3-2008

Semantic Components: A Model for Enhancing Retrieval of Domain- Specific Information

Susan Loucette Price
Portland State University

ABSTRACT

An abstract of the dissertation of Susan Loucette Price for the Doctor of Philosophy in Computer Science presented March 7, 2008.

Title: Semantic Components: A Model for Enhancing Retrieval of Domain-Specific Information

Despite the success of general Internet search engines, information retrieval remains an incompletely solved problem. Our research focuses on supporting domain experts when they search domain-specific libraries to satisfy targeted information needs. The *semantic components model* introduces a schema specific to a particular document collection. A semantic component schema consists of a two-level hierarchy, *document classes* and *semantic components*. A document class represents a document grouping, such as topic type or document purpose. A semantic component is a characteristic type of information that occurs in a particular document class and represents an important aspect of the document's main topic. Semantic component indexing identifies the location and extent of semantic component instances within a document and can supplement traditional full text and keyword indexing techniques. Semantic component searching allows a user to refine a topical search by indicating a preference for documents containing specific semantic components or by indicating terms that should appear in specific semantic components.

We investigate four aspects of semantic components in this research. First, we describe lessons learned from using two methods for developing schemas in two domains. Second, we demonstrate use of semantic components to express domain-specific concepts and relationships by mapping a published taxonomy of questions asked by family practice physicians to the semantic component schemas for two document collections about medical care. Third, we report the results of a user study, showing that manual semantic component indexing is comparable to manual keyword indexing with respect to time and perceived difficulty and suggesting that semantic component indexing may be more accurate and consistent than manual keyword indexing. Fourth, we report the results of an interactive searching study, demonstrating the ability of semantic components to enhance search results compared to a baseline system without semantic components.

In addition, we contribute a formal description of the semantic components model, a prototype implementation of semantic component indexing software, and a prototype implementation adding semantic components to an existing commercial search engine. Finally, we analyze metrics for evaluating instances of semantic component indexing and keyword indexing and illustrate use of a session-based metric for evaluating multiple-query search sessions.

SEMANTIC COMPONENTS:

A MODEL FOR ENHANCING RETRIEVAL OF DOMAIN-SPECIFIC INFORMATION

by

SUSAN LOUCETTE PRICE

A dissertation submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY
in
COMPUTER SCIENCE

Portland State University

Acknowledgments

I would especially like to thank Lois Delcambre for her constant support and guidance throughout my PhD journey. She has been not only my advisor, but also a mentor, a role model, and a friend. I also thank the other members of my committee, David Maier, Len Shapiro, Warren Harrison, and Wayne Wakeland for their excellent suggestions and advice. In addition, I thank Marianne Lykke Nielsen, from the Royal School of Library and Information Science in Denmark. Our collaboration has been a source of both intellectual stimulation and personal friendship.

I am also grateful to the many other collaborators and participants in the research described in this study. Chapter One describes some key contributions to this work by Vibeke Luk, Peter Vedsted, Jens Rubak, Kalervo Jarvelin, and Timothy Tolle. In addition, I thank sundhed.dk for being our partner in several pieces of this work, Frans la Cour for providing us with a research license to use Ultraseek, and the many physicians, indexers, and others who participated in, or facilitated, the user studies.

My graduate experience has been far more than writing a dissertation. I am grateful to all the members of the datalab group, current and past, who have been friends and teachers.

Most importantly, I thank my husband Scott, who has been amazingly patient and supportive throughout my graduate years, and my parents, who always supported my education and my desire to take on new challenges.

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

Retrieving information from online resources is an increasingly prevalent task, supporting many work-related activities. So much information is available that even an expert cannot know, and remember, all the knowledge accumulated in his area of expertise. With billions of pages available on the Web as static HTML pages, and an untold number potentially available as dynamically generated web pages in response to database queries, the cliché *information overload* understates the problem. Furthermore, web technologies make it easy for anyone to make information available to others. Although the ease of providing information allows a variety of information types and opinions to be accessible, the diversity of information sources also creates new challenges. A searcher must sift through search results, deciding which documents are relevant to his need while evaluating the quality and authority of the information as well.

Despite the enormous success of general search engines, such as Google™, information retrieval (IR) remains an incompletely solved problem. Search queries are typically incomplete representations of the searcher's underlying information need and the matching algorithms used by search engines rely on incomplete representations of the semantic content of documents (what the document content means). As a result, search engines sometimes return many unwanted documents and fail to return documents containing the desired information at, or near, the top of the search results.

Choosing how and where to search for information is an important strategy for coping with the challenges presented by the high quantity of information and the prevalence of low quality information. One can search the entire Web, relying on a general purpose search engine to return documents ranked not only by relevance to the query but also by factors that reflect its authority and popularity. One can also choose a portal devoted to a specific domain ("a sphere of activity, concern, or function; a field" [1]). The contents of the portal might have been manually curated, with documents being selected only if they meet some criteria for quality. Or, documents might have been included because the contents, or the sequence of links that led to finding the page, met automated criteria that suggested the page is relevant to the domain. In either case, the universe of possible pages to be returned is more limited than for a general search engine, possibly decreasing the likelihood of the portal returning completely irrelevant results. However, limiting the size and nature of the document collection can also inhibit the effectiveness of link-based algorithms that are used so successfully by general search engines. Sometimes the information task itself dictates a particular information resource, such as an employer's intranet. The resource chosen by a particular searcher for a particular task may depend on prior knowledge about the topic of the information need, about the candidate resources, about the ease of searching or browsing each candidate resource, about the relative authoritativeness of a resource, about the intended audience or presentation style of candidate resources, or it may depend on other personal preferences.

We are primarily interested in those cases when a searcher chooses a domain-specific document collection instead of using a general search engine to search the entire Web. The research described in this dissertation is motivated by the desire to support domain experts when they are using domain-specific digital libraries to satisfy certain kinds of information needs. Each of these stipulations has implications for the scope of, and approach taken in, this work.

