 \times n)$), Description ($O(m^2 \times n)$), Focus ($O(m \times n^2)$) and Exhaustivity ($O(m \times n^2)$) matrices.

Despite the polynomial complexity of these procedures, the final time-cost for the proposed clustering algorithm is not high in practice. This is because EXTENDER's clustering problem does not involve large data sets. In contrast to most clustering situations, EXTENDER's clustering problem only involves a small number of terms and text excerpts, which originate from the data readily available from Google's search results. In addition, the costly IO associated with most clustering tasks is not an issue in our case because all the material can be represented in main memory. It is worth noticing that while the computational cost for the proposed algorithm is higher than the cost for some existing clustering algorithms, the evaluations reported in section 6.3 provide good evidence supporting that our algorithm is more appropriate than other less expensive clustering mechanisms for dealing with EXTENDER's topic identification problem.

An Illustrative Example

In this section we illustrate the operation of EXTENDER's clustering algorithm for a data set consisting of 12 text excerpts, all containing the term *mars* but with themes varying across diverse, more specific topics. We will show how the algorithm successfully identifies four cohesive topics from the set of documents. The following document excerpts are used as the input data set:

D1:	mars, exploration, nasa, science, missions, educational.
D2:	mars, exploration, rover, landing, nasa, lander.
D3:	mars, lander, water, missions, science, nasa.
D4:	mars, nasa, science, launch, missions, landing.
D5:	mars, astrology, stars, passion, ambition, energy.
D6:	mars, red, horoscope, astrology, zodiac, stars.
D7:	mars, horoscope, astrology, zodiac, passion, aries.
D8:	mars, ares, mythology, god, olympians, greek.
D9:	mars, ares, war, god, roman, greek.
D10:	mars, fiction, book, reviews, genre, movies.
D11:	mars, fiction, book, robinson, trilogy, novel.
D12:	mars, stars, life, science, fiction, book.

Figures 5.5 and 5.6 illustrate the operation of the algorithm. During the first iteration, the algorithm assumes that all documents are possible medoids. For the topic of each document, the algorithm identifies a term-medoid. In our example, the terms *nasa, astrology, horoscope, greek* and *fiction* are selected as candidate term-medoids. Figure 5.5 shows how term-medoids are associated with document-medoids, together with the terms' $\rho^{[\Lambda\Delta]}$ values. We can see that, for example, the term *nasa* has a $\rho^{[\Lambda\Delta]}$ value of 0.23 with regard to the topic of document D1. In the second part of iteration 1 the algorithm searches for a new set of potential document-medoids, selecting documents D2, D7, D9 and D10. In iteration 2 the algorithm identifies *nasa, horoscope, greek* and *fiction* as the best term-medoids for the topics of documents D2, D7, D9 and D10 respectively (figure 5.6). Reciprocally, these four documents are found to be the best document-medoids for the topics represented by *nasa, horoscope, greek* and *fiction*. Since two consecutive iterations produce

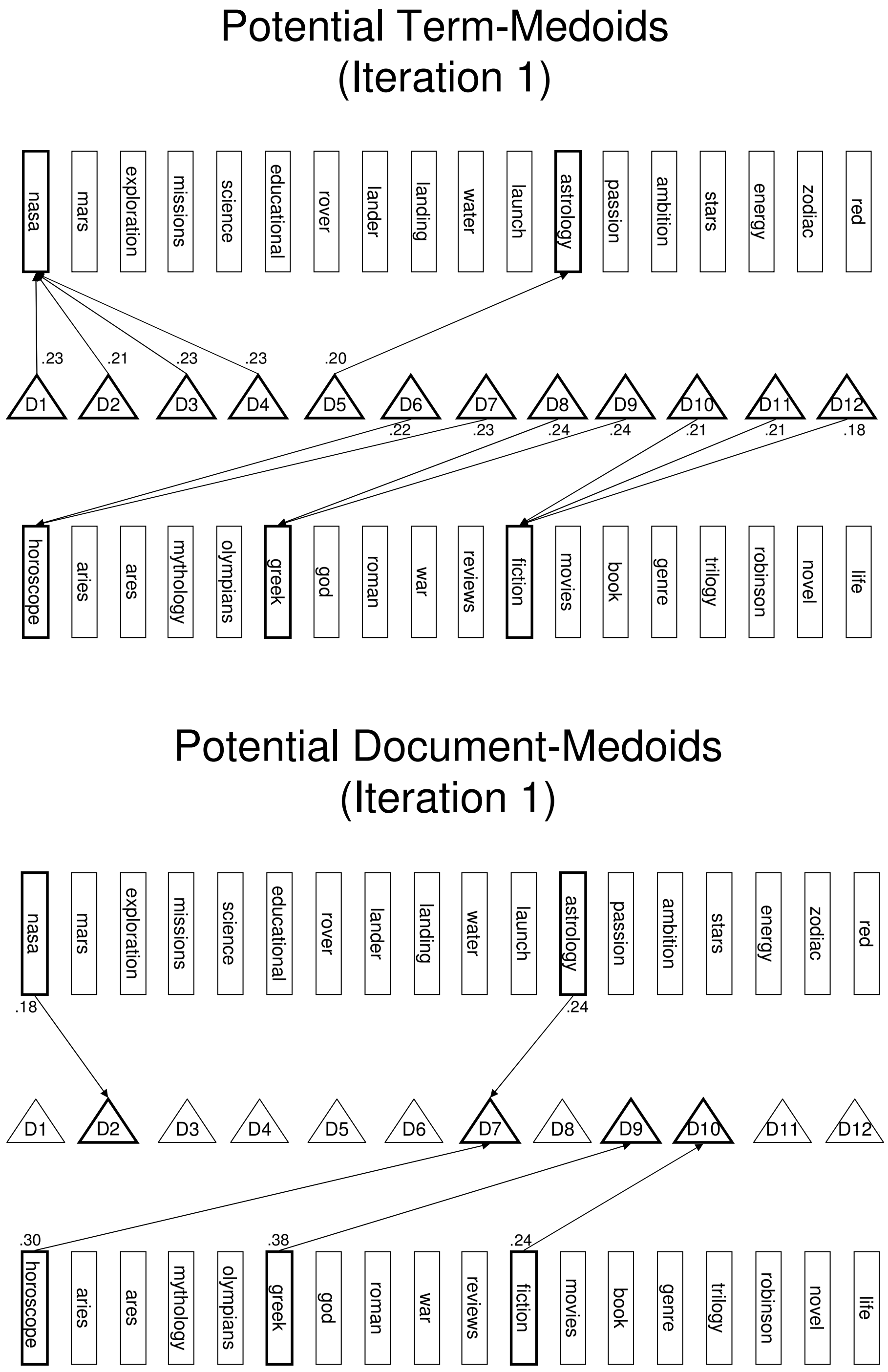
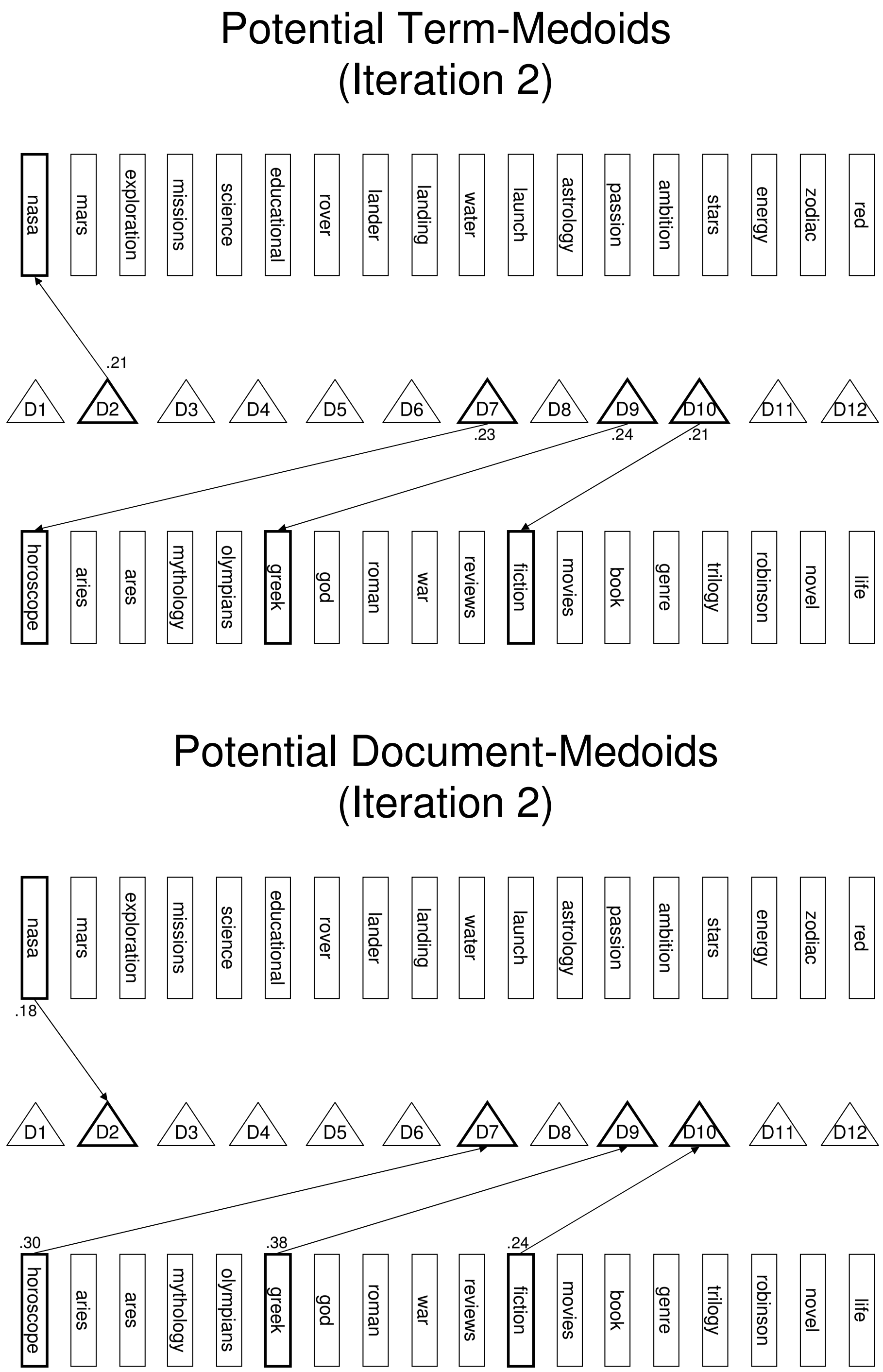


Figure 5.5: Searching for term-medoids and document-medoids during the first co-clustering iteration.



Potential Term-Medoids (Iteration 2)

Potential Document-Medoids (Iteration 2)

Figure 5.6: Searching for term-medoids and document-medoids during the second co-clustering iteration.

the same set of medoids, the main loop is terminated. Finally, the algorithm uses the document-medoids and the term-medoids to obtain the membership coefficients for the other terms and documents in the collection. Only terms that occur more than once are used to characterize topics. Table 5.4 presents the membership coefficients of the terms in each of the four identified topics, highlighting the terms ranked by the system as most representative of each cluster's topic (up to 0.05). Similarly, Table 5.5 presents the membership coefficients of the 12 documents in the four topics, highlighting the documents with highest representative value in each cluster (up to 0.01). This example demonstrates that the co-clustering algorithm returns intuitively correct results for a simple case.

Topic 1		**Topic 2**		**Topic 3**		**Topic 4**	
nasa	0.21	**horoscope**	0.23	**greek**	0.24	**fiction**	0.21
lander	0.19	**zodiac**	0.23	**ares**	0.24	**book**	0.21
landing	0.19	**astrology**	0.23	**god**	0.24	**mars**	0.12
exploration	0.19	**passion**	0.20	**mars**	0.11	**stars**	0.06
mars	0.13	**mars**	0.12	science	0.03	**science**	0.05
missions	0.13	**stars**	0.11	nasa	0.03	nasa	0.03
science	0.11	science	0.03	book	0.03	astrology	0.03
book	0.02	nasa	0.03	fiction	0.03	missions	0.03
fiction	0.02	book	0.03	astrology	0.03	god	0.02
astrology	0.02	fiction	0.03	stars	0.03	ares	0.02
stars	0.02	missions	0.03	missions	0.03	greek	0.02
god	0.02	god	0.02	zodiac	0.02	zodiac	0.02
ares	0.02	ares	0.02	horoscope	0.02	horoscope	0.02
greek	0.02	greek	0.02	passion	0.02	passion	0.02
zodiac	0.02	lander	0.02	lander	0.02	lander	0.02
horoscope	0.02	landing	0.02	landing	0.02	landing	0.02
passion	0.02	exploration	0.02	exploration	0.02	exploration	0.02

Table 5.4: Terms' membership coefficients in the four identified topics.

Topic 1		Topic 2		Topic 3		Topic 4	
D2	0.178	**D7**	0.301	**D9**	0.380	**D10**	0.240
D4	0.163	**D6**	0.280	**D8**	0.380	**D11**	0.240
D3	0.163	**D5**	0.044	D12	3.71E-04	**D12**	0.126
D1	0.163	D12	0.004	D11	3.71E-04	D6	0.003
D12	0.007	D11	4.17E-04	D10	3.71E-04	D5	0.003
D11	0.001	D10	4.17E-04	D7	3.71E-04	D4	0.002
D10	0.001	D9	4.17E-04	D6	3.71E-04	D3	0.002
D9	0.001	D8	4.17E-04	D5	3.71E-04	D1	0.002
D8	0.001	D4	4.17E-04	D4	3.71E-04	D9	4.85E-04
D7	0.001	D3	4.17E-04	D3	3.71E-04	D8	4.85E-04
D6	0.001	D2	4.17E-04	D2	3.71E-04	D7	4.85E-04
D5	0.001	D1	4.17E-04	D1	3.71E-04	D2	4.85E-04

Table 5.5: Documents' membership coefficients in the four identified topics.

5.5 Topic Extension Algorithm

The previous techniques are applied in EXTENDER's topic extension algorithm. Because retrieving and processing large numbers of Web pages is costly, EXTENDER applies a less expensive *distillation phase*, in which a series of queries is submitted to a search engine and only the information that is readily available from the search results (e.g. title, "snippet" of text, url, Open Directory Project summary) is used to identify good topic descriptors and discriminators. After this preliminary step, the best topic descriptors and discriminators are used as query terms in a *search phase* to search for additional material on the Web. The new set of search results is filtered according to the search context and then clustered to produce the next generation of artificial topics. The clustering algorithm returns a pair of matrices (DC and DT) codifying term membership in a topic and document membership in a topic. These two matrices are used to compose the new set of topics. During iteration I, for each topic $\mathbf{T}_k$ only terms t_j such that $\mathrm{TC}(k, j) > \mathcal{T}_{\mu^T}(\mathrm{I})$ are preserverd. Similarly, only documents d_j such that $\mathrm{DC}(k, j) > \mathcal{T}_{\mu^D}(\mathrm{I})$ are associated with topic $\mathbf{T}_k$. Tables 5.6 and 5.7 present a high-level description of this algorithm

```
PROCEDURE EXTEND-TOPIC
INPUT:
 M: source concept map
 s: total number of iterations
 $q_d$, $q_s$: number of queries submitted for distillation and search
 $n_d$, $n_s$: number of results for each distillation and search query
OUTPUT:
 A set of topics related to T
BEGIN
 Topics[0]= { M }
 for (i=0; i < s; i++)
 do
   Topics[i+1]=$\emptyset$.
   for each Topic T $\in$ Topics[i]
   do
     N = NEXT-GENERATION-OF-TOPICS(T, i,$q_d$,$q_s$,$n_d$,$n_s$)
     Topics[i+1]= Topics[i+1] $\cup$ N
 return Topics
END
```

Table 5.6: Pseudocode of the topic extension algorithm.

This section has described the application of our theoretical framework in the design of EXTENDER system. The component algorithms have been implemented in a robust prototype, and have been evaluated individually with good results. In the next chapter, we report a set of controlled studies to evaluate the techniques.

```
PROCEDURE NEXT-GENERATION-OF-TOPICS
INPUT:
 T: topic to extend
 i: present iteration
 $q_d$, $q_s$: number of queries submitted for distillation and search
 $n_d$, $n_s$: number of results per distillation and search queries
OUTPUT:
 N: A new set of topics
BEGIN
 //distillation
  Use the terms $t_j$ with highest $\lambda(\mathbf{T}, j)$ value to form $q_d$ queries
  Submit the queries to a search engine and collect $n_d$ results
  Use search result's "readily available information" to compute
   $\Lambda(\mathbf{T}, j)$ and $\Delta(j, \mathbf{T})$ for each term $t_j$
 //search
  Combine the terms $t_j$ with highest $\Delta(j, \mathbf{T})$ value and the terms with
   highest $\Lambda(\mathbf{T}, j)$ value to form $q_s$ queries
  Submit the queries to a search engine and collect $n_s$ document excerpts (Documents)
  D = COMPUTE-TERM-DESCRIPTIVE-POWER-IN-DOCUMENTS(Documents)
  L = COMPUTE-TERM-DISCRIMINATING-POWER-IN-DOCUMENTS(Documents)
 //filtering
  Only keep documents $d_j$ such that $\sigma(j, \mathbf{T}) \geq \mathcal{T}_\sigma$(i)
  Only keep terms $t_j$ such that $\Delta(j, \mathbf{T}) \geq \mathcal{T}_\Delta$(i) or $\Lambda(\mathbf{T}, j) \geq \mathcal{T}_\Lambda$(i)
  //clustering
  [DC,TC]= GENERATE-COHESIVE-TOPICS(L,D)
 //clean-up
  For each topic $T_k$ only keep terms $t_j$ such that $TC(k, j) > \mathcal{T}_{\mu^T}$
  For each topic $T_k$ only keep documents $d_j$ such that $DC(k, j) > \mathcal{T}_{\mu^D}$
  Collect resulting topics into set N
 return N
END
```

Table 5.7: Procedure for producing the next generation of topics.