1.1. Domain-specific Digital Libraries

The term "digital library" has been used in myriad ways. Gonçalves and colleagues [2] defined a formal model for digital libraries that they call the *5S* model, where the five S's are: Streams, Structures, Spaces, Scenarios, and Societies. Their informal definition affords a useful summary: "Informally, a digital library involves a managed *collection* of information with associated *services* involving *communities* where information is stored in digital formats and accessible over a network. Information in digital libraries is manifest in terms of *digital objects*, which can contain textual or multimedia content (e.g., images, audio, video), and *metadata*. … Basic services provided by digital libraries are indexing, searching, and browsing." In summary, digital libraries involve collections of documents (possibly accompanied by metadata, i.e., data about the documents, that may be descriptive or structural), users (community), and services (indexing, searching and browsing). In this work, we interpret the term digital library broadly to include publicly available web portals, specialized collections that are accessible electronically but that have access restricted

to members of an organization or holders of a subscription, and enterprise information portals that may provide access to either externally available documents produced by an organization (extranet), internally available documents (intranet), or both.

By a domain-specific digital library, we mean a digital library that pertains to a particular area of knowledge or activity (a domain). A domain-specific library has a collection of documents, which are pertinent to a particular domain, and a retrieval system that provides access to those documents. While the library may provide browsing services in addition to searching services, this work focuses only on the searching capabilities. The retrieval system typically has an index, consisting of a representation for each document, a query module that accepts user requests in a query language that is understood by the retrieval system, a search module that matches the user requests to document representations, and an interface to present the retrieved documents, usually in ranked order, to the user. Document representations typically consist either of words extracted from the document (full text indexing), keywords assigned from a controlled vocabulary appropriate to the domain (keyword indexing), or a combination of both. Figure 1.1 shows a schematic of a digital library retrieval system.

Fagin and colleagues [3] studied corporate intranets (which they also called the workplace web). A corporate intranet is not necessarily a domain-specific digital library, since there may be more emphasis on documents related to the corporation itself (such as personnel directories and corporate policies) rather than about the domain in which its activities occur. However, their analysis is useful for

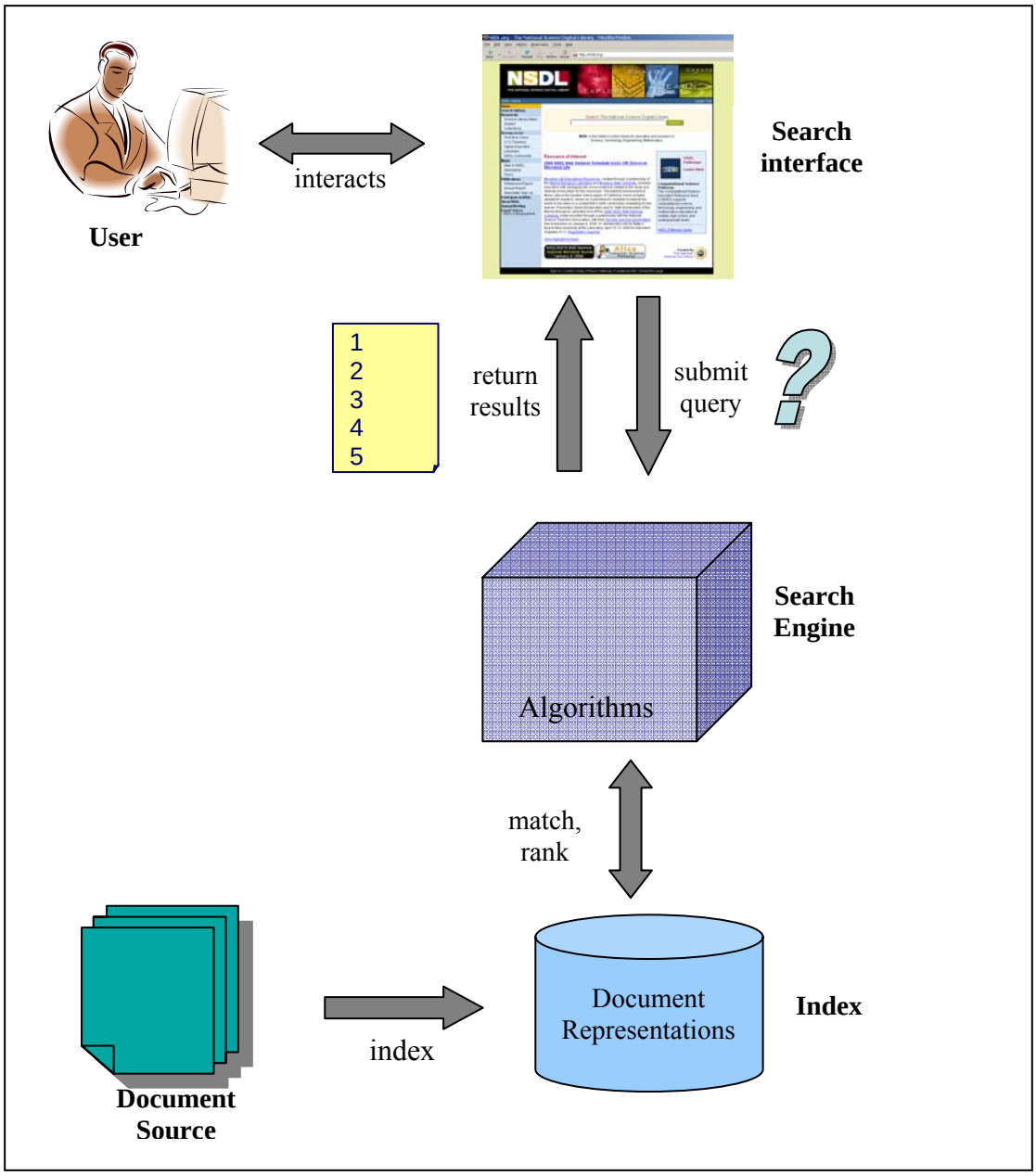


Figure 1.1 Schematic of a digital library retrieval system

understanding some of the challenges posed by domain-specific digital libraries. The authors observed four characteristics, which they posited as axioms, that distinguish corporate intranets from the Internet at large: (1) "Intranet documents are often created for simple dissemination of information, rather than to attract and hold the attention of

any specific group of users;" (2) "A large fraction of queries tend to have a small set of correct answers (often unique), and the unique answer pages do not usually have any special characteristics;" (3) "Intranets are essentially spam-free;" and (4) "Large portions of intranets are not search-engine friendly".

Fagin's axioms, intended to explain several characteristics of intranets that affect intranet searching, are generally true of domain-specific digital libraries as well. Hubs, which are web pages that contain links to various useful pages about a particular topic [4], may be uncommon or absent. As a result, the link-based algorithms that are so effective for general web search are not very useful for searching intranets and may also be ineffective for domain-specific digital libraries. The lack of redundant content places additional pressure on retrieval algorithms. If any one of many relevant pages will satisfy the user, then the search engine need only return one of the pages at a high rank to be successful. If only a particular page will suffice, then the demand for accuracy is much higher. On the other hand, there is less need in intranet retrieval systems for defensive algorithms that can detect efforts to manipulate search engine rankings. For example, information in metadata tags can be useful for intranet searching — and for searching domain-specific digital libraries — whereas metadata tags are generally ignored by Internet search engines because metadata tags have been so often abused on the Web. Fagin's final axiom reflects the diversity of document types and formats present in corporate environments and the prevalence of dynamically generated content resulting from database queries. If a large proportion of corporate information is stored in database records, instead of being stored in a

document repository, the information is not available for indexing by the search engine. Although developing crawlers that automatically find web pages with searchable forms and developing applications to automatically fill out such forms are areas of active research, most information in online databases remains "hidden" from search engines. The technical challenges presented by the different document formats and access methods may occur in some domain-specific digital libraries as well, but these challenges are largely orthogonal to issues surrounding the content in the documents. The fundamental implication of these axioms is that current web searching techniques employed by the major search engines may not be adequate for successful retrieval from domain-specific digital libraries.

1.2. Information Needs

The information needs of domain experts, and their information seeking behavior, are affected by many factors: task, context, urgency of the problem, time constraints, level of domain expertise, amount of prior knowledge related to the particular need, and the information goal (such as learning about a topic vs. finding facts or instructions) [5-7]. Two groups of professionals whose information needs have been studied, physicians and engineers, provide illustrative examples regarding the types of information needs that experts encounter and some of the constraints imposed by the workplace settings in which the needs occur.

In a review of the literature on the information needs of physicians, Gorman commented on a sample of typical questions asked by primary care physicians during

routine office practice. He noted that "although some of these questions are fairly simple and direct, many of them are complex, multidimensional questions embedded in the context of the individual patients." One of the sample questions was "In a woman with history of delivering at 33 weeks, now having Braxton-Hicks contractions at 32 weeks, on terbutaline and bedrest, in breech position, is c-section indicated if labor cannot be stopped?" [6]. Clearly the question is complex and there are multiple details about a particular patient situation embedded in the question. A certain degree of domain knowledge is needed to even understand why various elements are included in the question and what constraints they impose on the clinical situation and on the desired answer.

Ely and colleagues [8, 9] developed two taxonomies of clinical questions collected during observational studies of family practice physicians. One taxonomy was by topic, the other taxonomy was by generic questions, which abstract the entity types and relationship types in the question. Examples of generic questions are "What is the cause of symptom X?" and "Should I use treatment Y for condition X?" The most common generic question types were about the cause of a particular symptom, about the proper dose for a particular drug, and about how to manage a particular disease or finding. That the specific questions asked by physicians could be abstracted into generic questions suggests that many questions share a relatively small set of entity types (such as *disease*, *drug*, *symptom*, and *therapy*) that are connected by a finite and predictable set of relationship types (such as *causes*, *treats*, and *prevents*).

Freund et al. [5] studied contextual influences on the information behavior of software engineering consultants and developed a model with four spheres: the consultant, the consulting engagement, the work task, and the problem situation. Work tasks had two dimensions: the high-level task in a consulting engagement (such as project management, training, mentoring, and technical support) and the technical task (such as design, implementation, configuration, and integration). The problem situation determines not only the topic but also the information goal. The typical information goals they identified were: *learn about*, *collect advice to make a decision*, *find instructions*, *find facts*, and *find examples to reuse*. All of the information goals except *learn about* are specific, targeted goals closely tied to a particular context and task.

From these examples, we conclude that some of the information searching tasks of domain experts require targeted information. These are searching tasks in which the information need is specific, often motivated by a particular work task or situation. The information need is likely to be satisfied by one, or a few, documents that provide the answer to a relatively well-defined question. The tasks are precision-oriented, meaning that search precision (exclusion of irrelevant documents from a search result) is more valuable than recall (return of all relevant documents somewhere in the search result).[1] While domain experts can have some information tasks that are recall-

[1] In search systems with ranked results, rather than set-based results, ranking relevant documents higher than irrelevant documents is functionally equivalent to excluding irrelevant documents. Evaluation, therefore, measures the quality of the ranking instead of precision and recall.

oriented, for which it is important to find all relevant documents, and some tasks that are open-ended, for which iterative, exploratory searching is necessary and for which finding serendipitous information is highly valuable, in this work we focus on supporting precision-oriented information tasks.

It is clear that, for both of these groups of domain-experts — and probably for experts in other domains as well — relevance is situational [10]. That is, what constitutes a relevant document for a particular information need is determined largely by features of the searcher's internal (cognitive) situation and by features of the external situation in which the information need occurs. For a document to be relevant, it may need to provide the answer to a specific question, or cover a particular aspect of a topic, not just be about the general topic of the search. Physicians need documents that fill gaps in their knowledge and that can provide information that is applicable to a given patient, who may have multiple characteristics that influence the decision being made. Engineers need documents that provide information that will help a particular individual complete a particular task in a particular context.

Another characteristic of experts' information needs is that experts have far more information needs than they have time to pursue. Time constraints force them to make decisions about which information needs to pursue and which resources to search. We, again, note examples from medicine and engineering.

Based on a review of the literature, Gorman estimated that physicians have two questions for every three patients [7, 11]. Only about 30% of these questions are immediately answered [7, 8, 11] despite a plethora of electronic and nonelectronic

medical knowledge resources. The two factors that Gorman found to be significant predictors of whether physicians would actively seek answers to questions were belief that a definitive answer exists and the urgency of the patient's problem [7]. Curley et al. studied physicians' selection of knowledge resources in the context of patient care using a cost-benefit framework. They found that the significant variables affecting resource selection, which were *availability*, *searchability*, *understandability*, and *clinical applicability*, were all related to the cost of finding useful information. Characteristics of resources that can result in information having greater benefit, such as *extensiveness* and *credibility*, did not affect resource use [12].