6

Evaluation

In chapter 1 we formulated a number of hypotheses that provide the basis for the methods proposed in the last three chapters. The focus of this chapter is the empirical analysis of these hypotheses. In order to evaluate our hypotheses we performed three experimental studies. One of our studies involved the use of human subjects while the others consisted of semi-automatic evaluations.

The first study examines the models discussed in section 3.3. The goal of this study is to evaluate how the topology of a concept map affects the human rating of keywords occurring in a concept map. The statistical analysis for this experiment was performed by Thomas Reichherzer and the results have been published in [Leake et al., 2004a]. The second study evaluates the theoretical framework for topic generation discussed in section 4.2. The goal of this study is to examine the performance of our methods for the dynamic extraction of topic descriptors and discriminators. These results appear in [Maguitman et al., 2004b]. Finally, our third study evaluates EXTENDER's topic extension algorithm in terms of global coherence, coverage and novelty. These evaluations are reported in [Maguitman et al., 2004a].

6.1 Effects of Structure on Term Importance

In this section we present a human-subject evaluation of the models discussed in section 3.3. The purpose of this evaluation is to examine how the topology of a concept map affects the human rating of the keywords occurring in the map. This study was completed with encouraging results [Leake et al., 2004a], providing evidence for the following two hypotheses:

- Concepts that are closer to the root of a concept map are considered better descriptors of the topic of the map.
- Concepts with higher connectivity are considered better descriptors of the topic of the map.

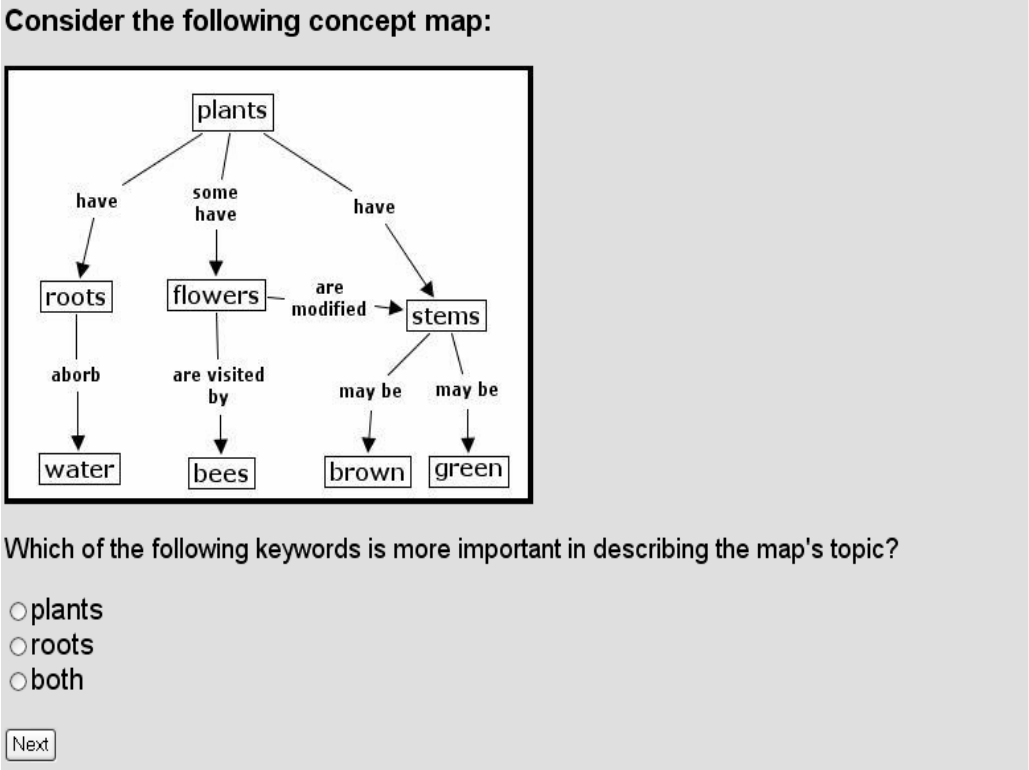
To carry out this study human subjects were first trained to familiarize themselves with concept maps. At the conclusion of the training phase, volunteers were presented with a sequence of simple concept maps and were asked to answer a series of questions. To answer each question the participants had to decide, given two keywords from a concept map, which one plays a more important role in describing the topic of a map. To analyze structure effects alone, we replaced the concept labels in real concept maps with artificial keywords minimizing the impact of common sense knowledge in the choices made by participants.

In addition to enabling us to evaluate the topology-based models, the inspection of the experimental data led us to choose suitable parameters for the CRD and HARD models both in terms of the node's distance to the root and its connectivity.

Method

Twenty paid subjects, all students admitted to Indiana University, were recruited by postings on electronic message boards and bulletin boards for a one-hour experiment conducted on the Web.

The experiment was divided into a training phase (to familiarize participants with the study and to provide background information on concept maps) and a test phase. In the training phase, participants were given a brief description of concept maps and their applications, and then asked to write a short summary of two concept maps from different domains. In the test phase, subjects answered 56 questions about a total of 12 small concept maps (fewer than 15 concepts each). The maps were designed with controlled differences in their topological structure and layout, to investigate the presence or absence of influences from particular types of changes (e.g., changing position of a node without affecting topology). Each question presented a concept map and two concepts selected from that map. Participants were asked to examine a map and to answer which of the two concepts best described the map's topic, or whether both described it equally well. To allow

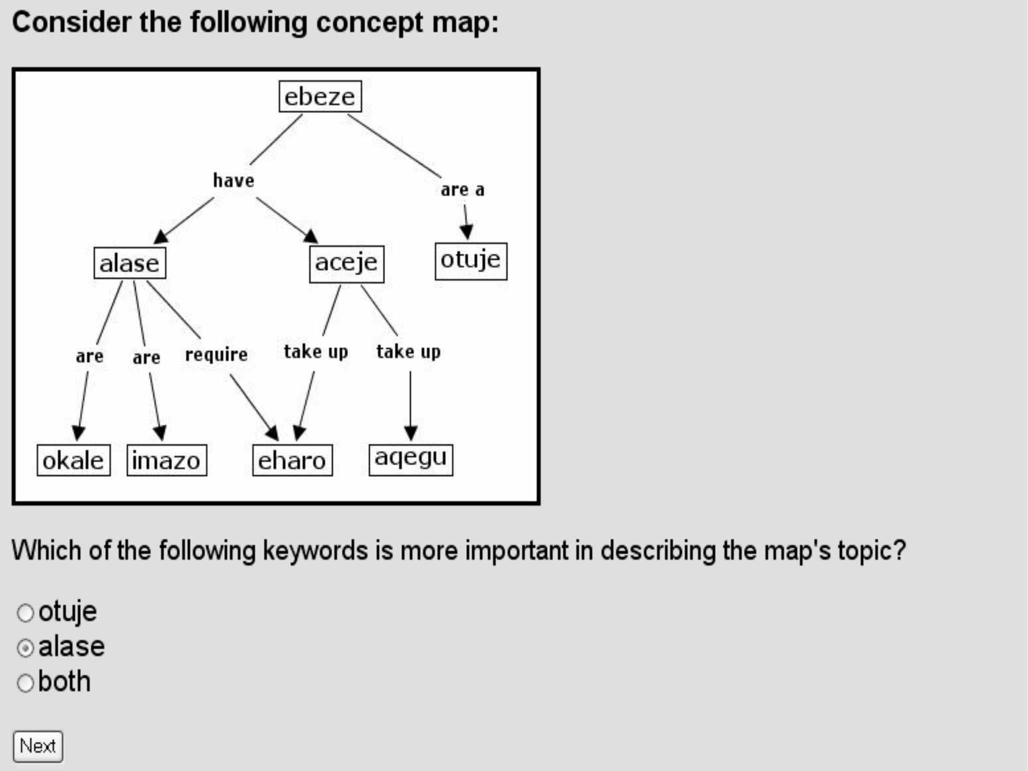


Consider the following concept map:

Which of the following keywords is more important in describing the map's topic?

○ plants
○ roots
○ both

Figure 6.1: Example of a training question based on a regular concept map.



Consider the following concept map:

Which of the following keywords is more important in describing the map's topic?

○ otuje
◉ alase
○ both

Figure 6.2: Example of a test question based on a concept map with artificial terms.

participants to first practice decision making on regular concept maps, the first 2 of the 12 concept maps used regular words in the concepts. Figure 6.1 is an example of a question based on a regular concept map. In the remaining 10 maps, concept labels were replaced with artificial and only responses concerning the latter 10 test maps were used in evaluating the models. An example of a question based on a concept map with artificial terms is presented in figure 6.2. The use of artificial terms as labels, the topological and layout changes between the concept maps, and randomization of the order of options to answer a question were all done to ensure that the participants made their choice independently of the concept maps they have already examined.

Influence	Significant	χ^2 Test of Independence
distance to root concept	yes	$(1, N = 40) = 17.04, p < 0.05$
concept connectivity	yes	$(1, N = 40) = 19.37, p < 0.05$
map layout	no	$(1, N = 40) = 0.23, p > 0.05$
direct, hub concept	yes	$(1, N = 40) = 7.74, p < 0.05$
direct, authority concept	yes	$(1, N = 40) = 15.82, p < 0.05$
indirect, hub concept	no	$(1, N = 40) = 3.73, p > 0.05$
indirect, authority concept	no	$(1, N = 40) = 3.73, p > 0.05$

Table 6.1: Statistical evaluation of influences on concept importance.

The concept maps in the experiment were designed to test specific hypotheses about the topological and layout factors that may influence subjects' evaluation of relevance of concepts to a concept map's topic. Because domain knowledge is absent, evaluations had to rely entirely on topology and layout.

Results

To test whether subjects' judgments of the importance of two concepts changed significantly from one map to another, we used a χ^2 test of independence when comparing the subjects' selections from two different maps. Table 6.1 summarizes the statistical results, which are discussed individually below.

Distance to root concept

To test the influence of distance to the root concept, subjects evaluated two concept maps in which the distance from a test concept to the root concept was changed from 2 to 1, by inserting an intermediate node. In a series of questions, subjects were asked to compare importances of the test concept, which was moved in the map's hierarchy, to the root concept and neighboring concepts of the moved concept. The results show that the root concept was considered most important compared to the other concepts, and that the importance of the test concept increased as it moved up the hierarchy. The differences in the selection of the moved concept over its neighboring concepts between the two concept maps were statistically significant.

Connectivity of a concept

To test the influence of connectivity, we used two concept maps which differed by increasing a test concept's connectivity—the number of incoming and outgoing connections to neighboring concepts—from 1 in the first map to 6 in the second. Subjects were asked to compare importances of the test concept to the root concept and the neighboring concepts of the modified concept. When the test concept's connectivity was increased, participants favored it over neighboring concepts and sometimes even over the root concept. All differences were statistically significant except for the preference over the root concept.

Layout of a map

To test whether a difference in layout affects subject's selections, two concept maps were constructed with identical topology but substantially different layout. The layout changes primarily involved horizontal organization, but in one instance a single concept was moved from the center right to the bottom left position. The questions asked for both layouts compared the concept that changed its position to its neighboring concepts. The statistical evaluation revealed that the layout changes had no significant affect on the concept ratings.

Direct and indirect influences of hub and authority nodes in a map

To test the effects of direct and indirect influences, a total of four concept maps were constructed with strong hub and authority concepts connected to other concepts in the map. The results showed that hub and authority concepts have an influence on the selection of concepts, and that authorities play a stronger role than hubs. However, the indirect influence of either a hub or authority concept on other concepts (when a hub or authority is indirectly connected to a test concept) did not significantly affect concept importance.

Fitting the Models to the Data

A hill-climbing algorithm was used to determine the parameter settings for the CRD and the HARD models which gave the best fit between the models and user data. Table 6.2 summarizes the chosen parameter values, the root-mean-square error (RMSE) of user and model data, and the cumulative error. The cumulative error is the percentage of the total questions (44 questions per subject, involving the 10 test concept maps) for which the models determine different responses from the subjects. To determine a model's preference between two concepts in a concept map, we compared the model's importance values for the two nodes. The model was considered to treat the concepts as equally relevant when their relevance values were within a fixed threshold of each other, for a threshold distance determined by hill-climbing. The last row of the table shows the RMSE and the cumulative error for a baseline model. In this model each concept in a map is rated equally important by assigning it a weight of 1.

The results show that the CRD model provides the best fit to the user data, followed by HARD and PF. All models except the baseline agree with more than 67% percent of the decisions reached

Model	Parameters for Best Fit			RMSE	Cumul.
	α	β	γ / δ		Error
CRD	0.930	4.959	3.603	0.072	27.5%
HARD	0	2.235	1.764	0.1487	32.8%
PF	N/A	N/A	N/A	0.170	27.8%
Baseline	N/A	N/A	N/A	0.564	66.8%

Table 6.2: Summary of model parameters and RMSE.

by the participants, who were in a few cases strongly divided in their vote for the best topic-describing concepts. For the remaining 33%, in most cases the models' predictions match the decisions of some subjects. Only once for the CRD model, twice for the HARD model, and four times for the PF model were model and user predictions entirely disjoint. Overall, CRD, HARD, and PF perform better than the baseline model.

Further analysis of the best-fit parameters for the CRD and HARD models supports the importance of nodes with many incoming connections. For the CRD model, nodes with incoming connections are more relevant than nodes with outgoing connections because their β is greater than α. Similarly, for the HARD model, nodes that play the role of authorities are more important than hub nodes.

Discussion

The reported experiments studied how topology and layout affect assessments of the importance of concepts within concept maps. They compared four candidate models which, using only analysis of a map's topology, compute a weight for each concept in a map. The computed weights provide an estimate of the importance of each concept as a descriptor of the topic of the map, according to subjects' judgments of topic importance.

This study highlights the importance of topological information, and also suggests that specific layout does not have a significant effect. It is also interesting to note that despite the importance of topology, local information alone was sufficient to account for the observed results. The CRD model, which considers distance from the root node and local connectivity, outperformed the more sophisticated HARD model, which takes indirect influences into account as well.

The current experiment studied small concept maps and therefore the best parameters reported for the CRD and HARD model may not generalize to larger maps. However, these results suggest that structure plays a surprisingly strong role, with structural information alone often sufficient to make high-quality predictions of human judgments of concept importance. Modeling such judgments helps elucidate the knowledge captured in concept maps and aids the development of intelligent support systems to provide relevant material during concept mapping.

6.2 Dynamic Extraction of Topic Descriptors and Discriminators from Unstructured Text-Data

It is relatively simple to evaluate the effectiveness of techniques for selecting good discriminators to use as query terms. This can be done by providing an approximate measure of the relevance of the retrieved documents (e.g., by measuring the mean similarity between the retrieved documents and the source) and using that relevance measure to compare the performance of the new technique against baseline techniques. In this section we report a controlled study to evaluate the distillation method for query formation proposed in section 4.2. However, it is more difficult to develop objective measures for evaluating term descriptive power. The study reported in section 6.1 provides evidence for the significance of topological factors in human assessments of concept descriptive power in concept maps. Because topological factors are good predictors of human assessments of concept descriptive power, they provide a good standard for evaluating the effectiveness of techniques used to identify good topic descriptors—provided we have access to a concept map representation of the topic as a starting point. In our study we propose the use of existing concept map libraries as data for assessing term descriptive power.

Evaluating the Descriptor Extraction Method

We took advantage of the fit of the PF model to human data to perform an indirect evaluation of the descriptor extraction method by means of concept maps. We decided to use the PF model instead of the CRD or HARD models because the PF model is non-parameterized, but still a good predictor of human assessments of concept descriptive power in concept maps. As data we used the Mars 2001 knowledge model, a large multimedia knowledge model on Mars (http://www.cmex.arc.nasa.gov), constructed entirely by NASA scientists using CmapTools [Briggs et al., 2004]. The Mars 2001 knowledge model contains 118 concept maps and 3654 concepts. Our goal in this evaluation was to test if the descriptor extraction method discussed in section 4.2 was able to predict the weights assigned by the PF model.

We used each concept in a concept map to submit a query to GOOGLE (using the GOOGLE Web API) and up to 20 results were collected for each query (approximately 600 Web pages were collected for each concept map). The queries were constructed using all the terms in a concept label, after stop-word filtering and disregarding the topological role of the concept in the map. For example, a concept with the label "Search for evidence of Past Life" was presented to GOOGLE as '*search* AND *evidence* AND *past* AND *life*'. For each concept map **M** in the Mars 2001 project we tested if the descriptor-extraction method was able to predict the topological term weighting suggested by the PF model. In order to do so, given a concept map **M** and a collection of retrieved Web pages, we computed the $\Lambda(\mathbf{M}, t)$ measure (defined in section 4.2) for each term in the collection. Results were compared to a baseline model in which all terms in a map were assigned the same weight.

The RMSE between the PF model data and the descriptor-extraction method (Λ) was of 0.237 while the RMSE between the PF model and the baseline model was 0.824. Table 6.3 summarizes the RMSE for each test. In addition, the Pearson correlation coefficient between the PF model weighting and that of the descriptor-extraction method was 0.42 for 6901 pairs, where the pairs contain the PF and Λ weights of the terms found in the Mars 2001 knowledge model. This result indicates a statistically significant correspondence between the two weighting schemes. Hence, by transitivity, the combination of this result with the results obtained in the human subject experiment reported in section 6.1 suggests a considerable correspondence between human judgments of concept descriptive power and the data returned by the descriptor-extraction method. This correspondence is encouraging for the hypothesis that the proposed method provides good predictions on the importance of terms in describing a topic.

Model	User Data	Λ	Baseline
PF	0.170	0.237	0.824

Table 6.3: Summary of RMSE of PF compared to user data, Λ, and baseline.

As a sidenote, it is interesting to note that the Pearson correlation coefficient between the PF model weighting and that of the discriminator-extraction method was only 0.01. This result reflects the fact that topology alone is not a very good predictor of term discriminating power, highlighting the need to recognize descriptive power and discriminating power as separate notions of term importance.

Evaluating the Distillation Method

In order to test the distillation method for query formation, we used again the Mars 2001 knowledge model. For each map, a baseline static method and three different dynamic feature selection methods were applied to select query terms. We use *Inverse Map Frequency* (IMF) as the baseline static feature selection method. IMF is an adaptation of the IDF weighting scheme [Salton and Yang, 1973], designed to measure the overall rarity of a term in a knowledge model. Each term t in a map was weighted as $IMF(t) = \log \frac{1+|\mathcal{K}|}{|\mathcal{K}_t|}$, where $|\mathcal{K}|$ represents the number of concept maps in the knowledge model (118 for $\mathcal{K}$ = "Mars 2001") and $|\mathcal{K}_t|$ stands for the number of concept maps containing term t. IMF was used to sort the terms occurring in a concept map and to generate queries of incremental size, starting from a query of size 1 consisting of the most highly weighted term and incrementally adding the next most highly weighted terms.

The dynamic weighting schemes evaluated here are three variations on the framework for query distillation proposed in section 5.5. We refer to these methods as *Dynamic Basic* (DB), *Dynamic Concept-Root* (DCR), and *Dynamic Concept-Root-Disjunction* (DCRD). All three methods are based on the algorithm discussed in section 5.5, but differ on how the queries are constructed for each concept in a concept map. Consider a concept map with concept root whose label consists of terms $r_1, r_2, \ldots, r_x$. Given a concept c with terms $t_1, t_2, \ldots, t_y$ the three types of queries associated with c are the following:

DB: t_1 AND t_2 AND ... AND t_y.

DCR: t_1 AND t_2 AND ... AND t_y AND r_1 AND r_2 AND ... AND r_x.

DCRD: (t_1 AND t_2 AND ... AND t_y AND r_1 AND r_2 AND ... AND r_x) OR t_1 OR t_2 OR ... OR t_y OR r_1 OR r_2 OR ... OR r_x.

Because GOOGLE limits queries to 10 words, we truncated those queries that resulted in more than

10 term occurrences. In our evaluation we constructed a query for each concept in a concept map and considered up to 30 returned results per query. The search results associated with a concept were divided into 3 sets of equal size. In a three-stage evaluation, we used one of the three sets for query distillation and the other two for testing, rotating the roles of the sets at each stage. For each stage, the distillation data was used to compute an approximation of the discriminating power Δ (discussed in section 4.2) of each term. Only the information readily available from the search results (snippets, etc.) was used in the distillation phase. The query involving terms with highest Δ value was identified as the *most promising query*, as done in the algorithm of section 5.5. To test the query distillation method we selected from the testing data the remaining two sets of returned results (i.e., the search results not used for query distillation) associated with the most promising query and used those sets for performance analysis of the corresponding dynamic method.

To evaluate the performance of our methods, we took the full documents associated with the returned results, and computed their mean similarity to the source concept map. Similarity was measured as the proportion of novel terms (terms not in the query) in a retrieved document that are also part of the source map. Given a set **Q** of terms in a query, a set **M** of terms in a source map, and a set **D** containing the terms of a query result, the similarity of the query result to the source map can be measured by:

$$\mathbf{Similarity}^{N}(\mathbf{M}, \mathbf{D}, \mathbf{Q}) = \frac{|(\mathbf{D} \cap \mathbf{M}) - \mathbf{Q}|}{|(\mathbf{D} \cup \mathbf{M}) - \mathbf{Q}|}.$$

Measure $\mathbf{Similarity}^{N}$ is an adaptation of the *Jaccard coefficient*. It computes the proportion of terms in the source map or in a retrieved result that are in both the map and the retrieved result but are not in the query. If the set of search results for a given query is empty, the value for that query is considered to be 0.

In order to control for query size when comparing the performance of the dynamic methods against IMF, we set the size of the IMF queries to the number of terms occurring in the conjunctive portion of the corresponding dynamic-method query.

Figures 6.3, 6.4, and 6.5 compares performance of the three dynamic methods to the IMF method. Each concept map in the Mars 2001 project corresponds to a trial and is represented by a point. The point's horizontal coordinate corresponds to the average performance of IMF for that case, while the vertical coordinate corresponds to the average performance of the dynamic method. In this evaluation DB outperforms IMF in 74% of the cases, DCR outperforms IMF in 77% of the cases, and DCRD outperforms IMF in 64% of the cases. In particular, there are several cases in which queries formed using the IMF method resulted in no search results. This highlights one of the main advantages of using a dynamic approach involving a distillation phase to discover which are the most useful terms to use in a query. In Tables 6.4, 6.5 and 6.6 we present the mean similarity confidence interval resulting from each of the dynamic methods, and we compare it against the mean similarity confidence interval resulting from applying the IMF method with query size adjusted as we explained above. These comparison tables show that the three dynamic methods result in statistically significant improvements over IMF.

Method	N	MEAN	STDEV	SE	95% C.I.
DB	118	0.2196	0.0645	0.0059	**(0.2079, 0.2311)**
IMF	118	0.1627	0.1563	0.0144	**(0.1345, 0.1909)**

Table 6.4: DB vs. IMF: confidence intervals for the mean similarity to source map.

Method	N	MEAN	STDEV	SE	95% C.I.
DCR	118	0.3111	0.0893	0.0082	**(0.2950, 0.3272)**
IMF	118	0.1798	0.2037	0.0188	**(0.1430, 0.2165)**

Table 6.5: DCR vs. IMF: confidence intervals for the mean similarity to the source map.

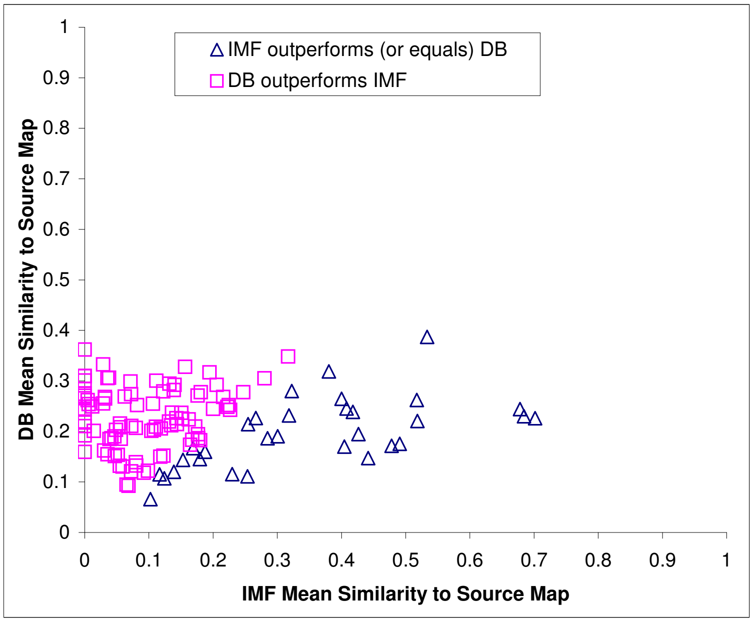


Figure 6.3: Average similarity to source map of documents retrieved using IMF vs. DB.

Method	N	MEAN	STDEV	SE	95% C.I.
DCRD	118	0.2498	0.0903	0.0083	**(0.2335, 0.2661)**
IMF	118	0.1880	0.1955	0.0180	**(0.1527, 0.2232)**

Table 6.6: DCRD vs. IMF: confidence intervals for the mean similarity to the source map.

Discussion

In this section we presented a semi-automatic evaluation of our framework for the dynamic extraction of topic descriptors and discriminators. The reported results highlight the advantage of using a dynamic distillation approach for query formation: Queries formed using terms that tend to occur only in similar pages resulted in higher precision than queries that were formed using terms with high IMF value.

The fact that the dynamic methods rely on the submission of a first round of queries (distillation phase) to approximate a term's discriminating power suggests that they are less efficient than the

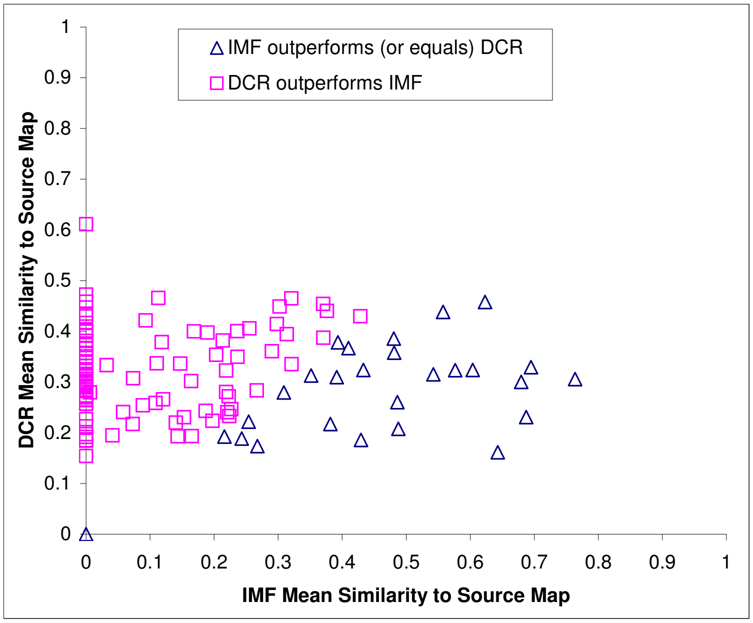


Figure 6.4: Average similarity to source map of documents retrieved using IMF vs. DCR.

static approaches. However, given that knowledge will be extended incrementally during the concept mapping process, multiple rounds of queries will be submitted in any case, and the generation of second-round and subsequent queries can significantly benefit from examining previous search results, at a small additional cost.

During EXTENDER's first cycle, a term's descriptive power is obtained directly from the topology of the source map. However, for subsequent iterations, when topics are compiled as topology-free bags of terms, extracting good topic descriptors dynamically is important. When the system presents the final generation of topics to the user, the topic descriptors are used to produce labels for the suggested topics. The results reported in this section suggest that our methods for the dynamic extraction of topic descriptors are good predictors of human assessments of term descriptive power.

The evaluation presented in this section took a bottom-up approach, focusing on the ability

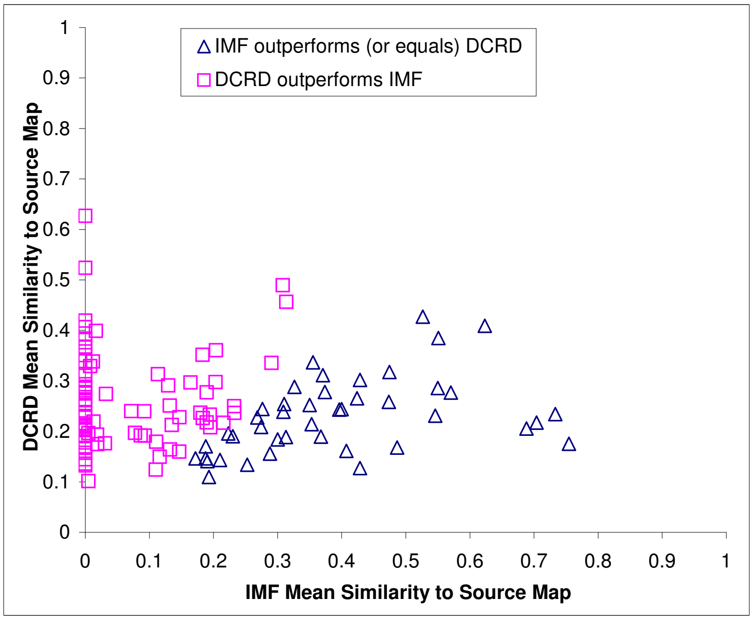


Figure 6.5: Average similarity to source map of documents retrieved using IMF vs. DCRD.

of EXTENDER to find good topic descriptors and discriminators at each step of its process. The next section examines EXTENDER's performance in the light of the desiderata for topic suggestion discussed in section 5.2.

6.3 EXTENDER Global Coherence, Coverage and Novelty

The performance of EXTENDER is hard to assess in a controlled way because the usefulness of topic suggestions is highly subjective. In order to perform an objective test we evaluated whether the system was able to generate artificial topics with content similar to hand-crafted ones. As the hand-crafted topics, we used the set of concept maps in the Mars 2001 knowledge model.

In our tests the top-level concept map from the knowledge model was used as the starting point (corresponding the map under construction) and EXTENDER's topic extension algorithm was used to produce a collection of artificial topics, without access to any of the other maps in the knowledge model. As a baseline method for comparison we implemented a simple algorithm which constructs queries using all the concepts from the same concept map EXTENDER used as a starting point, submits them as queries to the Google Web API, and clusters the results to generate topics.

We expected EXTENDER's mechanism to provide results with superior global coherence, novelty, and coverage for equal number of Web queries. The data obtained from this analysis is used to test the following hypotheses:

- Using the search context to maintain the relationship between the set of generated topics and the initial concept map helps to preserve global coherence, ensuring that the system maintains its focus on topics relevant to the initial concept map.
- The use of the curiosity mechanism to incrementally search the Web increases novelty and coverage compared to a baseline mechanism that generate the same number of queries directly from the originating knowledge model.

An evaluation based on global coherence and coverage requires an operational definition of *topic relevance*. Here, we consider the expert-generated Mars 2001 topics as *target topics*, with the relevance of a system-generated topic measured by the accuracy with which a system-generated topic replicates an expert-generated topic. Note that the accuracy measure also provides an indication of topic quality, because its results depend on the similarity between EXTENDER's topics and the expert-generated set, which we expect to be of good quality for the domain.

The measures of accuracy, coherence and coverage are formalized in the next section.

Criterion Functions for Evaluating a Topic Generation Strategy

To measure global coherence assume that $R = \{r_1, \ldots, r_m\}$ is a target set of relevant topics and $A = \{a_1, \ldots, a_n\}$ is a set of topics generated by the topic-generation strategy under evaluation. Similarity between topics a_i and r_j can be measured using, for example, the *Jaccard coefficient*, definded as:

$$\textbf{Similarity}(a_i, r_j) = \frac{|a_i \cap r_j|}{|a_i \cup r_j|}.$$

Then, we can define the *accuracy* of topic a_i in R as follows:

$$\textbf{Accuracy}(a_i, R) = \max_{r_j \in R} \textbf{Similarity}(a_i, r_j).$$

The **Accuracy** function measures the precision with which a given topic replicates some topic in a given set of topics.

We use the **Accuracy** function to define **Global_Coherence** as follows:

$$\textbf{Global\_Coherence}\ (A, R) = \frac{\sum_{a_i \in A} \textbf{Accuracy}(a_i, R)}{|A|}.$$

The **Global_Coherence** function measures the fraction of relevant topics that has been generated, weighted with the level of accuracy with which relevant topics are actually generated. The notion of global coherence is a generalization of the IR notion of precision, and as such, it has its limitations. This criterion function can be maximized if the system generates a single artificial topic identical to some relevant topic, which clearly does not guarantee acceptable topic generation performance. Hence, a *coverage* factor must be introduced to favor topic-generation strategies that cover many topics of a target set of relevant topics. To address this issue, we define a criterion function able to measure *coverage* as a generalization of the standard IR notion of recall:

$$\textbf{Coverage}\ (A, R) = \frac{\sum_{r_i \in R} \textbf{Accuracy}(r_i, A)}{|R|}.$$

Because novelty is one of our desiderata for topic generation, we want to favor strategies that produce relevant topics with a high number of novel terms. Consider the set o, containing the terms of the originating topic, i.e., the knowledge model that is used as a starting point to search for topics. We propose a modified similarity measure reflecting the proportion of *novel terms* (terms not in the starting knowledge model) in a system-generated topic a_i that are also part of an r_j from a set of relevant topics:

$$\textbf{Similarity}^N(a_i, r_j, o) = \frac{|(a_i \cap r_j) - o|}{|(a_i \cup r_j) - o|}.$$

The accuracy function can be rewritten in terms of the new similarity function, to measure the precision with which a given topic replicates some topic in the given set, disregarding those terms that are in the starting knowledge model:

$$\textbf{Accuracy}^N(a_i, o, R) = \max_{r_j \in R} \textbf{Similarity}^N(a_i, r_j, o).$$

We use this accuracy function to define a measure of global coherence that accounts for novelty:

$$\textbf{Global\_Coherence}^N(o, A, R) = \frac{\sum_{a_i \in A} \textbf{Accuracy}^N(a_i, o, R)}{|A|}.$$

Analogously, the coverage measure can be re-stated as

$$\mathbf{Coverage}^N(o, A, R) = \frac{\sum_{r_i \in R} \mathbf{Accuracy}^N(r_i, o, A)}{|R|}.$$

Parameter Settings

EXTENDER's methods depend on parameters such as the number of iterations (generations of topics), the number of queries submitted from the source concept map and from each generated topic, the maximum number of topic descendants for each topic, the starting and stopping thresholds for curiosity mechanisms and the similarity threshold for merging topics. This results in a large parameter space. In practice, however, pragmatic concerns for the interface, such as the desire for rapid response and low memory use, suggest constraining some parameters. Accordingly, our tests limited the number of generations to 4, the number of queries from each topic to 20 for distillation and 10 for search, and the number of topic descendants at each stage to 8.