Similar findings are available in studies of engineers. Freund et al. studied software engineers who reported spending about 20-30% of their time looking for and consulting information sources. They quoted one participant who indicated that the time would have been greater if they could spare more time [5]. Fidel and Green used interviews to study how engineers select information sources. The most common factors were related to accessibility (*sources I know*, *saves time*, *is physically close*, *has the right format*, *can give the right level of details*, *is accessible*, *is available*) and quality (*can give data that meets the needs of the project*, *is most likely to have the information needed*, *is reliable*).

In summary, experts have many information needs and limited time to pursue them, which affects their choices of which information needs to pursue and which resources to consult. Their information needs are often specific, are often shaped by

the workplace context in which they occur, and often entail predictable types of information.

1.3. Domain Experts as Information Seekers

In a classic article, Belkin et al. put forth the anomalous state of knowledge (ASK) hypothesis, suggesting that "an information need arises from a recognized anomaly in the user's state of knowledge concerning some topic or situation and that, in general, the user is unable to specify precisely what is needed to resolve that anomaly" [13]. We posit that the ASK hypothesis is often false for domain-expert searchers. An expert in a domain is likely to have a mental framework for understanding what kinds of information she needs, to understand how entities that are important in the domain usually relate to each other, and to know what kind of information will satisfy a particular information need. An expert in a domain is also likely to be familiar with the types of documents often created in the domain, and the types of documents that can be found in various resources, such as a particular digital library.

Domain experts often have extensive knowledge about how information in the domain is typically organized and expressed within documents. This claim is supported by observations in multiple domains. Dillon [14] showed that experienced researchers have a mental model of typical academic articles. When given pieces of cut up articles, with approximately every other paragraph removed, experimental subjects rapidly assembled the fragments into an order that followed an Introduction-Method-Result-Discussion format. In related work, Bishop [15] described a series of

focus groups, interviews, and usability tests investigating how academic researchers use structural components of scientific journal articles (such as figures, tables, references, author lists, methods sections) to select which documents to use, to read and comprehend the documents, and to extract, transform and use the information in their own work. During a usability study of an experimental digital library system for forest management, the investigators observed that the forestry professionals exhibited a striking familiarity with the organization of long documents, rapidly homing in on sections of interest [16]. When physicians were tasked with a scenario of familiarizing themselves with the medical record of a patient for whom they were to assume responsibility, the physicians rapidly focused attention on the relevant portions of the relevant documents in the medical record, attending only to information that would influence the scenario [17].

1.4. Semantic Components

We believe that the location of words in relation to the logical structure and semantic organization of documents can provide useful data that can inform the retrieval and ranking of documents in IR systems. Most IR systems use the words in documents (usually by calculating the frequency of occurrence of each word in a document and in the document collection as a whole) or keywords (usually chosen from a controlled indexing vocabulary) to represent document content. Queries are represented similarly, as a collection of words appearing in a natural language query, as keywords from an indexing vocabulary, or as a combination of both. When IR

systems match the text words or keywords that represent the documents and queries, the list of documents they retrieve is sometime unsatisfactory because these representations are only weak surrogates for the actual document content and information needs. The user wants the IR system to match the underlying intent of queries to the semantic content of documents, which is a more difficult problem than term matching. Consider, for example, a physician who must decide whether to administer a vaccine that prevents polio to a patient with a respiratory infection. (Many vaccines should not be given in the presence of a current illness.) The query "polio prevention respiratory infection" will return documents about polio, about polio causing respiratory failure, about polio causing symptoms of an upper respiratory infection, about respiratory infections, about preventing polio, and about preventing upper respiratory infections, in addition to possibly returning documents about preventing polio in the presence of a respiratory infection.

In this dissertation we introduce a model we call *Semantic Components* that takes one step toward using additional semantics to improve the matching of queries to documents. We supplement existing representations of document content by exploiting domain-specific characteristics of document types and content. Semantic components provide a richer representation of document content than full text or keyword indexing techniques. The representation occurs at a subdocument level, providing additional information about where various kinds of information are located in a document. Semantic components also allow use of an extended query language to capture additional detail about the information need.

The semantic components model has two main elements: *document classes* and *semantic components*. Documents are classified by grouping documents that will tend to contain the same kinds of information. Different domains and document collections can have different axes that are most appropriate for classifying documents, such as topic type or document purpose. In health-related collections, we have found topic type to be useful. For example, such collections often have documents about diseases (one document class) and documents about medications (another document class). Documents within a class tend to contain characteristic types of information, usually information about important aspects of the main topic of the document. For example, in the medical domain, documents about diseases often contain information about *diagnosis* and *treatment* whereas documents about medications often contain information about *dosage* and *side effects*. We call these types of information semantic components. We call the set of document classes and associated sets of semantic components that are identified for a particular document collection a *semantic component schema*. Table 1.1 shows part of a semantic component schema (with two document classes and their semantic components) for the document collection that we used for the searching study that is described in Chapter 8.

A *semantic component instance* is the text in a document that contains information about an aspect of the main topic (a subtopic) that is the semantic component. Semantic component instances may or may not correspond to structural elements in documents, can overlap with other instances, and may consist of discontiguous segments of text. Any given text in a document can belong to zero, one, or multiple

semantic component instances. A semantic component is the type (that is, a label that indicates the type) for a semantic component instance that corresponds to a particular aspect. For example, in a document about a particular disease, a text segment that describes the diagnosis of the disease is an instance (or part of an instance) of the *diagnosis* semantic component. A segment that describes the treatment of the disease is an instance (or part of an instance) of the *treatment* semantic component.

Table 1.1 Partial semantic component schema used in a searching study

Document Class	**Semantic Components**
Clinical problem	General information Diagnosis and evaluation Referral Treatment
Drugs	General information Practical information Target group Effect Side effects, interactions and contraindications

Figure 1.2 shows instances of two semantic components, *epidemiology* and *etiology* (causation), highlighted in a document about asthma (in a class of documents about diseases) that has been excerpted and highly condensed from a web page.[2] Note that each semantic component instance consists of two discontiguous segments, that the instances do not correspond to the document structure, and that the semantic component names do not correspond to the words used in document subheadings.