Experimental Results

We first analyzed the performance of EXTENDER as a function of the number of iterations. The test was performed for 1, 2, 3 and 4 iterations. For each number of iterations our evaluation involved 48 trials, with different settings for EXTENDER's parameters. Table 6.7 and figure 6.6 summarize the highest performances attained by EXTENDER in each of the cases. We observed that in general three iterations appears sufficient to generate a rich variety of topics with the system response time kept below 20 seconds. A smaller number of iterations significantly decreases coverage of novel material, while it usually increases global coherence.

When comparing the performance of EXTENDER against the baseline, we set the number of

Number of Iterations	Global Coherence	Coverage	Global Coherence (Novel Material)	Coverage (Novel Material)
1	0.371428	0.039718	0.666667	0.053158
2	0.193281	0.057206	0.502954	0.143117
3	0.177684	0.059784	0.433845	0.264514
4	0.171254	0.059856	0.422741	0.269998

Table 6.7: Highest performance for EXTENDER's topic generation algorithm as a function of the number of iterations.

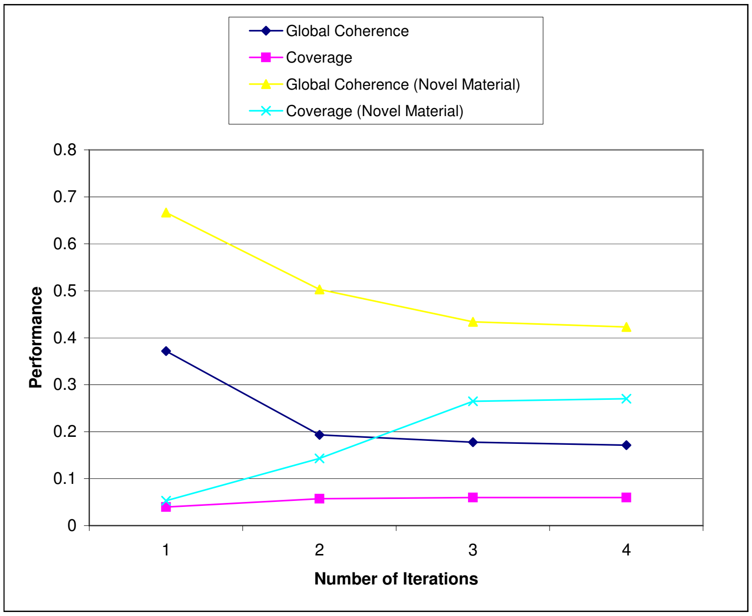


Figure 6.6: Highest performance for EXTENDER's topic generation algorithm as a function of the number of iterations.

EXTENDER's iterations to 3 and the number of queries for the baseline to the total number of queries submitted by EXTENDER. For each trial, EXTENDER and the baseline method used the same similarity threshold and method for merging topics.

Figures 6.7 and 6.8 compare the performance of EXTENDER's topic generation algorithm to the baseline method in terms of global coherence and coverage. Figures 6.9 and 6.10 present a comparison between EXTENDER and the baseline method that also accounts for novelty. A particular setting corresponds to a trial and is represented by a point. The point's horizontal coordinate corresponds to the performance of EXTENDER for that case, while the vertical coordinate corresponds to the performance of the baseline method. In Tables 6.8, 6.9, 6.10, and 6.11 we present the mean confidence interval resulting from computing the performance criterion functions for EXTENDER and the baseline method. These comparison tables show that EXTENDER results in statistically significant improvements over the baseline method.

Table 6.12 summarizes the parameter settings for EXTENDER's highest performance according to each of the criterion functions used for this evaluation. Because of the pragmatic concerns mentioned earlier, the number of queries from each topic was limited to 20 for distillation and 10 for search and the maximum number of topic descendants at each stage was set to 8. In all cases the highest performance was obtained when EXTENDER used the maximum number of queries for distillation and search. The highest performance in terms of global coherence and coverage was achieved when the number of topic descendants at each stage was set to 4 and 8 respectively. We also searched for the best values for parameter p used in the co-clustering algorithm for computing $\rho^{[\Lambda\Delta]}(i,j)$, the representation value of a term t_i in the topic of a document d_j:

$$\rho^{[\Lambda\Delta]}(i,j) = \Lambda(j,i) \times \Delta(i,j)^p.$$

The search was made for p taking the values 0.25, 1, 4 and 8. The highest performance was consistently achieved for $p = 4$. Similarly, we searched for the best value for parameter q used in the computation of $\rho^{[\Xi\Phi]}(j,i)$, the representation value of a document d_j in the topic of term t_i:

$$\rho^{[\Xi\Phi]}(j,i) = \Xi(i,j) \times \Phi(j,i)^q.$$

Again the analysis was made for values 0.25, 1, 4 and 8. In this case the highest performance for global coherence resulted from $q = 1$, while the highest performance for coverage was achieved for $q = 4$. We also searched for the best starting and stopping threshold parameters used in the curiosity mechanisms for the survival of descriptors and discriminators and for filtering documents. The search space was limited to values between 0 and 0.4. The results presented in table 6.12 show that higher thresholds favor global coherence while lower thresholds favor coverage. This agrees with our expectations: if only closely related material is collected, then the system will be able to maintain its focus on relevant topics. On the other hand, if more terms and documents are collected, then coverage increases.

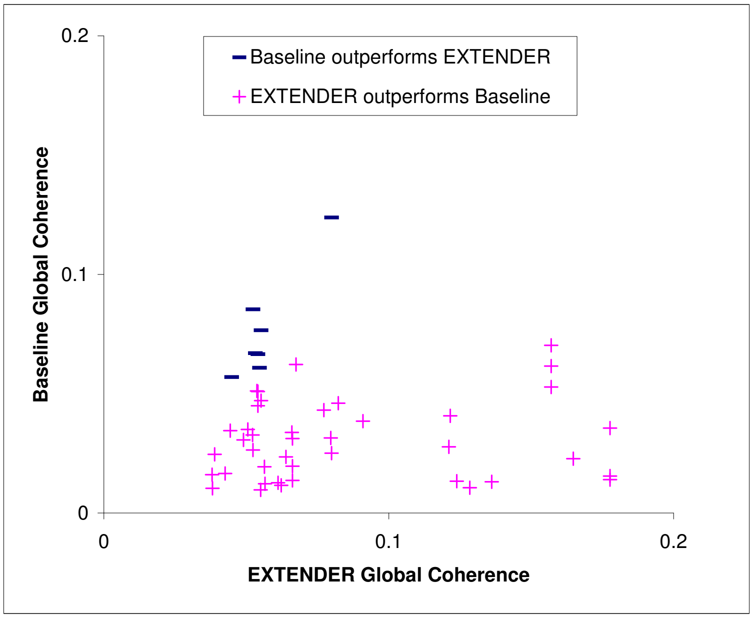


Figure 6.7: EXTENDER Global Coherence vs. Baseline Global Coherence.

Method	N	MEAN	STDEV	SE	95% C.I.
EXTENDER	48	0.082	0.043	0.006	**(0.069, 0.094)**
Baseline	48	0.037	0.024	0.003	**(0.03, 0.044)**

Table 6.8: Confidence intervals for the mean global coherence of EXTENDER and baseline.

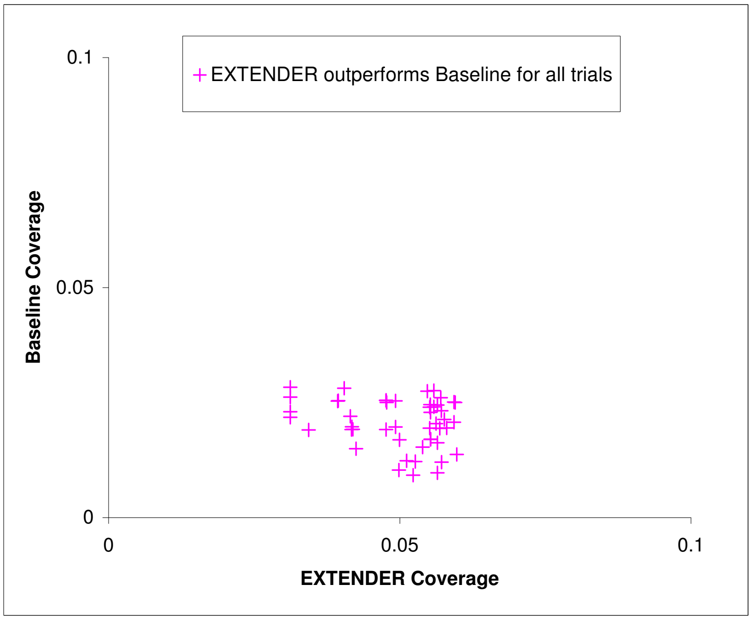


Figure 6.8: EXTENDER Coverage vs. Baseline Coverage.

Method	N	MEAN	STDEV	SE	95% C.I.
EXTENDER	48	0.05	0.009	0.001	**(0.047, 0.052)**
Baseline	48	0.02	0.005	0.001	**(0.02, 0.022)**

Table 6.9: Confidence intervals for the mean coverage of EXTENDER and baseline.

Discussion

In this section we performed an objective test for evaluating the performance of EXTENDER's topic generation strategy. We proposed a set of criterion functions for evaluating topic generation in terms of global coherence, novelty and coverage. A performance evaluation through these criterion functions requires access to a target set of relevant topics. In our scenario, generating new topics from Web searches, we do not have access to a predefined set of relevant topics. In order to provide an approximation of the set of relevant topics we used an expert-generated set of concept maps on Mars as our "gold standard". As a consequence, the notion of relevant topic is

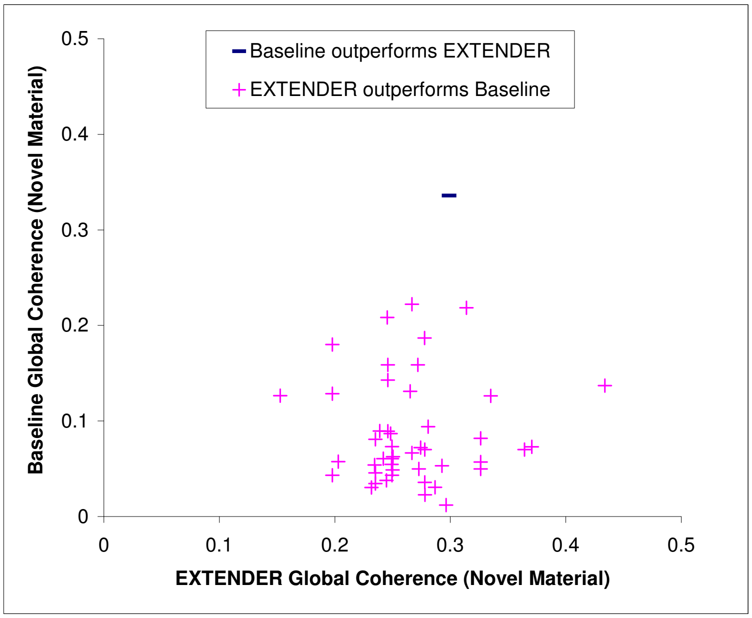


Figure 6.9: EXTENDER Global Coherence vs. Baseline Global Coherence (Novel Material.)

Method	N	MEAN	STDEV	SE	95% C.I.
EXTENDER	48	0.267	0.05	0.007	**(0.253, 0.281)**
Baseline	48	0.101	0.085	0.012	**(0.077, 0.125)**

Table 6.10: Confidence intervals for the mean global coherence of EXTENDER and baseline considering novel material only.

defined relative to our corpus of topics represented by concept maps in the Mars knowledge model. Despite the fact that our evaluation is only partial, our tests provide substantial evidence showing that EXTENDER's approach significantly outperforms a baseline at recovering topics close to those of an expert's hand-coded knowledge model.

When we analyzed the relationship between parameter settings and EXTENDER's results we noticed that different parameter settings favor different aspects of EXTENDER's performance. For example, higher thresholds for the curiosity mechanism favor global coherence while lower thresholds favor coverage. Therefore, these parameters could be adjusted, depending on whether the goal

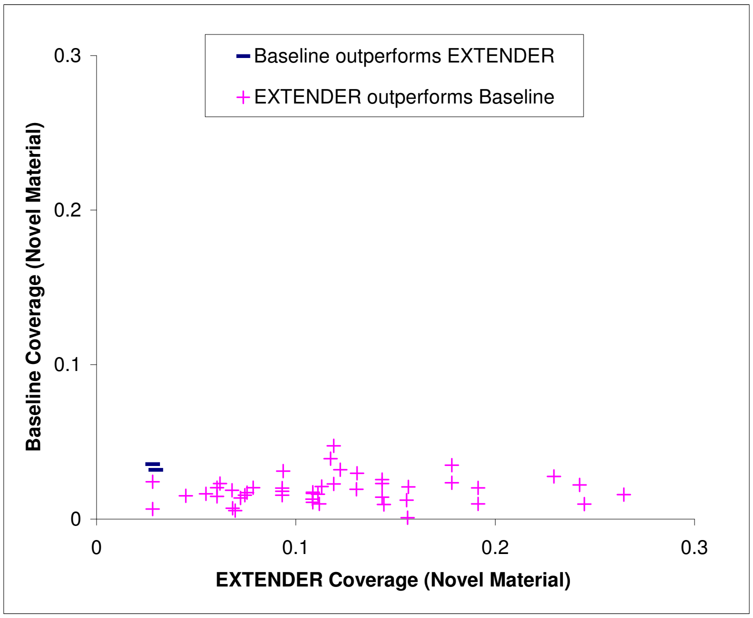


Figure 6.10: EXTENDER Coverage vs. Baseline Coverage (Novel Material.)

Method	N	MEAN	STDEV	SE	95% C.I.
EXTENDER	48	0.116	0.059	0.008	**(0.099, 0.132)**
Baseline	48	0.019	0.009	0.001	**(0.017, 0.022)**

Table 6.11: Confidence intervals for the mean coverage of EXTENDER and baseline considering novel material only.

is to focus on topics more or less similar to the user's current topic. These results shed light on several issues, helping us to improve the design of both EXTENDER's algorithm and EXTENDER's interface.

Parameter	Global	Coherence	Global Coherence	Coverage
		Coverage	(Novel Material)	(Novel Material)
Queries for distillation	20	20	20	20
Queries for search	10	10	10	10
Topic Descendants	4	8	4	8
Value of p in $\rho^{[\Lambda\Delta]}$	4	4	4	4
Value of q in $\rho^{[\Xi\Phi]}$	1	4	1	4
Starting threshold for $\mathcal{T}_{\boldsymbol{\Lambda}}$	0.1	0	0.1	0
Stopping threshold for $\mathcal{T}_{\boldsymbol{\Lambda}}$	0.2	0.1	0.2	0.1
Starting threshold for $\mathcal{T}_{\boldsymbol{\Delta}}$	0.1	0	0.1	0
Stopping threshold for $\mathcal{T}_{\boldsymbol{\Delta}}$	0.2	0.1	0.2	0.1
Starting threshold for $\mathcal{T}_{\boldsymbol{\sigma}}$	0.2	0.1	0.2	0.1
Stopping threshold for $\mathcal{T}_{\boldsymbol{\sigma}}$	0.4	0.2	0.4	0.2

Table 6.12: Best parameters for EXTENDER's topic generation algorithm.

7

Conclusions

7.1 Review

An important question in knowledge management is how to determine the information to capture and how to capture it. In traditional views, knowledge capture may be seen primarily as acquiring knowledge that exists within the expert. In this dissertation we have presented methods for supporting an alternative approach, “knowledge extension,” based on the premise that a knowledge model evolves from coordinated processes of knowledge acquisition and knowledge construction. In this view, it is crucial to support experts’ construction of new knowledge as they extend existing knowledge models. This dissertation has addressed these needs by studying and evaluating methods that use information automatically extracted from a knowledge model under construction to search the Web for novel but relevant topics. Using these methods, we have developed EXTENDER, a support tool that starts from a concept map and automatically produces a set of suggestions for topics to include, proactively supporting users as they extend knowledge models.

Searching the Web to support knowledge extension presents new challenges. This search problem requires:

- **Methods that can identify terms that best describe the user's context.** In this dissertation, we have proposed three models of the influence of concept maps' topology on concept importance. EXTENDER applies topological analysis to the starting knowledge model to identify an initial set of terms that are good descriptors of the user's current concept map. Our experimental studies show that the models used by EXTENDER to identify good descriptors in concept maps are good predictors of human-assessments of concept importance.
- **Search methods for the dynamic extraction of good topic representatives.** We have proposed a framework for the dynamic extraction of topic descriptors and discriminators to aid information search in the context of a knowledge model under construction. In this framework, we represent the relationships between terms and documents using hypergraphs and study a series of dual notions that reflect interesting properties of terms and documents. Our framework suggests that terms are good topic descriptors if they occur often in documents similar to the topic, while terms are good discriminators if they occur primarily in similar documents. EXTENDER dynamically extracts topic descriptors and discriminators for query formation and term-weight reinforcement. Experimental studies described in this dissertation indicate a considerable correspondence between human judgments of concept descriptive power and the results returned by our descriptor-extraction methods. Our evaluations also indicate that the proposed methods for the extraction of topic discriminators result in statistically significant improvement over traditional approaches when applied to the task of retrieving material similar to the current context.
- **Search methods that can identify candidate topics with the right balance of relevance and novelty.** EXTENDER searches for novel but related topics through an iterative process of Web

search, context-based filtering, and clustering. This dissertation proposes criterion functions for measuring the *coverage* and *global coherence* of a topic generation strategy. These criterion functions are a natural adaptation of the commonly used measures of precision and recall to the topic generation scenario. The evaluations based on coverage and global coherence reported in this work show that EXTENDER's methods result in statistically significant improvements over a baseline method at recovering novel topics close to those of an expert's hand-coded knowledge model. Data collected during these evaluations has been used to tune-up EXTENDER's methods and to design a user interface to easily adapt the methods to individual needs.

7.2 Broader Applicability

EXTENDER has been developed as part of a knowledge modeling support system within the framework of CmapTools. However, the generality of the proposed algorithms makes them applicable to a broad class of tasks:

- **Going beyond previously captured information.** EXTENDER's task is an instance of a more general one: to suggest material that is novel but related to a user's context. Search engines are more appropriate than this kind of suggester when the user knows *what* to seek and *how* to seek it. However, sometimes a system may need to go beyond the known user desires, to automatically form suitable queries and find what might be useful for the user. This kind of system can reveal similarities that were not previously apparent and present a "big picture" that can give the user a broader understanding of the current task.
- **Augmenting the user's memory.** The Web is a rich collective memory repository. A suggester system that incrementally searches this repository to find material that is useful to the user's

current task can act as a memory augmentation aid. By an association of similarities, this aid can help users (1) remember information, (2) assure that areas relevant to the current task have been considered, and (3) pursue new directions.

- **Automatic query refinement.** Because Web search engines restrict queries to a small number of terms (e.g., the 10-term limit for Google), human-generated queries cannot reflect extensive contextual information. For human-generated queries, users frequently decide, based on initial results, to refine subsequent queries. If contextual information is available, part of the query formation and refinement process can be automated using techniques proposed in this dissertation. Our methods for the dynamic extraction of topic descriptors and discriminators are not restricted to concept maps but are applicable to any form of textual representation.
- **Finding good index terms.** Good topic descriptors can be identified by searching for terms that occur often in documents similar to the given topic. As shown in chapter 3, human assessments of term descriptive power in a topic are in good correspondence with this notion. Because the best descriptors for a topic are the most commonly used terms in the context of that topic, it is reasonable to expect them to be the same terms people will use when searching for material on that topic. Therefore, our techniques for finding good topic descriptors can be applied to the generation of indices. Our techniques enable a document on a topic to be indexed under terms that are good descriptors for that topic, even when the terms are absent from the document.

7.3 Further Research Avenues

This research work opens up many research avenues:

- **Implementing a non-real time topic suggester.** One of the most important characteristics of EXTENDER is its ability to provide suggestions to the user on real time. To achieve this, EXTENDER relies entirely on Google to search the Web for topics and uses only the information readily available from the search results (e.g., snippets) to generate suggestions—it does not crawl the Web or parse documents. An alternative approach would perform a more intensive and careful analysis, by collecting links associated with initial search results, and performing different kinds of content and link analysis on the collected pages. This alternative approach would help to identify topically coherent subgraphs in the Web and would also enable a more informed decision-making to filter documents and terms. While such an approach may not be worth pursuing in practice for implementing a usable tool—long delays on topic suggestions would make the use of EXTENDER less attractive—it could provide some interesting new insight on the topic extraction and extension problem. In addition, a non-real time topic suggester could be useful for certain off-line analysis tasks (e.g., it could provide support for building topical indices).

- **Exploiting semantic information sources.** EXTENDER operation could be extended to take advantage of several semantic information sources available on the Web. For instance, it could greatly benefit from information available on hand-coded topic directory services (e.g. Dmoz or the Yahoo Web site directory). Directory services usually include an ontology of topics that can be used to identify similar topics and similar pages. This kind of similarity, usually called semantic similarity, is extremely valuable because it comes directly from human hand-coded classifications. EXTENDER methods could be augmented, to search on directory services for topics similar to a user's context, as well as additional semantically related material.

- **Integrating EXTENDER with lexical databases.** Currently, EXTENDER methods rely on exact term matching. An area of future research is the use of WordNet or similar electronic lexical databases to enable the system discover a wider range of relevant topics using information on synonyms.

- **End-to-end human-subjects evaluation.** User studies that directly test the usefulness of EXTENDER suggested topics during the knowledge model extension process could help us to further refine our methods. However, a study based on monitoring the user interaction with EXTENDER would be insufficient to test the ability of EXTENDER to provide useful suggestions. On many occasions EXTENDER's suggestions could jog the user's memory and help the user pursue new directions, even when the suggested topics are not selected for inclusion.

7.4 Concluding Remarks

Capturing expert knowledge is an essential component of the knowledge management process. In light of the difficulties in capturing knowledge through traditional knowledge engineering processes, it is important to facilitate the knowledge capture process through methods that allow more direct and natural interaction between system and user.

The research presented in this dissertation combines aspects of knowledge acquisition with knowledge construction, for a knowledge extension approach to knowledge management. By searching the Web, EXTENDER provides a tremendous resource for the knowledge modeling process.

Tools enabling experts to directly capture their own knowledge, augmented with intelligent support, hold great promise for transforming how users capture new knowledge, refine old conceptualizations, and seek to better understand a domain. We hope the methods proposed in this work provide a solid base for further studies into this new, fascinating and important area.

Bibliography

[Abiteboul et al., 2000] Abiteboul, S., Buneman, P., and Suciu, D. (2000). *Data on the Web: from relations to semistructured data and XML*. Morgan Kaufmann Publishers Inc.

[Abiteboul et al., 1997] Abiteboul, S., Quass, D., McHugh, J., Widom, J., and Wiener, J. L. (1997). The Lorel query language for semistructured data. *International Journal on Digital Libraries*, 1(1):68–88.

[Aidman and Egan, 1998] Aidman, E. and Egan, G. (1998). Academic assessment through computerized concept mapping: validating a method of implicit map reconstruction. *International Journal of Instructional Media*, 25(3):277–294.

[Aiken and Sleeman, 2003] Aiken, A. and Sleeman, D. (2003). Refiner++: A knowledge acquisition and refinement tool. In Sleeman, D. and Gil, Y., editors, *KCAP Workshop on Capturing knowledge from domain experts: Progress & Prospects*.

[Altman et al., 1999] Altman, R. B., Bada, M., Chai, X. J., Carillo, M. W., Chen, R. O., and Abernethy, N. F. (1999). RiboWeb: An ontology-based system for collaborative molecular biology. *IEEE Intelligent Systems*, 14(5):68–76.

[Anick and Tipirneni, 1999] Anick, P. G. and Tipirneni, S. (1999). The paraphrase search assistant:

terminological feedback for iterative information seeking. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 153–159. ACM Press.

[Anick and Vaithyanathan, 1997] Anick, P. G. and Vaithyanathan, S. (1997). Exploiting clustering and phrases for context-based information retrieval. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 314–323. ACM Press.

[Anjewierden et al., 1992] Anjewierden, A., Wielemaker, J., and Toussaint, C. (1992). Shelley - computer aided knowledge engineering. knowledge acquisition. *Knowledge Acquisition*, 4(1).

[Armstrong et al., 1995] Armstrong, R., Freitag, D., Joachims, T., and Mitchell, T. (1995). WebWatcher: A learning apprentice for the World Wide Web. In *AAAI Spring Symposium on Information Gathering*, pages 6–12.

[Arpírez et al., 2001] Arpírez, J. C., Corcho, O., Fernández-López, M., and Gómez-Pérez, A. (2001). WebODE: a scalable workbench for ontological engineering. In *Proceedings of the international conference on Knowledge capture*, pages 6–13. ACM Press.

[Ashish and Knoblock, 1997] Ashish, N. and Knoblock, C. A. (1997). Semi-automatic wrapper generation for internet information sources. In *Conference on Cooperative Information Systems*, pages 160–169.

[Asnicar and Tasso, 1997] Asnicar, F. and Tasso, C. (1997). ifWeb: a prototype of user models based intelligent agent for document filtering and navigation in the World Wide Web. In *Sixth International Conference on User Modeling*, Chia Laguna, Sardinia, Italy.

[Ausubel, 1963] Ausubel, D. P. (1963). *The psychology of meaningful verbal learning*. Grune and Stratton.

[Ausubel, 1968] Ausubel, D. P. (1968). *Educational psychology: a cognitive view*. Holt, Rinehart and Winston.

[Babaian et al., 2002] Babaian, T., Grosz, B. J., and Shieber, S. M. (2002). A writer's collaborative assistant. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 7–14. ACM Press.

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

[Baker and McCallum, 1998] Baker, L. D. and McCallum, A. K. (1998). Distributional clustering of words for text classification. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 96–103. ACM Press.

[Balabanović and Shoham, 1997] Balabanović, M. and Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72.

[Balabanovic et al., 1995] Balabanovic, M., Shoham, Y., and Yun, Y. (1995). An adaptive agent for automated Web browsing. *Journal of Visual Communication and Image Representation*, 6(4).

[Baldonado and Winograd, 1997] Baldonado, M. Q. W. and Winograd, T. (1997). SenseMaker: an information-exploration interface supporting the contextual evolution of a user's interests. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 11–18. ACM Press.

[Barker et al., 2001] Barker, K., Porter, B., and Clark, P. (2001). A library of generic concepts for composing knowledge bases. In *Proceedings of the international conference on Knowledge capture*, pages 14–21. ACM Press.

[Bauer and Leake, 2001] Bauer, T. and Leake, D. (2001). A research agent architecture for real time data collection and analysis. In *Proceedings of the Workshop on Infrastructure for Agents, MAS, and Scalable MAS*.

[Bechhofer et al., 2001] Bechhofer, S., Horrocks, I., Goble, C., and Stevens, R. (2001). OilEd: A reason-able ontology editor for the semantic Web. *Lecture Notes in Computer Science*, 2174:396–408.

[Belkin, 2000] Belkin, N. J. (2000). Helping people find what they don't know. *Commun. ACM*, 43(8):58–61.

[Bennett, 1985] Bennett, J. (1985). ROGET: a knowledge-based system for acquiring the conceptual structure of a diagnostic expert system. *Journal of Automated Reasoning*, 1:49–74.

[Berge, 1973] Berge, C. (1973). *Graphs and Hypergraphs*. North Holland.

[Berkhin, 2002] Berkhin, P. (2002). Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA.

[Berners-Lee, 1998] Berners-Lee, T. (1998). Semantic Web road map. Technical report, W3C Design Issues.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American*.

[Bernstein et al., 1991] Bernstein, M., Bolter, J. D., Joyce, M., and Mylonas, E. (1991). Architectures for volatile hypertext. In *Proceedings of the third annual ACM conference on Hypertext*, pages 243–260. ACM Press.

[Bharat et al., 1998] Bharat, K., Broder, A., Henzinger, M. R., Kumar, P., and Venkatasubramanian, S. (1998). The connectivity server: Fast access to linkage information on the Web. In *Proceedings of the 7th International World Wide Web Conference (WWW-7)*, pages 469–477, Brisbane, Australia.

[Bharat and Henzinger, 1998] Bharat, K. and Henzinger, M. R. (1998). Improved algorithms for topic distillation in a hyperlinked environment. In *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 104–111, Melbourne, AU.

[Billsus and Pazzani, 1999] Billsus, D. and Pazzani, M. J. (1999). A hybrid user model for news classification. In *In Kay J. (ed.), UM99 User Modeling - Proceedings of the Seventh International Conference*, pages 99–108. Springer-Verlag.

[Blythe et al., 2001] Blythe, J., Kim, J., Ramachandran, S., and Gil, Y. (2001). An integrated environment for knowledge acquisition. In *Proceedings of the 6th international conference on Intelligent user interfaces*, pages 13–20. ACM Press.

[Boldi and Vigna, 2003] Boldi, P. and Vigna, S. (2003). The WebGraph framework i: Compression techniques.

[Bollacker et al., 1998] Bollacker, K., Lawrence, S., and Giles, C. L. (1998). CiteSeer: An autonomous Web agent for automatic retrieval and identification of interesting publications. In Sycara, K. P. and Wooldridge, M., editors, *Proceedings of the Second International Conference on Autonomous Agents*, pages 116–123, New York. ACM Press.

[Boose and Bradshaw, 1987] Boose, J. and Bradshaw, J. (1987). Expertise transfer and complex problems: Using AQUINAS as a knowledge acquisition workbench for knowledge-based systems. *International Journal of Man-Machine Studies*, 26:3–28.

[Boose, 1985] Boose, J. H. (1985). A knowledge acquisition program for expert systems based on personal construct psychology. *International Journal of Man-Machine Studies*, 20:21–43.

[Borges and Levene, 1999] Borges, J. and Levene, M. (1999). Data mining of user navigation patterns. In *WEBKDD*, pages 92–111.

[Botafogo, 1993] Botafogo, R. A. (1993). Cluster analysis for hypertext systems. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 116–125. ACM Press.

[Botafogo et al., 1992] Botafogo, R. A., Rivlin, E., and Shneiderman, B. (1992). Structural analysis of hypertexts: identifying hierarchies and useful metrics. *ACM Transactions on Information Systems (TOIS)*, 10(2):142–180.

[Botafogo and Shneiderman, 1991] Botafogo, R. A. and Shneiderman, B. (1991). Identifying aggregates in hypertext structures. In *Proceedings of the third annual ACM conference on Hypertext*, pages 63–74. ACM Press.

[Brachman and Schmolze, 1985] Brachman, R. and Schmolze, J. (1985). An overview of the klone knowledge representation system. *Cognitive Science*, 9(2):171–216.

[Bradshaw, 1997] Bradshaw, J. M. (1997). An introduction to software agents. In Bradshaw, J. M., editor, *Software Agents*, pages 3–46. AAAI Press / The MIT Press.

[Bradshaw et al., 2000] Bradshaw, S., Scheinkman, A., and Hammond, K. (2000). Guiding people to information: providing an interface to a digital library using reference as a basis for indexing. In *Proceedings of the 5th international conference on Intelligent user interfaces*, pages 37–43. ACM Press.

[Briggs et al., 2004] Briggs, G., Shamma, D., Cañas, Carff, R., Scargle, J., and Novak, J. D. (2004). Concept maps applied to Mars exploration public outreach. In Cañas, A. J., Novak, J. D., and González, F., editors, *Concept Maps: Theory, Methodology, Technology. Proceedings of the First International Conference on Concept Mapping*, pages 125–133.

[Brin and Page, 1998] Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.

[Budzik and Hammond, 1999] Budzik, J. and Hammond, K. (1999). Watson: Anticipating and contextualizing information needs. In *62nd Annual Meeting of the American Society for Information Science*, Medford, NJ.

[Budzik and Hammond, 2000] Budzik, J. and Hammond, K. J. (2000). User interactions with everyday applications as context for just-in-time information access. In *Proceedings of the 2000 International Conference on Intelligent User Interfaces*, New Orleans, Louisiana. ACM Press.

[Budzik et al., 2001] Budzik, J., Hammond, K. J., and Birnbaum, L. (2001). Information access in context. *Knowledge based systems*, 14(1–2):37–53.

[Budzik et al., 2000] Budzik, J., Hammond, K. J., Birnbaum, L., and Krema, M. (2000). Beyond similarity. In *Proceedings of the 2000 Workshop on Artificial Intelligence and Web Search*. AAAI Press.

[Bustamante et al., 1996] Bustamante, F. R., , and León, F. S. (1996). GramCheck: A grammar and style checker.

[Cañas et al., 2002] Cañas, A., Carvalho, M., and Arguedas, M. (2002). Mining the Web to suggest concepts during concept mapping: Preliminary results. In *XIII Simpsio Brasileiro de Informtica na Educao*, SBIE UNISINOS.

[Cañas et al., 2001] Cañas, A., Leake, D., and Maguitman, A. (2001). Combining concept mapping with CBR: Experience-based support for knowledge modeling. In *Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference*, pages 286–290. AAAI Press.

[Cañas et al., 1998] Cañas, A. J., Coffey, J., Reichherzer, T., Hill, G., Suri, N., Carff, R., Mitrovich, T., and Eberle, D. (1998). El-Tech: a performance support system with embedded training for electronics technicians. In *Proceedings of the Eleventh International Florida Artificial Intelligence Research Society Conference*, pages 79–83. AAAI Press.