We use semantic components for information retrieval in three ways:

[2] Based on condensed excerpts from http:// www.emedicinehealth.com/asthma/article_em.htm

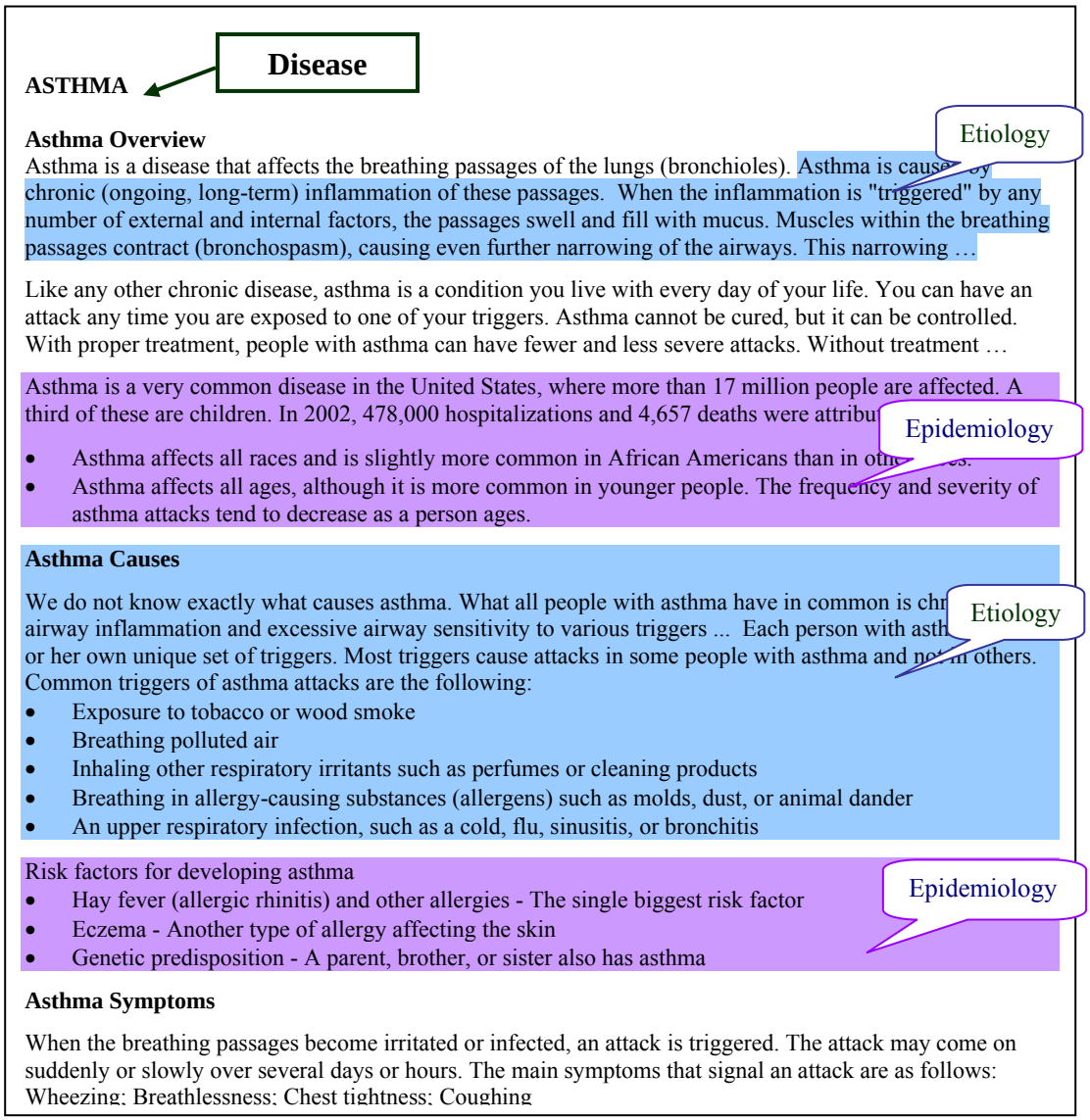


Figure 1.2. Two semantic component instances

1. We allow a searcher to refine a search by searching for one or more query terms within a specific semantic component in addition to searching for terms in whole documents. A query might consist of the topical term “asthma” plus a request for documents that contain the term “pregnancy” within a *treatment* semantic component instance. A searcher can also specify query terms (the

same terms or different terms) to be searched for within multiple semantic components.

2. We allow a searcher to specify a preference for documents containing particular semantic components without searching for a specific query term within a component. In this case, a query could consist of the topical term "asthma" and a preference for documents that contain a *treatment* semantic component.
3. We display a list of the semantic components present in each document in a hit list (the list of documents returned by the search engine) plus an indication of the size of each component to provide a short synopsis of the document that can help a searcher decide whether a particular document is likely to be useful. For example, a search might return one document about asthma with a *diagnosis* component consisting of 400 words and a *treatment* component with 100 words and a second document with only a *treatment* component that consists of 500 words. A searcher interested in asthma treatment might choose to look at the second document first because it appears to contain more information about treatment.

The semantic components model leverages an expert's knowledge about information organization in a domain, allowing him to use characteristics of the domain and of document classes to create a richer representation of his information need than a list of search terms. The semantic components model also allows a richer representation of the content and semantic structure of documents. These enhanced

representations offer search engines an opportunity to more accurately match documents to information needs.

This dissertation provides a detailed description and analysis of the semantic components model. It also reports the results of a series of investigations into the potential usefulness of semantic components for retrieval in domain-specific digital libraries. We studied manual identification of semantic component instances in documents in this work; automating the identification of semantic components instances is an important topic for future work.

1.5. Domains, Settings, and Collaborations Involved in this Research

The goal of this research is to support users with domain-specific tasks, so it was important to study the semantic components model in the context of specific domains. The two domains in which most of this work is focused are (1) medicine and (2) public land management. The reasons for choosing these two domains are largely pragmatic. Most of the research reported in this dissertation was done in the medical domain, for the following reasons:

- The author is a physician as well as a computer scientist, so the medical domain is a natural setting for exploring these ideas.
- The medical domain is often used for information retrieval and information science research because of the rich terminological resources and bibliographic databases that are available. This work builds on a long history of research in the medical domain.

- The research was largely funded by a National Science Foundation (NSF) grant from the International Digital Government program.[3,4] Our government partner was sundhed.dk, the national Danish health portal [18] (see below).
- Much of the preliminary work was funded by a National Library of Medicine postdoctoral research fellowship.[5]

Some of the early work described in this dissertation also builds on the work of previous graduate students, Shawn Bowers and Mathew Weaver. Both students built applications to support informational activities of public land managers from the U.S. Department of Agriculture (U.S.D.A.) Forest Service [19, 20]. In the process of gathering requirements and designing the applications, they learned much about the domain, about documents produced as part of the land management process, and about the work tasks and information flow of forest supervisors and other land managers. The semantic components model was, in part, inspired by the Schematics Browser [19] and therefore public land management was a natural domain for initial exploration and testing of these ideas.

Sundhed.dk[6] is a web-based portal that provides access to information about health and medicine and about the Danish healthcare system. Intended for both healthcare professionals and citizens in Denmark, it has been operational since 2001. As of July

---

[3] *Accelerated Indexing in a Domain-Specific Digital Library* (NSF award 0514238)
[4] Some of the work was also funded by a grant to study the use of superimposed information for education in a digital library (NSF award 0511050) and by a grant to develop generic mechanisms for capturing and using superimposed information (NSF award 0534762).
[5] National Library of Medicine Training Grant 5-T15-LM07088
[6] "sundhed" can be translated as "health."

2006, sundhed.dk hosted nearly 25,000 of its own documents and also provided links to a variety of external sources. Sundhed.dk generously provided help in understanding its organization and editorial processes, donated considerable employee time, and provided access to both its documents and to a configuration file that specifies many parameters for its search engine.

The research reported in this dissertation would not have been possible without a number of essential collaborators who provided resources, gave feedback, recruited study participants, facilitated arrangements for the user studies, and contributed to our understanding of the domains and settings that served as sources for both inspiration and testing of the ideas underlying this work. In addition, the research has been greatly enriched by exchanging ideas with a number of people, but especially with Lois Delcambre, Professor of Computer Science at Portland State University, and Marianne Lykke Nielsen, Associate Professor at the Danmarks Biblioteksskole (Royal School of Library and Information Science) in Denmark.

The following paragraphs are intended to credit various individuals who played a significant role in the conduct of the research described in this dissertation.[7] Although the descriptions cannot quantify the value or the amount of work contributed by each individual, this should at least make it clear that it was a joint effort to produce this research.

---

[7] The contributions of my advisor, Dr. Delcambre, are not specifically mentioned because she was involved in every stage of the research.

Dr. Marianne Lykke Nielsen. The indexing and searching studies described in Chapters 7 and 8 were a joint effort with Dr. Nielsen and would not have been possible without her contributions. Dr. Nielsen initiated and facilitated the collaboration with sundhed.dk, and also with Drs. Peter Vedsted and Jens Rubak. She arranged for our research license for the use of the Ultraseek software from Frans la Cour of Ensight (now Metier). She also initiated our collaboration with Dr. Kalervo Järvelin. She proposed the initial designs for the indexing and searching studies (such as number of participants, duration of each subject's participation, number of documents to be indexed, and number of searchers). The preliminary interviews with indexers and users of sundhed.dk were conducted by Dr. Nielsen and the author, but primarily by Dr. Nielsen because the participants, who were all Danish, found it easier to converse in Danish than in English. Development of the semantic component schemas used in the indexing and searching studies was a joint process between Dr. Nielsen and the author. Dr. Nielsen provided valuable advice and pointers to the information science literature, especially regarding keyword indexing and the user-centered approach to IR. She coordinated the Danish indexers who performed semantic component indexing for the documents we used in the searching study. She also facilitated the local arrangements for the indexing and searching studies that were both held in Denmark. She provided valuable feedback regarding drafts of the questionnaires and did formal pilot testing of the materials we used in the studies. Establishing the requirements for the experimental search system and carrying out the indexing and searching studies was a team effort by Dr. Nielsen, Dr. Delcambre, and

the author. In addition, Dr. Nielsen recognized the importance of analyzing the sequences of queries issued by the searchers and is leading work that will be published elsewhere to analyze the refinements that searchers made when their initial queries were unsuccessful.

Vibeke Luk. Ms. Luk was our primary contact at sundhed.dk and provided support at every stage of this research. She helped us understand how the web portal is organized and how the indexing and searching processes are implemented. She arranged for our access to the documents and configuration files. She also assisted with local arrangements for each of our studies. In addition, she recruited the participants in our indexing study and the indexers who indexed documents for the searching study. She also participated in the indexing study as a subject and indexed some of the documents for the searching study.

Dr. Peter Vedsted. Dr. Vedsted is a family physician and researcher at The Research Unit for General Practice at the University of Århus in Denmark. He was also a key developer of praxis.dk, a predecessor to sundhed.dk. Dr. Vedsted provided useful feedback about the semantic components model early in the research, obtained funding from the regional government in Århus to support physician participation in the searching study, helped design the scenarios we used in the searching study, developed the reference standard of relevance judgments for the searching study, and helped recruit physicians to participate in the searching study.

Dr. Jens Rubak. Dr. Rubak, also of praxis.dk and a family physician in Århus, recruited physicians to participate in the searching study.

Dr. Kalervo Järvelin. Dr. Järvelin is Academy Professor in the Department of Information Studies at the University of Tampere in Finland. Dr. Järvelin collaborated with us to develop a new session-based metric, sDCG, for evaluating IR systems in interactive searching studies. We discuss sDCG, and describe how we used it as one method for assessing search results, in Chapter 8.

Dr. Timothy Tolle. Dr. Tolle, recently retired from the USDA Forest Service, provided valuable assistance in understanding the work tasks of forestry professionals and the processes and documents mandated by the National Environmental Protection Act (NEPA). He, along with Dr. Nielsen, provided valuable feedback regarding early ideas that led to the semantic components model. Dr. Tolle also developed semantic components to describe Environmental Analysis and Decision Notice documents as part of our early studies.