[Cañas et al., 2004] Cañas, A. J., Hill, G., Carff, R., Suri, N., Lott, J., Eskridge, T., Gómez, G., Arroyo, M., and Carvajal, R. (2004). CmapTools: A knowledge modeling and sharing environment. In Cañas, A. J., Novak, J. D., and González, F., editors, *Concept Maps: Theory, Methodology, Technology. Proceedings of the First International Conference on Concept Mapping*.

[Cañas et al., 2003] Cañas, A. J., Valerio, A., Lalinde-Pulido, J., Carvalho, M., and Arguedas, M. (2003). Using WordNet for word sense disambiguation to support concept map construction. In *Proceedings of SPIRE 2003. 10th International Symposium on String Processing and Information Retrieval*, Manaus, Brazil. Springer-Verlag.

[Carvalho et al., 2001] Carvalho, M., Hewett, R., and Cañas, A. (2001). Enhancing Web searches from concept map-based knowledge models. In *Proceedings of the SCI Conference*, Orlando, Florida.

[Chakrabarti et al., 1998a] Chakrabarti, S., Dom, B., Gibson, D., Kleinberg, J., Raghavan, P., and Rajagopalan, S. (1998a). Automatic resource list compilation by analyzing hyperlink structure and associated text. In *Proceedings of the 7th International World Wide Web Conference*. 1998a.

[Chakrabarti et al., 1998b] Chakrabarti, S., Dom, B., and Indyk, P. (1998b). Enhanced hypertext categorization using hyperlinks. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 307–318. ACM Press. 1998b.

[Chakrabarti et al., 1999a] Chakrabarti, S., Dom, B. E., Gibson, D., Kleinberg, J. M., Kumar, S. R., Raghavan, P., Rajagopalan, S., and Tomkins, A. (1999a). Hypersearching the Web. *Scientific American*. 1999b.

[Chakrabarti et al., 1999b] Chakrabarti, S., Dom, B. E., Kumar, S. R., Raghavan, P., Rajagopalan, S., Tomkins, A., Gibson, D., and Kleinberg, J. (1999b). Mining the Web's link structure. *Computer*, 32(8):60–67.

[Chakrabarti et al., 2002] Chakrabarti, S., Joshi, M. M., Punera, K., and Pennock, D. M. (2002). The structure of broad topics on the Web. In *Proceedings of the eleventh international conference on World Wide Web*, pages 251–262. ACM Press. 1999c.

[Chakrabarti et al., 1999c] Chakrabarti, S., van den Berg, M., and Dom, B. (1999c). Focused crawling: a new approach to topic-specific Web resource discovery. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1623–1640. 1999a.

[Chandrasekaran, 1983] Chandrasekaran, B. (1983). Toward a taxonomy of problem-solving types. *AI Magazine*, 4(4):9–17.

[Chandrasekaran, 1986] Chandrasekaran, B. (1986). Generic tasks in knowledge-based reasoning: High level building blocks for expert system design. *IEEE Expert*, 1(3):23–30.

[Chen, 1997] Chen, C. (1997). Structuring and visualising the WWW by generalised similarity analysis. In *Proceedings of the eighth ACM conference on Hypertext*, pages 177–186. ACM Press.

[Chen and Dhar, 1990] Chen, H. and Dhar, V. (1990). Online query refinement on information retrieval systems: a process model of searcher/system interactions. In *Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 115–133. ACM Press.

[Chen and Dumais, 2000] Chen, H. and Dumais, S. (2000). Bringing order to the Web: Automatically categorizing search results. In *Proceedings of CHI'00, Human Factors in Computing Systems*.

[Chu and Rosenthal, 1996] Chu, H. and Rosenthal, M. (1996). Search engines for the World Wide Web: A comparative study and evaluation methodology. In *Annual Conference Proceedings (ASIS'96)*, pages 127–135.

[Chui, 2002] Chui, M. (2002). *I Still Haven't Found What I'm Looking For: Web Searching as Query Refinement*. PhD thesis, Indiana University.

[Church and Rau, 1995] Church, K. and Rau, L. (1995). Commercial applications of natural language processing. *Communications of the ACM*, 38(11).

[Clancey, 1981] Clancey, W. (1981). NEOMYCIN: reconfiguring a rule-based expert system for application to teaching. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence (IJCAI81)*, Vancouver, B.C.

[Clancey, 1983] Clancey, W. (1983). The advantages of abstract control knowledge in expert system design. In *Proceedings of the National Conference on Artificial Intelligence (AAAI83)*.

[Clancey, 1985] Clancey, W. (1985). Heuristic classification. *Artificial Intelligence*, 27:289–350.

[Clancey, 1984] Clancey, W. J. (1984). Classification problem solving. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-84)*.

[Clark et al., 2001] Clark, P., Thompson, J., Barker, K., Porter, B., Chaudhri, V., Rodriguez, A., Thomere, J., Mishra, S., Gil, Y., Hayes, P., and Reichherzer, T. (2001). Knowledge entry as the graphical assembly of components. In *Proceedings of the international conference on Knowledge capture*, pages 22–29. ACM Press.

[Coffey et al., 2002] Coffey, J. W., Hoffman, R., Cañas, A. J., and Ford, K. M. (2002). A concept map-based knowledge modeling approach to expert knowledge sharing. In *Proceedings of the IASTED International Conference on Information and Knowledge Sharing (IKS-2002)*, Virgin Islands.

[Collins and Quillian, 1969] Collins, A. M. and Quillian, M. R. (1969). Retrieval time from semantic memory. *Journal of verbal learning and verbal behavior*, 8:240–248.

[Cooley, 2000] Cooley, R. (2000). *Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data*. Ph.d. thesis, University of Minnesota.

[Cormen et al., 1990] Cormen, T. H., Leiserson, C. E., and L., R. L. R. R. (1990). *Introduction to Algorithms*. MIT Press.

[Craik, 1943] Craik, K. (1943). *The Nature of Explanation*. Cambridge University Press.

[Craven et al., 2000] Craven, M., DiPasquo, D., Freitag, D., McCallum, A. K., Mitchell, T. M., Nigam, K., and Slattery, S. (2000). Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence*, 118(1/2):69–113.

[Croft and Turtle, 1989] Croft, W. B. and Turtle, H. (1989). A retrieval model incorporating hypertext links. In *Proceedings of the second annual ACM conference on Hypertext*. ACM Press.

[Crouch, 1988] Crouch, C. J. (1988). A cluster-based approach to thesaurus construction. In *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 309–320. ACM Press.

[Cutting et al., 1992] Cutting, D. R., Pedersen, J. O., Karger, D., and Tukey, J. W. (1992). Scatter/Gather: A cluster-based approach to browsing large document collections. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 318–329.

[Cypher, 1991] Cypher, A. (1991). Eager: Programming repetitive tasks by example. In *Proceedings of CHI'91*, pages 33–39.

[Davies et al., 2003] Davies, J., Duke, A., and Sure, Y. (2003). OntoShare: a knowledge management environment for virtual communities of practice. In *Proceedings of the international conference on Knowledge capture*, pages 20–27. ACM Press.

[Davis, 1979] Davis, R. (1979). Interactive transfer of expertise: Acquisition of new inference rules. *Artificial Intelligence*, 12:121–157.

[Davis, 1982] Davis, R. (1982). TEIRESIAS: Applications of meta-level knowledge. In Davis, R. and Lenat, D. B., editors, *Knowledge-based Systems in Artificial Intelligence*, pages 227–490. McGrawHill, New York.

[de Hoog et al., 1993] de Hoog, R., Martil, R., Wielinga, B., Taylor, R., Bright, C., and Velde, W. (1993). The Common KADS model set. Technical report, University of Amsterdam, Lloyd's Register.

[Dean and Henzinger, 1999] Dean, J. and Henzinger, M. R. (1999). Finding related pages in the World Wide Web. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1467–1479.

[Deerwester et al., 1990] Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

[Defays, 1977] Defays, D. (1977). An efficient algorithm for a complete link method. *The Computer Journal*, 20(4):364–366.

[Dempster et al., 1977] Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

[DeVaul and Pentland, 2002] DeVaul, R. W. and Pentland, A. (2002). Toward the zero attention interface: Wearable subliminal cuing for short term memory support memory. In *Proceedings of ISWC*, pages 141–142.

[Dey and Abowd, 2000] Dey, A. K. and Abowd, G. D. (2000). CybreMinder: A context-aware system for supporting reminders. In *International Symposium on Handheld and Ubiquitous Computing, HUC 2000*, pages 172–186.

[Dhillon, 2001] Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274. ACM Press.

[Dill et al., 2001] Dill, S., Kumar, S. R., McCurley, K. S., Rajagopalan, S., Sivakumar, D., and Tomkins, A. (2001). Self-similarity in the Web. In *The VLDB journal*, pages 69–78.

[Doorenbos et al., 1997] Doorenbos, R. B., Etzioni, O., and Weld, D. S. (1997). A scalable comparison-shopping agent for the World-Wide Web. In Johnson, W. L. and Hayes-Roth, B., editors, *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, pages 39–48, Marina del Rey, CA, USA. ACM Press.

[Dubes, 1987] Dubes, R. C. (1987). How many clusters are best?—an experiment. *Pattern Recogn.*, 20(6):645–663.

[Duda et al., 1979] Duda, R. O., Gasching, J. G., and Hart, P. E. (1979). Model design in the prospector consultant system for mineral exploration. In Michie, D., editor, *Expert Systems in the Micro-Electronic Age*, pages 153–167. Edinburgh University Press.

[Duda and Hart, 1973] Duda, R. O. and Hart, P. E. (1973). *Pattern classification and scene analysis*. New York, Wiley. B-Swain Q327 .D847.

[Eirinaki and Vazirgiannis, 2003] Eirinaki, M. and Vazirgiannis, M. (2003). Web mining for Web personalization. *ACM Transactions on Internet Technology (TOIT)*, 3(1):1–27.

[Engelbart, 1962] Engelbart, D. (1962). Augmenting human intellect: A conceptual framework. Summary report, Stanford Research Institute, on Contract AF 49(638)-1024.

[Eshelman, 1988] Eshelman, L. (1988). MOLE: A knowledge-acquisition tool for cover-and-differentiate systems. In Marcus, S., editor, *Automating Knowledge Acquisition for Expert Systems*, pages 37–80. Kluwer Academic Publishers, The Netherlands.

[Esposito, 1990] Esposito, C. (1990). A graph-theoretic approach to concept clustering. In *Pathfinder associative networks: studies in knowledge organization*, pages 89–99. Ablex Publishing Corp.

[Etzioni, 1996a] Etzioni, O. (1996a). Moving up the information food chain: Deploying Softbots on the World Wide Web. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, pages 1322–1326, Menlo Park. AAAI Press / MIT Press.

[Etzioni, 1996b] Etzioni, O. (1996b). The World-Wide Web: quagmire or gold mine? *Commun. ACM*, 39(11):65–68.

[Etzioni and Weld, 1994] Etzioni, O. and Weld, D. (1994). A Softbot-based interface to the Internet. *Communications of the ACM*, 37(7):72–76.

[Everitt, 1980] Everitt, B. S. (1980). *Cluster Analysis*. Halsted Press, New York.

[Farquhar et al., 1997] Farquhar, A., Fikes, R., and Rice, J. (1997). The Ontolingua server: A tool for collaborative ontology construction. *International Journal of Human-Computer Studies*, 46(6):707–727.

[Fellbaum, 1998] Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. MIT Press.

[Fischer et al., 1993] Fischer, G., Nakakoji, K., Ostwald, J., Stahl, G., and Sumner, T. (1993). Embedding computer-based critics in the contexts of design. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 157–164. ACM Press.

[Forbus and Usher, 2002] Forbus, K. D. and Usher, J. (2002). Sketching for knowledge capture: A progress report. In *Proceedings of the 7th International Conference on Intelligent User Interfaces*, pages 71–77. ACM Press.

[Ford et al., 1996] Ford, K. M., Coffey, J. W., Cañas, A. J., Andrews, E. J., and Turner, C. W. (1996). Diagnosis and explanation by a nuclear cardiology expert system. *International Journal of Expert Systems*, 9:499–506.

[Frei and Stieger, 1992] Frei, H. P. and Stieger, D. (1992). Making use of hypertext links when retrieving information. In *Proceedings of the ACM conference on Hypertext*, pages 102–111. ACM Press.

[Fu et al., 2000] Fu, X., Budzik, J., and Hammond, K. J. (2000). Mining navigation history for recommendation. In *Intelligent User Interfaces*, pages 106–112.

[Gädenfors, 2000] Gädenfors, P. (2000). *Conceptual Spaces: The Geometry of Thought*. Bradford Books MIT Press.

[Gaines and Shaw, 1993] Gaines, B. and Shaw, M. (1993). Knowledge acquisition tools based on personal construct psychology. *The Knowledge Engineering Review*, 8(1).

[Ganter and Wille, 1999] Ganter, B. and Wille, R. (1999). *Formal Concept Analysis: Mathematical Foundations.* Springer, Berlin - Heidelberg - New York.

[Gibson et al., 1998] Gibson, D., Kleinberg, J. M., and Raghavan, P. (1998). Inferring Web communities from link topology. In *UK Conference on Hypertext*, pages 225–234.

[Gil, 1994] Gil, Y. (1994). Knowledge refinement in a reflective architecture. In *Twelfth National Conference on Artificial Intelligence*. AAAI Press.

[Göker and Thompson, 2000] Göker, M. and Thompson, C. (2000). Personalized conversational case-based recommendation. In E. Blanzieri, L. P., editor, *Advances in Case-Based Reasoning. Proceedings of the 5 th European Workshop on Case-Based Reasoning, (EWCBR '2000) LNAI 1898*. Springer.

[Goldberg et al., 1992] Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70.

[Goldsmith and Davenport, 1990] Goldsmith, T. E. and Davenport, D. M. (1990). Assessing structural similarity of graphs. In *Pathfinder associative networks: studies in knowledge organization*, pages 75–87. Ablex Publishing Corp.

[Goldsmith et al., 1991] Goldsmith, T. E., Johnson, P. J., and Acton, W. H. (1991). Assessing structural knowledge. *Journal of Educational Psychology*, 83:88–96.

[Greenberg, 1998] Greenberg, J. (1998). *An Examination of the Impact of Lexical-Semantic Relationships on Retrieval Effectiveness During the Query Expansion Process*. PhD thesis, University of Pittsburgh.

[Gruber, 1992] Gruber, T. R. (1992). Ontolingua: A mechanism to support portable ontologies. Ksl report ksl-91-66, Stanford University.

[Gruber, 1993] Gruber, T. R. (1993). Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In Guarino, N. and Poli, R., editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Deventer, The Netherlands. Kluwer Academic Publishers.

[Guillaume and Latapy, 2002] Guillaume, J.-L. and Latapy, M. (2002). The Web graph: an overview. In *AlgoTel'2002*.

[Hara and Kasahara, 1990] Hara, Y. and Kasahara, Y. (1990). A set-to-set linking strategy for hypertext systems. In *Proceedings of the conference on Office information systems*, pages 131–135. ACM Press.

[Hara et al., 1991] Hara, Y., Keller, A. M., and Wiederhold, G. (1991). Implementing hypertext database relationships through aggregations and exception. In *Proceedings of the third annual ACM conference on Hypertext*, pages 75–90. ACM Press.

[Hartigan and Wong, 1979] Hartigan, J. A. and Wong, M. A. (1979). A K-means clustering algorithm. *Applied Statistics*, 28.

[Hasling et al., 1984] Hasling, D. W., Clancey, W., and Rennels, G. (1984). Strategic explanations for a diagnostic consultation system. *International Journal of Man-Machine Studies*, 20(1).

[Hayes-Roth et al., 1983] Hayes-Roth, F., Waterman, and Lenat, D. (1983). *Building Expert Systems*. Addison-Wesley.

[Hearst and Pedersen, 1996] Hearst, M. A. and Pedersen, J. O. (1996). Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, pages 76–84, Zürich, CH.

[Hoffman et al., 2001] Hoffman, R. R., Coffey, J. W., Ford, K. M., , and Carnot, M. J. (2001). Storm-LK: A human-centered knowledge model for weather forecasting. In *Proceedings of the 45th Annual Meeting of the Human Factors and Ergonomics Society*, Minneapolis, MN, USA.

[Horvitz, 1999] Horvitz, E. (1999). Principles of mixed-initiative user interfaces. In *CHI*, pages 159–166.

[Horvitz et al., 1998] Horvitz, E., Breese, J., Heckerman, D., Hovel, D., and Rommelse, K. (1998). The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users.

In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 256–265, Madison, WI.

[Iwanska and Shapiro, 2000] Iwanska, L. M. and Shapiro, S. C. (2000). *Natural language processing and knowledge representation: language for knowledge and knowledge for language*. AAAI Press.

[Jaczynski and Trousse, 1997] Jaczynski, M. and Trousse, B. (1997). BROADWAY: A World Wide Web browsing advisor reusing past navigations from a group of users. In *Proceedings of the Third UK Case-Based Reasoning Workshop (UKCBR3)*, Manchester, UK.

[Jain et al., 1999] Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323.

[Jones, 1972] Jones, S. K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21.

[Kahle and Gilliat, 1998] Kahle, B. and Gilliat, B. (1998). Alexa—navigate the Web smarter, faster, easier. Technical report, alexa internet, Presidio of San Francisco, CA.

[Kaski et al., 1998] Kaski, S., Lagus, K., and Kohonen, T. (1998). Websom - self-organizing maps of document collections. *Neurocomputing*, 21:101–117.

[Kaufman and Rousseeuw, 1989] Kaufman, L. and Rousseeuw, P. J. (1989). *Finding groups in data: an introduction to cluster analysis*. New York : Wiley.

[Kautz et al., 1997] Kautz, H., Selman, B., and Shah, M. (1997). Referral Web: combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65.

[Kelly, 1955] Kelly, G. (1955). *The Psychology of Personal Constructs*. Norton.

[Kingston, 1995] Kingston, J. K. (1995). Applying KADS to KADS: knowledge based guidance for knowledge engineering. *Expert Systems: The International Journal of Knowledge Engineer*, 12(1).

[Kira and Rendell, 1992] Kira, K. and Rendell, L. (1992). The feature selection problem: Traditional methods and a new algorithm. In *Tenth National Conference Conference on Artificial Intelligence (AAAI-92)*, pages 129–134. MIT Press.

[Klein, 2001] Klein, M. (2001). XML, RDF, and Relatives. *IEEE Intelligent Systems*, 16(2):26–28.

[Kleinberg, 1999] Kleinberg, J. (1999). Authoritative sources in a hyperlinked environment. *journal of the ACM*, 46(5):604–632.

[Kobayashi and Takeda, 2000] Kobayashi, M. and Takeda, K. (2000). Information retrieval on the Web. *ACM Comput. Surv.*, 32(2):144–173.

[Kolodner, 1993] Kolodner, J. (1993). *Case-Based Reasoning*. Morgan Kaufmann, San Mateo, CA.

[Konstan et al., 1997] Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., and Riedl, J. (1997). GroupLens: applying collaborative filtering to Usenet news. *Communications of the ACM*, 40(3):77–87.

[Kosala and Blockeel, 2000] Kosala and Blockeel (2000). Web mining research: A survey. *SIGKDD: SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery & Data Mining, ACM*, 2.

[Kukich, 1992] Kukich, K. (1992). Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24(4):377–439.

[Kushmerick et al., 1997] Kushmerick, N., Weld, D. S., and Doorenbos, R. B. (1997). Wrapper induction for information extraction. In *Intl. Joint Conference on Artificial Intelligence (IJCAI)*, pages 729–737.

[Kwok, 1985] Kwok, K. L. (1985). A probabilistic theory of indexing and similarity measure based on cited and citing documents. *Journal of the American Society for Information Science*, 36(5):342–351.

[Lamming and Flynn, 1994] Lamming, M. and Flynn, M. (1994). Forget-me-not: intimate computing in support of human memory. In *Proceedings FRIEND21 Symposium on Next Generation Human Interfaces*, Tokyo Japan.

[Langley et al., 1999] Langley, P., Thompson, C. A., Elio, R., and Haddadi, A. (1999). An adaptive conversational interface for destination advice. In *Cooperative Information Agents*, pages 347–364.

[Lau, 2001] Lau, T. (2001). *Programming by Demonstration: a Machine Learning Approach*. PhD thesis, University of Washington.

[Laurel, 1997] Laurel, B. (1997). Interface agents: Metaphors with character. In Bradshaw, J., editor, *Software Agents*. AAAI Press/MIT Press, Menlo Park, California.

[Leake, 1996] Leake, D. (1996). CBR in context: The present and future. In Leake, D., editor, *CaseBased Reasoning. Experiences, Lessons & Future Directions*, pages 3–30. AAAI Press.

[Leake et al., 2002] Leake, D., Maguitman, A., and Cañas, A. (2002). Assessing conceptual similarity to support concept mapping. In *Proceedings of FLAIRS-2002*, pages 168–172. AAAI Press.

[Leake et al., 2003a] Leake, D., Maguitman, A., and Reichherzer, T. (2003a). Topic extraction and extension to support concept mapping. In *Proceedings of FLAIRS-2003*, pages 325–329. AAAI Press.

[Leake et al., 2004a] Leake, D., Maguitman, A., and Reichherzer, T. (2004a). Understanding knowledge models: Modeling assessment of concept importance in concept maps. In *Proceedings of CogSci-2004*.

[Leake et al., 2003b] Leake, D., Maguitman, A., Reichherzer, T., Cañas, A., Carvalho, M., Arguedas, M., Brenes, S., and Eskridge, T. (2003b). Aiding knowledge capture by searching for extensions of knowledge models. In *Proceedings of KCAP-2003*. ACM Press.

[Leake et al., 2004b] Leake, D., Maguitman, A., Reichherzer, T., Cañas, A., Carvalho, M., Arguedas, M., and Eskridge, T. (2004b). "googling" from a concept map: Towards automatic concept-map-based query formation. In Cañas, A. J., Novak, J. D., and González, F., editors, *Concept Maps: Theory, Methodology, Technology. Proceedings of the First International Conference on Concept Mapping*.

[Leake et al., 2000] Leake, D. B., Bauer, T., Maguitman, A., and Wilson, D. C. (2000). Capture, storage and reuse of lessons about information resources: Supporting task-based information search. In *Proceedings of the AAAI-00 Workshop on Intelligent Lessons Learned Systems. Austin, Texas*, pages 33–37. AAAI Press.

[Lenat et al., 1990] Lenat, D. B., Guha, R. V., Pittman, K., Pratt, D., and Shepherd, M. (1990). CYC: Toward programs with common sense. *Communications of the ACM*, 33(8):30–49.

[Levesque and Brachman, 1987] Levesque, H. J. and Brachman, R. J. (1987). Expressiveness and tractability in knowledge representation and reasoning. *Computational Intelligence*, 3:78–93.

[Licklider, 1960] Licklider, J. C. R. (1960). Man-machine symbiosis. *IRE Transactions on Human Factors in Electronics*, HFE-1:4–11.

[Lieberman, 1987] Lieberman, H. (1987). An example-based environment for beginning programmers. *Artificial Intelligence and Education*, pages 135–151.

[Lieberman, 1995] Lieberman, H. (1995). Letizia: An agent that assists Web browsing. In Mellish, C. S., editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. IJCAI-95*, pages 924–929, Montreal, Quebec, Canada. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.

[Lieberman et al., 1999] Lieberman, H., Dyke, N. W. V., and Vivacqua, A. S. (1999). Let's browse: a collaborative Web browsing agent. In *Proceedings of the 1999 International Conference on Intelligent User Interfaces (IUI'99)*, pages 65–68, Los Angeles, CA, USA. ACM Press.

[Lieberman and Maulsby, 1996] Lieberman, H. and Maulsby, D. (1996). Instructible agents: Software that just keeps getting better. *IBM Systems Journal*, 35(3–4):539–556.

[Linton et al., 2000] Linton, F., Joy, D., Schaefer, H.-P., and Charron, A. (2000). OWL: a recommender system for organization-wide learning. *Educational Technology & Society*, 3(1).

[Maes, 1994] Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM*, 37(7):30–40.

[Maglio et al., 2000] Maglio, P. P., Barrett, R., Campbell, C. S., and Selker, T. (2000). SUITOR: an attentive information system. In *Proceedings of the 5th international conference on Intelligent user interfaces*, pages 169–176. ACM Press.

[Maglio and Campbell, 2003] Maglio, P. P. and Campbell, C. S. (2003). Attentive agents. *Communications of the ACM*, 46(3):47–51.

[Maguitman et al., 2004a] Maguitman, A., Leake, D., and Reichherzer, T. (2004a). Suggesting novel but related topics: Towards context-based support for knowledge model extension. (Submitted).

[Maguitman et al., 2004b] Maguitman, A., Leake, D., Reichherzer, T., and Menczer, F. (2004b). Dynamic extraction of topic descriptors and discriminators: Towards automatic context-based topic search. In *Proceedings of the Thirteenth Conference on Information and Knowledge Management (CIKM)*, Washington, DC. ACM Press.

[Marchiori, 1997] Marchiori, M. (1997). The quest for correct information on the Web: hyper search engines. In *Selected papers from the sixth international conference on World Wide Web*, pages 1225–1235. Elsevier Science Publishers Ltd.

[Marcus and McDermott, 1989] Marcus, S. and McDermott, J. P. (1989). SALT: A knowledge acquisition language for propose-and-revise systems. *Artificial Intelligence*, 39(1):1–37.

[Maulsby and Witten, 1989] Maulsby, D. L. and Witten, I. H. (1989). Inducing programs in a direct-manipulation environment. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 57–62. ACM Press.

[McCarthy, 1959] McCarthy, J. (1959). Programs with common sense. In *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, pages 75–91, London. Her Majesty's Stationary Office.

[McDonald et al., 1990] McDonald, J. E., Paap, K. R., and McDonald, D. R. (1990). Hypertext perspectives: using pathfinder to build hypertext systems. In *Pathfinder associative networks: studies in knowledge organization*, pages 197–211. Ablex Publishing Corp.

[Medin and Smith, 1984] Medin, D. and Smith, E. (1984). Concepts and concept formation. *Annual Psychological Review*, 35:113–138.

[Menczer et al., 2004] Menczer, F., Pant, G., and Srinivasan, P. (2004). Topic-driven crawlers: Machine learning issues. *ACM TOIT (To appear)*.

[Michael, 1994] Michael, R. S. (1994). *The Validity of Concept Maps for Assessing Cognitive Structure*. PhD thesis, Indiana University.

[Middleton et al., 2001] Middleton, S., DeRoure, D., and Shadbolt, N. (2001). Capturing knowledge of user preferences: Ontologies in recommender systems. In *Proceedings of the ACM K-CAP'01*, Victoria, Canada. ACM Press.

[Middleton, 2003] Middleton, S. E. (2003). *Capturing knowledge of user preferences with recommender systems*. PhD thesis, University of Southampton.

[Middleton et al., 2003] Middleton, S. E., Shadbolt, N. R., and De Roure, D. C. (2003). Capturing interest through inference and visualization: ontological user profiling in recommender systems. In *Proceedings of the international conference on Knowledge capture*, pages 62–69. ACM Press.

[Miller, 1983] Miller, P. (1983). Attending: Critiquing a physician's management plan. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI-5)*, pages 449–461.

[Minksy, 1975] Minksy, M. (1975). A framework for representing knowledge. In *The Psychology of Computer Vision*, pages 211–277. McGraw-Hill.

[Mitchell, 1982] Mitchell, T. M. (1982). Generalization as search. *Artiicial Intelligence*, 18(2):203–226.

[Mladenic, 1996] Mladenic, D. (1996). Personal WebWatcher: Design and implementation. Technical report ijs-dp-7472, School of Computer Science, Carnegie-Mellon University, Pittsburgh, USA.

[Mladenic, 1999] Mladenic, D. (1999). Text-learning and related intelligent agents: a survey. *IEEE Intelligent Systems*, 14(4):44–54.

[Modha and Spangler, 2000] Modha, D. S. and Spangler, W. S. (2000). Clustering hypertext with applications to Web searching. In *Proceedings of the eleventh ACM on Hypertext and hypermedia*, pages 143–152. ACM Press.

[Musen, 1989] Musen, M. A. (1989). An editor for the conceptual models of interactive knowledge-acquisition tools. *International Journal of Man-Machine Studies*, 31:673–698.

[Musen et al., 1988] Musen, M. A., Combs, D. M., Shortliffe, E. H., and Fagan, L. M. (1988). OPAL:

Toward the computer-aided design of oncology advice systems. In Miller, P. L., editor, *Selected Topics in Medical Artificial Intelligence*, pages 166–180. Springer-Verlag, New York, NY.

[Neches et al., 1985] Neches, R., Swartout, W., and Moore, J. (1985). Enhanced maintenance and explanation of expert systems through explicit models of their development. *IEEE Transactions on Software Engineering*, 11(11):1337–1351.

[Negroponte, 1997] Negroponte, N. (1997). Agents: From direct manipulation to delegation. In Bradshaw, J., editor, *Software Agents*. AAAI Press/MIT Press, Menlo Park, California.

[Newell, 1982] Newell, A. (1982). The knowledge level. *Artificial Intelligence*, 18:87–127.

[Nick et al., 1998] Nick, A., Koenemann, J., and Schal&#252;ck, E. (1998). ELFI: information brokering for the domain of research funding. *Comput. Netw. ISDN Syst.*, 30(16-18):1491–1500.

[Norman, 1991] Norman, D. A. (1991). Cognitive artifacts. In Carroll, J. M., editor, *Designing interaction*. Cambridege University Press, Cambridge, MA.

[Novak, 1977] Novak, J. (1977). *A Theory of Education*. Ithaca, Illinois, Cornell University Press.

[Novak, 2002] Novak, J. (2002). The theory underlying concept maps and how to construct them. Technical report, IHMC.

[Novak and Gowin, 1984] Novak, J. and Gowin, D. B. (1984). *Learning How to Learn*. Cambridge University Press.

[Noy et al., 2000] Noy, N., Fergerson, R., and Musen, M. (2000). The knowledge model of Protégé-2000: Combining interoperability and flexibility. In *Proceedings of EKAW 2000*.

[Oyama et al., 2001] Oyama, S., Kokubo, T., Ishida, T., Yamada, T., and Kitamura, Y. (2001). Keyword Spices: A new method for building domain-specific Web search engines. In *IJCAI*, pages 1457–1466.

[Paley et al., 1997] Paley, S. M., Lowrance, J. D., and Karp, P. D. (1997). A generic knowledge-base browser and editor. In *AAAI/IAAI*, pages 1045–1051.

[Pant et al., 2004] Pant, G., Srinivasan, P., and Menczer, F. (2004). Crawling the Web. In Levene, M. and Poulovassilis, A., editors, *Web Dynamics: Adapting to Change in Content, Size, Topology and Use*. Springer-Verlag.

[Pazzani et al., 1996] Pazzani, M. J., Muramatsu, J., and Billsus, D. (1996). Syskill & Webert: Identifying interesting Web sites. In *AAAI/IAAI, Vol. 1*, pages 54–61.

[Pirolli et al., 1996] Pirolli, P., Pitkow, J., and Rao, R. (1996). Silk from a sow's ear: Extracting usable structures from the Web. In *Proc. ACM Conf. Human Factors in Computing Systems, CHI*. ACM Press.

[Pitkow and Pirolli, 1997] Pitkow, J. and Pirolli, P. (1997). Life, death, and lawfulness on the electronic frontier. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 383–390. ACM Press.

[Puerta et al., 1992] Puerta, A. R., Egar, J., Tu, S., and Musen, M. (1992). A multiple-method shell for the automatic generation of knowledge acquisition tools. *Knowledge Acquisition*, 4:171–196.

[Quillian, 1968] Quillian, M. R. (1968). Semantic memory. In Minsky, M., editor, *Semantic Information Processing*, pages 216–270. MIT Press.

[Raghavan and Garcia-Molina, 2003] Raghavan, S. and Garcia-Molina, H. (2003). Representing Web graphs.

[Randall et al., 2001] Randall, K., Stata, R., Wickremesinghe, R., and Wiener, J. (2001). The link database: Fast access to graphs of the Web.

[Resnick et al., 1994] Resnick, P., Iacovou, N., Suchak, M., Bergstorm, P., and Riedl, J. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina. ACM.

[Resnick and Varian, 1997] Resnick, P. and Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, 40(3):56–58.

[Rhodes, 1997] Rhodes, B. (1997). The wearable remembrance agent: A system for augmented memory. In *Personal Technologies Journal Special Issue on Wearable Computing, Personal Technologies*, volume 1, pages 218–224.

[Rhodes and Starner, 1996] Rhodes, B. and Starner, T. (1996). The remembrance agent: A continuously running automated information retrieval system. In *The Proceedings of The First International Conference on The Practical Application of Intelligent Agents and Multi Agent Technology (PAAM '96)*, pages 487–495, London, UK.

[Rhodes, 2000] Rhodes, B. J. (2000). Margin notes: Building a contextually aware associative memory. In *The Proceedings of the International Conference on Intelligent User Interfaces (IUI '00)*.

[Rhodes et al., 1999] Rhodes, B. J., Minar, N., and Weaver, J. (1999). Wearable computing meets ubiquitous computing: Reaping the best of both worlds. In *The Proceedings of The Third International Symposium on Wearable Computers (ISWC '99)*, San Francisco, CA.

[Rijsbergen, 1979] Rijsbergen, C. J. V. (1979). *Information Retrieval*. Butterworth-Heinemann.

[Rosch et al., 1976] Rosch, E., Mervis, C., Gray, W., Johnson, D., and Boyes-Braem, P. (1976). Basic objects in natural categories. *Cognitive Psychoogy*, 8:382–439.

[Rucker and Polanco, 1997] Rucker, J. and Polanco, M. J. (1997). Siteseer: personalized navigation for the Web. *Communications of the ACM*, 40(3):73–76.

[Rumelhart and Norman, 1988] Rumelhart, D. E. and Norman, D. A. (1988). Representation in memory. In *Steven's handbook of experimental psychology*, volume 2, pages 511–587. Wiley.

[Ruspini, 1969] Ruspini, E. H. (1969). A new approach to clustering. *Information and Control*, 15(1):22–32.

[Salber et al., 1999] Salber, D., Dey, A. K., and Abowd, G. D. (1999). The context toolkit: Aiding the development of context-enabled applications. In *CHI*, pages 434–441.

[Salton, 1963] Salton, G. (1963). Associative document retrieval techniques using bibliographic information. *Journal of the ACM (JACM)*, 10(4):440–457.