1.6. Contributions

The contributions of this dissertation are:

1. An informal and a formal description of the semantic components model
2. A prototype implementation of semantic component indexing software, which we used to perform semantic component indexing for the searching study
3. A discussion of how we developed semantic component schemas to describe document collections using document classes and semantic components

4. An analysis of using the semantic components model to express the information needs represented in a published taxonomy of clinical medical questions
5. An evaluation framework for assessing the accuracy and consistency of semantic component indexing and keyword indexing
6. An indexing study that compared keyword indexing and semantic component indexing by participants who were experienced with keyword indexing for similar documents
7. A searching study in which domain experts completed realistic search scenarios using an interface that allowed searching with semantic components and a comparison interface that mimicked an existing retrieval system to search a familiar domain-specific digital library
8. A prototype implementation of a search system that uses semantic components and that we employed in the interactive searching study

We now detail the specific research efforts led by the author.[8] She led the development and elaboration of the semantic components model (which resulted from discussions and key feedback from Drs. Nielsen, Delcambre, and Tolle in response to her earlier ideas) and she formalized the semantic components model (Chapter 3). She also conducted the initial document sampling and analyses of the sundhed.dk documents, led the development of the initial semantic component schema for

[8] "led" is the most appropriate description because nearly every significant research activity involved some amount of input and collaboration by members of the research team.

sundhed.dk (with input from Dr. Nielsen), developed the semantic component schema for UpToDate® documents, and analyzed the evolutions of these schemas over time (Chapter 4). She performed the mappings from the information needs taxonomy to the semantic component schemas (Chapter 5). She developed the indexing evaluation framework, including the analyses of candidate evaluation metrics, with input from Dr. Nielsen about evaluation of keyword indexing (Chapter 6). She led the application for approval of the indexing and searching studies by the Human Subjects Research Review Committee at Portland State University and development of the questionnaires for the indexing and searching studies. She designed and implemented programs to select and prioritize documents to be indexed for the searching study (based on queries that she and Dr. Nielsen issued to produce the initial lists of documents), and designed and implemented the semantic component indexing software (Chapters 7 and 8). She performed the technical design and implementation of the experimental searching system (on top of the Ultraseek search engine being used by sundhed.dk) that was used in the searching study (Chapter 8). She also designed and implemented the data analyses reported in the dissertation. In particular, she designed and implemented the analysis of the effect of document selection for semantic component indexing on the searching study results and the overall approach to handling the data from the interactive searching experiment, which resulted in the recognition of the need for a metric to compare system performance in multiple query sessions (Chapters 7 and 8).

The remainder of the dissertation is organized as follows. Chapter 2 of this dissertation contains an introduction to information retrieval and related areas of

research to provide general background to the dissertation. Chapter 3 provides a more detailed introduction to semantic components, including an introduction to semantic component indexing that is facilitated by a description of the prototype indexing software and a formal description of the model. Chapter 3 also includes a detailed overview of the research presented in subsequent chapters. The discussion of semantic component schemas, including how we developed the schemas to describe document collections and the lessons we learned from iterative refinements to those schemas, is presented in Chapter 4. Our study of using semantic components to express information needs in a taxonomy of clinical questions is described in Chapter 5. In Chapter 6 we develop the evaluation framework for assessing semantic component and keyword indexing. In Chapter 7 we present the study of semantic component and keyword indexing and use the evaluation framework developed in Chapter 6 to analyze the results of the study. In Chapter 8 we describe the interactive searching study and analyze the results. We also describe the prototype implementation of a search system using semantic components, which was an essential component of the experimental searching system used by the study participants. In Chapter 9 we present our conclusions and discuss areas for future work.

Chapter 2 Background and Related Work

We begin this chapter with a brief introduction to information retrieval systems. The semantic components model builds on concepts from information retrieval research and is intended to supplement, not replace, existing information retrieval techniques. We then describe some areas of existing work that use techniques similar to the semantic components model. We also describe some work that uses different methods intended to achieve the same goal, that is, incorporating additional semantic and domain-specific information into retrieval systems. We discuss additional related work in later chapters when that work relates more specifically to the research presented in a single chapter.

2.1. Introduction to Information Retrieval Systems

Information retrieval (IR) systems return documents in response to queries that express an information need.[9] The retrieved documents can be full text documents or bibliographic records that describe the full text documents. A typical IR system consists of components that:

- interact with the user to accept queries and return search results
- match documents to queries

[9] Although information retrieval systems for multimedia objects exist as well, this dissertation focuses on information retrieval systems for text.

- create and store concise representations of each document to facilitate matching documents to queries (an index)
- store the documents that are returned to the user as search results (which might be abstracts instead of the complete documents)

In relational database systems, queries precisely specify the records that the system should return. In contrast, the queries in IR systems only approximate the user's actual information need, which may depend on such things as the user's situation, pre-existing knowledge, and depth of interest. Similarly, document representations stored in the IR system's index are incomplete indications of the information each document contains. As a result, IR systems typically fail to retrieve all potentially relevant documents and often retrieve many documents that are not relevant to the user's information need. Methods that retrieve more of the relevant documents (increase recall) generally also retrieve more nonrelevant documents (decrease precision).

2.1.1. Indexing

Indexing consists of creating a representation for a document that can be stored and retrieved in electronic form. Creating document representations can be done manually or automatically. Manual indexing is usually performed by a trained indexer and involves assigning a small number of keywords (single words or phrases) to describe what a document is about. We refer to indexing with keywords as *keyword indexing*. In most cases, keywords are chosen from a restricted set of words or phrases called a *controlled vocabulary*. Automatic indexing usually consists of recording each

word in the document, the frequency of the word's occurrence in the document, and the position of each occurrence. We refer to this type of indexing as *full text* indexing.

Some words, such as "the", "on", and "in", occur in so many documents that they may not be useful for distinguishing the content of one document from the content of other documents. Most automatic indexing systems have a list of such words, called *stopwords*, that they ignore in the indexing process. The type of indexing, either keyword or full text, is orthogonal to the method, either manual or automatic. In practice, however, manual indexing systems usually produce keyword indexes and automated indexing systems usually produce full text indexes. Some automated keyword indexing systems exist, although they are often used as computerized assistants to manual indexers.