[Salton, 1989] Salton, G. (1989). *Automatic Text Processing*. AddisonWesley.

[Salton and Buckley, 1988] Salton, G. and Buckley, C. (1988). Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523.

[Salton and Yang, 1973] Salton, G. and Yang, C. (1973). On the specification of term values in automatic indexing. *Journal of Documentation*, 29:351–372.

[Saracevic, 1995] Saracevic, T. (1995). Evaluation of evaluation in information retrieval. In *Proceedings of SIGIR*, pages 138–146.

[Schafer et al., 1999] Schafer, J. B., Konstan, J. A., and Riedi, J. (1999). Recommender systems in e-commerce. In *ACM Conference on Electronic Commerce*, pages 158–166.

[Schank and Abelson, 1977] Schank, R. and Abelson, R. (1977). *Scripts, plans, goals and understanding*. Lawrence Erlbaum Associates, Hillside, NJ.

[Schank et al., 1986] Schank, R. C., Collins, G. C., and Hunter, L. (1986). Transcending inductive category formation in learning. *Behavioral and Brain Sciences*, 9:639–686.

[Schlimmer and Hermens, 1993] Schlimmer, J. C. and Hermens, L. A. (1993). Software agents: Completing patterns and constructing user interfaces. *Journal of Artificial Intelligence Research*, 1:61–89.

[Schreiber and Wielinga, 1993] Schreiber, G. and Wielinga, B. (1993). KADS and conventional software engineering. In *KADS - A Principled Approach To Knowledge Based System Development*. Academic Press.

[Sebastiani, 2003] Sebastiani, F. (2003). Text categorization. In Zanasi, A., editor, *Text Mining and its Applications*. WIT Press, Southampton, UK. Invited chapter. Forthcoming.

[Selker, 1994] Selker, T. (1994). COACH: a teaching agent that learns. *Communications of the ACM*, 37(7):92–99.

[Shardanand and Maes, 1995] Shardanand, U. and Maes, P. (1995). Social information filtering: Algorithms for automating “word of mouth”. In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, volume 1, pages 210–217.

[Shneiderman, 1992] Shneiderman, B. (1992). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley.

[Shortliffe, 1976] Shortliffe, E. H. (1976). *Computer Based Medical Consultations: MYCIN*. American Elsevier, New York.

[Sibson, 1973] Sibson, R. (1973). SLINK: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34.

[Silverman, 1992] Silverman, B. G. (1992). Survey of expert critiquing systems: practical and theoretical frontiers. *Communications of the ACM*, 35(4):106–127.

[Skuce and Lethbridge, 1995] Skuce, D. and Lethbridge, T. (1995). CODE4: a unified system for managing conceptual knowledge. *International Journal of HumanComputer Studies*, 42(4):413–451.

[Smith, 1977] Smith, D. C. (1977). *Pygmalion: A Computer Program to Model and Stimulate Creative Thought*. Birkhauser, Basel.

[Smyth and McClave, 2001] Smyth, B. and McClave, P. (2001). Similarity vs. diversity. In *Proceedings of the 4th International Conference on Case-Based Reasoning. Vancouver, Canada*.

[Sowa, 1984] Sowa, J. F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley.

[Srinivasan et al., 2004] Srinivasan, P., Menczer, F., and Pant, G. (2004). A general evaluation framework for topical crawlers. *Information Retrieval (To appear)*.

[Srivastava et al., 2000] Srivastava, J., Cooley, R., Deshpande, M., and Tan, P.-N. (2000). Web usage mining: discovery and applications of usage patterns from Web data. *ACM SIGKDD Explorations Newsletter*, 1(2):12–23.

[Staab et al., 2000] Staab, S., Angele, J., Decker, S., Erdmann, M., Hotho, A., Maedche, A., Schnurr, H., Studer, R., and Sure, Y. (2000). AI for the Web—ontology-based community Web portals. In *AAAI-2000*, Menlo Park, USA. MIT Press.

[Stanfill and Waltz, 1986] Stanfill, C. and Waltz, D. (1986). Toward memory-based reasoning. *Communications of the ACM*, 29(12):1213–1228.

[Steels, 1990] Steels, L. (1990). Components of expertise. *AI Magazine*, 11(2):28–49.

[Suchman, 1987] Suchman, L. A. (1987). *Plans and Situated Actions*. Cambridge University Press, Cambridge.

[Suel and Yuan, 2001] Suel, T. and Yuan, J. (2001). Compressing the graph structure of the Web. In *Data Compression Conference*, pages 213–222.

[Sure et al., 2002] Sure, Y., Staab, S., and Angele, J. (2002). OntoEdit: Guiding ontology development by methodology and inferencing. In *Proceedings of the International Conference on Ontologies, Databases and Applications of Semantics (ODBASE 2002)*, University of California, Irvine, USA. Springer, LNCS.

[Sutherland, 1963] Sutherland, I. (1963). Sketchpad: a man–machine graphical communication system. In *Proceedings of the Spring Joint Computer Conference, IFIPS*, pages 329–346.

[Swartout et al., 1991] Swartout, W., Paris, C., and Moore, J. D. (1991). Design for explainable expert systems. *IEEE Expert*, 6(3):58–64.

[Terveen et al., 1997] Terveen, L., Hill, W., Amento, B., McDonald, D., and Creter, J. (1997). PHOAKS: a system for sharing recommendations. *Communications of the ACM*, 40(3):59–62.

[Terveen and Wroblewski, 1990] Terveen, L. G. and Wroblewski, D. A. (1990). A collaborative interface for editing large knowledge bases. In *Proceedings of AAAI*, pages 491–496, Boston.

[Thomas and Fischer, 1997] Thomas, C. G. and Fischer, G. (1997). Using agents to personalize the Web. In *Proceedings of the 2nd international conference on Intelligent user interfaces*, pages 53–60. ACM Press.

[Tulving, 1972] Tulving, E. (1972). Episodic and semantic memory. In *Organization of Memory*. Academic Press.

[Tversky, 1977] Tversky, A. (1977). Features of similarity. *Psychological Review*, 84(4):327–352.

[Van Dyke et al., 1999] Van Dyke, N. W., Lieberman, H., and Maes, P. (1999). Butterfly: a conversation-finding agent for internet relay chat. In *Proceedings of the 4th international conference on Intelligent user interfaces*, pages 39–41. ACM Press.

[Vélez et al., 1997] Vélez, B., Weiss, R., Sheldon, M. A., and Gifford, D. K. (1997). Fast and effective query refinement. In *Proceedings of the 20th ACM Conference on Research and Development in Information Retrieval (SIGIR 97). Philadelphia, PA*, pages 6–15.

[Vivacqua, 1999] Vivacqua, A. (1999). Agents for expertise location. In *Proceedings of AAAI Spring Symposium Workshop on Intelligent Agents in Cyberspace*, pages 9–13, Stanford, USA.

[Voss and Kreifelts, 1997] Voss, A. and Kreifelts, T. (1997). SOAP: social agents providing people with useful information. In *Proceedings of the international ACM SIGGROUP conference on Supporting group work : the integration challenge*, pages 291–298. ACM Press.

[Weiss et al., 1996] Weiss, R., Vélez, B., and Sheldon, M. A. (1996). HyPursuit: a hierarchical network search engine that exploits content-link hypertext clustering. In *Proceedings of the the seventh ACM conference on Hypertext*, pages 180–193. ACM Press.

[Weiss and Kulikowski, 1979] Weiss, S. and Kulikowski, C. A. (1979). EXPERT: A system for developing consultation models. In *Proceedings of the Sixth International Conference on Artificial Intelligence*, Tokyo.

[West et al., 2002] West, D., Park, J., Pomeroy, J., and Sandoval, J. (2002). Concept mapping assessment in medical education: a comparison of two scoring systems. *Medical Education*, 36(9):820–826.

[Wishard, 1998] Wishard, L. (1998). Precision among Internet search engines: An earth sciences case study. Issues in Science and Technology Librarianship.

[Wisniewski and Medin, 1991] Wisniewski, E. J. and Medin, D. L. (1991). Harpoons and long sticks: the interaction of theory and similarity in rule induction. In *Concept formation knowledge and experience in unsupervised learning*, pages 237–278. Morgan Kaufmann Publishers Inc.

[Zamir and Etzioni, 1999] Zamir, O. and Etzioni, O. (1999). Grouper: a dynamic clustering interface to Web search results. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1361–1374.

[Zhao and Karypis, 2001] Zhao, Y. and Karypis, G. (2001). Criterion functions for document clustering: Experiments and analysis. Technical report 01-40, University of Minnesota, Department of Computer Science.

[Ziegler and Fahnrich, 1988] Ziegler, J. E. and Fahnrich, K. P. (1988). Direct manipulation. In Helander, M., editor, *Handbook of Human-Computer Interaction*. Elsevier, Amsterdam.

Curriculum Vitae

Ana Gabriela Maguitman
Computer Science Department, Indiana University, Bloomington
Contact Information: (812)857-6787 / anmaguit@cs.indiana.edu

Education

2004. PhD (Computer Science) Indiana University, Bloomington, USA. Advisor: David B. Leake.

2004. Logic Certificate. Logic Program, Indiana University, Bloomington, USA.

1997. Magister en Ciencias de la Computación, Universidad Nacional del Sur, Bahía Blanca, Argentina. Advisor: Guillermo R. Simari.

1994. Licenciada en Ciencias de la Computación, Universidad Nacional del Sur, Bahía Blanca, Argentina.

Teaching Experience

August 1999-May 2000. Associate Instructor. Computer Science Department, Indiana University, Bloomington.

July 1995-July 1999. Teaching Assistant. Computer Science Department. Universidad Nacional del Sur, Argentina.

July 1991-June 1995. Junior Teaching Assistant. Computer Science Department. Universidad Nacional del Sur, Argentina.

Research Assistantships

August 2004-December 2004. School of Informatics, Indiana University, Bloomington. Supervisor: Filippo Menczer.

August 2000-July 2004. Computer Science Department, Indiana University, Bloomington. Supervisor: David B. Leake.

Participation in Projects

Intelligent Support for Knowledge Capture Refinement and Sharing. P.I:. David Leake and Alberto Cañas. Supported by NASA, Intelligent Systems program under award No NCC 2-1216. August 2000-May 2004.

Logic and Argumentation Systems. P.I.: Guillermo R. Simari. Supported by Universidad Nacional del Sur, Argentina. March 1997-August 1999.

Defeasible Reasoning Systems. P.I.: Guillermo R. Simari. Supported by Universidad Nacional del Sur, Argentina. July 1995-February 1997.

Visits to Universities

Computer Science Department. Stanford University - USA. January to March, 1999.

Computer Science Department. Washington University. Saint Louis-USA. September to November, 1997.

Professional Service

Student volunteer for NASSLLI 2003, AAAI 2000, and AAAI 2002.

Member of the organizing committee of FLAIRS 2004, FLAIRS 2005 and the Concept Mapping Conference 2004. Reviewer for IJAIT.

Honors

Fulbright Fellowship. August 1999.

Fellowship from CONICET (National Research Council, Argentina.) July, 1996.

Fellowship from Universidad Nacional del Sur (Initiation to Research.) May, 1995.

Articles in Journals and Conference Proceedings

Suggesting Novel but Related Topics: Towards Context-Based Support for Knowledge Model Extension. Ana Maguitman, David Leake, and Thomas Reichherzer. To appear in the International Conference of Intelligent User Interfaces (IUI'05). ACM Press. San Diego, January 2005

Combining Argumentation and Web Search Technology: Towards a Qualitative Approach for Ranking Results. Carlos I. Chesñevar and Ana G. Maguitman. To appear in the International Journal of Advanced Computational Intelligence & Intelligent Informatics. Vol. 9, No. 1. January 2005. Fuji Technology Press Ltd., Japan. Honorary Editor: Lofti Zadeh.

Dynamic Extraction of Topic Descriptors and Discriminators: Towards Automatic Context-Based Topic Search. Ana Maguitman, David Leake, Thomas Reichherzer and Filippo Menczer. Proceedings of the Thirteenth Conference on Information and Knowledge Management, CIKM. ACM Press. Washington, DC, November 2004.

"Googling" from a Concept Map: Towards Automatic Concept-Map-Based Query Formation. David Leake, Ana Maguitman, Thomas Reichherzer, Alberto Cañas, Marco Carvalho, Marco Arguedas, and Tom Eskridge. Concepts Maps: Theory, Methodology, Technology. Proceedings of the First Conference on Concept Mapping. A.J. Cañas, J.D. Novak, F. M. González Ed. Pamplona, Spain, September 2004.

Understanding Knowledge Models: Modeling Assessment of Concept Importance in Concept Maps. David Leake, Ana Maguitman, Thomas Reichherzer. Proceedings of CogSci 2004. Chicago, August, 2004.

An Argumentative Approach to Assessing Natural Language Usage based on the Web Corpus. Carlos I. Chesñevar and Ana G. Maguitman. Proceedings of the European Conference on Artificial Intelligence (ECAI'04). Valencia, Spain, August 2004.

A First Approach to Argument-Based Recommender Systems based on Defeasible Logic Programming. Carlos I. Chesẽvar, Ana G. Maguitman, G. Simari. Proceedings of the International Workshop on Non Monotonic Reasoning (NMR'04), Whistler, Canada, June 2004.

ArgueNet: An Argument-Based Recommender System for Solving Web Search Queries. Carlos I. Chesñevar, Ana G. Maguitman. Proceedings of the International IEEE Conference on Intelligent Systems (IS 2004). Varna, Bulgaria, June 2004.

Didactic Strategies for Promoting Significant Learning in Formal Languages and Automata Theory. Carlos I. Chesñevar, Maria P. González, Ana G. Maguitman. Proceedings of the International ACM-ITICSE Conference (Innovation and Technology in Computer Science Education). Leeds, UK, ACM Press, June, 2004.

Aiding Knowledge Capture by Searching for Extensions of Knowledge Models. David Leake, Ana Maguitman, Thomas Reichherzer, Alberto Cañas, Marco Carvalho, Marco Arguedas, Sofia Brenes, and Tom Eskridge. Proceedings of K-Cap-03. Sanibel Island, Florida. ACM Press, October, 2003.

Tecnología Informática en un curso de Lenguajes Formales y Teoría de Autómatas: Un Enfoque Constructivista. Carlos I. Chesñevar, Ana Maguitman, M. P. González, Laura Cobo. Proceedings of CACIC'03 (Congreso Argentino de Ciencias de la Computación), Buenos Aires, Argentina, October, 2003.

Topic Extraction and Extension to Support Concept Mapping. David Leake, Ana Maguitman, Thomas Reichherzer. Proceedings of FLAIRS-03. Saint Augustine, Florida. AAAI Press, May, 2003.

Assessing Conceptual Similarity to Support Concept Mapping. David Leake, Ana Maguitman, Alberto Cañas. Proceedings of FLAIRS-02. Pensacola, Florida. AAAI Press, May, 2002.

Context and Relevance: A Pragmatic Approach. Hamid R. Ekbia and Ana G. Maguitman. Modeling and Using Context. Third International and Interdisciplinary Conference, CONTEXT 2001. Dundee, UK. Lectures Notes in Artificial Intelligence. Springer, July 2001.

Combining Concept Mapping with CBR: Experience-Based Support for Knowledge Modeling. Alberto Cañas, David Leake, Ana Maguitman. Proceedings of FLAIRS-01. Key West, Florida. AAAI Press, May, 2001.

Logical Models of Argument. Carlos I. Chesñevar, Ana G. Maguitman and Ronald P. Loui. ACM Computing Surveys. 32 (4), pages 337-383. December, 2000.

Capture, Storage and Reuse of Lessons about Information Resources: Supporting Task-Based Information Search. David B. Leake, Travis Bauer, Ana Maguitman and David C. Wilson. Proceedings of the AAAI-00 Workshop on Intelligent Lessons Learned Systems. Austin, Texas. AAAI Press, July, 2000.

Belief Revision and Relevance. Marcelo Falappa, Ana G. Maguitman. Proceedings of WIIC'99 (Workshop de Investigadores en Ciencias de la Computacin). San Juan, May,1999.

Notions of Relevance for Modeling the Dynamics of Belief. Marcelo Falappa, Ana G. Maguitman. Proceedings of CACIC'99 (Congreso Argentino de Ciencias de la Computación).Tandil, October, 1999.

Rationality Postulates for Relevance Relations. Ana G. Maguitman and Guillermo R. Simari. Proceedings of the XVIII International Conference of the Chilean Computer Science Society. Edited by: IEEE Computer Sciences Series. November, 1998.

On the Use of Relevance to Characterize Explanations as Abductive Conclusions. Ana G. Maguitman y Guillermo R. Simari. Proceedings of the XVI International Conference of the Chilean Computer Science Society. Edited by: Marvin V. Zelwoitz and Pablo A. Straub, November, 1996.

Planificación de Procesos para la Interpretación de Programación en Lógica Concurrente en Prolog Secuencial. Ana G. Maguitman and Claudio Delrieux. Proceedings of CACIC’99 (Primer Congreso Argentino de Ciencias de La Computación.) Universidad Nacional del Sur, Baha Blanca, October, 1995.

Sobre el Uso de la Analogía como Medio de Inferencia. Ana G. Maguitman and Guillermo R. Simari. Proceedings of JAIIO’95 (24 Jornadas Argentinas de Informática e Investigación Operativa.) Ciudad Universitaria, Buenos Aires, August, 1995.