The entire collection of text words or keywords used to index all the documents in a collection is referred to as the *indexing language*. We will refer to text words (and phrases, for IR systems that extract phrases as well as individual words) or keywords[10] used for indexing as *terms*. Terms, especially keywords, are also sometimes called descriptors. A *concept* is a mental model of an object or an idea that is represented by one or more terms. Two important characteristics of indexing that can affect the retrieval process are exhaustivity and specificity. *Exhaustivity* is the degree to which indexing represents all the concepts that appear in a document. *Specificity* is the level

---

[10] A keyword can consist of a single word, a phrase, or multiple words connected with punctuation symbols to represent a single concept

of abstraction at which a concept is represented. For example, if a document is about dogs, the concept *dog* could be indexed using "dog," "mammal," or "pet." The term "dog" is more specific than either "mammal" or "pet". More exhaustive indexing means that a document can be retrieved in response to a greater variety of queries, which can be either an advantage or a disadvantage. Higher specificity usually results in fewer instances of returning unwanted documents, but can sometimes result in failure to retrieve desired documents.

Extensive discussions of the theory and practice of keyword indexing, and of the relative advantages and disadvantages of keyword versus full text indexing, are available elsewhere [21-27]. Here we summarize salient points and note that many modern information retrieval systems use a combination of keyword and full text indexing.

Keyword indexes are more compact than full text indexes, which include all the words in a document. When computational resources were more limited than they are now, storing keywords and comparing query terms to sets of keywords was computationally more feasible than using all the words in a document. Also, the use of a controlled vocabulary allows concepts to be represented by one agreed-upon term, instead of being represented by the multiple different words that can be used in natural language. Hierarchical controlled vocabularies that contain broader term/narrower term relationships allow the searcher to expand and narrow the focus of a search as needed. However, human intellectual keyword indexing is expensive and prone to inconsistency [28]. The indexer must not only determine what the document is about,

and translate the concepts into terms, but he must also anticipate which terms might be used by searchers wanting to find the document [24]. Also, existing vocabularies can fail to adequately represent either the documents' contents or the users' information needs. Poor representation can occur either because the scope of the vocabulary is inadequate or because the vocabulary is outdated. The lower exhaustivity of keyword indexing compared to full text indexing, which was an advantage when computing power was limited, can be a disadvantage for searchers whose interest may be about concepts less central to the document, or that were not deemed important by the indexer.

Automated full text indexing is less expensive than manual keyword indexing. Full text indexing is also more exhaustive than keyword indexing because it attempts to represent all of the content of a document, not just the main concepts. The vocabulary used for full text indexing is always up to date because it mirrors the vocabulary used by the document author. However, full text indexing requires the searcher to anticipate the language used by the author in relevant documents. The burden is on the searcher to be familiar with any specialized terminology and to consider synonyms and terms at broader and narrower degrees of specificity [21, 24]. Mismatch in the use of inflexional variants by author and searcher, such as different verb tenses, can also cause retrieval failures. Some automated indexing systems use *stemming*, the conflation of variants by reducing them to a common stem, to reduce such mismatches. Stemming can be very effective, such as by conflating "rains" and "raining" to "rain," but algorithms are imperfect. For example, the Porter stemming

algorithm [29] fails to conflate some variants (such as “mouse” and “mice”) and conflates some words with different meanings to the same stem (such as “dais” and “days” to “dai”).

2.1.2. Queries

A *query* is the expression of a user’s information need that is input to the IR system. Some systems accept queries expressed in natural language, typically treating the query as a set (or a list or a bag) of words. Other systems accept queries expressed as clauses connected with Boolean operators (such as *AND*, *OR*, and *NOT*). Additional refinements include restricting the search to specific bibliographic fields (such as title or author), searching for phrases instead of words, or using proximity operators that require query words to occur within a specified interval of words. Some modern systems use a combination of operators. The queries allowed by an IR system are expressed in its *query language*.

2.1.3. Retrieval

When an IR system receives a query, it searches its stored indexes for matches between the query representation and document representations. For a Boolean query, the matching algorithm is set based. All retrieved documents satisfy exactly the constraints indicated by the query. For example, the Boolean query “cat AND platypus” would retrieve a document only if the indexing data for that document contained the term “cat” and the term “platypus.” The Boolean query “cat OR

platypus" would return any document whose indexing data contained either the term "cat" or the term "platypus." For natural language queries, a similarity algorithm that allows partial matching retrieves and orders documents based on a measure of similarity between the query and each document. Similarity algorithms are often based on the vector space model [30], in which documents and queries are represented by weighted vectors. Each element in the vector represents a term in the document (or query) that is weighted according to the term's frequency in the document (or query) and its frequency in the entire collection. Similarity can be calculated as the cosine of the angle between two vectors. For example, the ranking of documents in response to the query "cat platypus" would be determined by the number of times the words "cat" and "platypus" appeared in each document and by the relative frequency of "cat" and "platypus" in the entire document collection. Newer models for ranked retrieval that have gained considerable popularity are the probabilistic model [31, 32] and language models [33]. Although the probabilistic model and language models are based on different mathematical theories than the vector space model, both models also use word frequencies to rank documents in response to queries.

Some IR systems, especially web search engines, use content-independent features in addition to similarity between document and query to rank candidate documents. Two well-known algorithms that use the hyperlink structure of the Web to estimate the relative popularity and authority of web pages are the page-rank algorithm [34] and the hypertext-induced topic selection (HITS) algorithm [4]. The words in anchor text (the text in the clickable link on web pages) and in URLs have been found to be quite

useful as an indicator of document content [35]. In other words, if document A contains the text "everything you want to know about cats" in the anchor text for a hyperlink to document B, and the URL for document C is "http://www.animals.com/cats," then an IR system might boost the rankings of documents B and C in the results for the query "cats."

2.1.4. Evaluation

Evaluation of IR systems is a complex topic; here we briefly summarize prominent issues that are related to the research presented in this dissertation. Historically, the most common approach to IR system evaluation is the use of experimental test collections, sometimes referred to as the Cranfield paradigm in reference to early experimental evaluations at Cranfield University [36]. Test collections have three components: documents, statements of information need, and relevance judgments that indicate which documents are relevant to each information need. Voorhees notes three simplifying assumptions in the Cranfield paradigm: (1) "... relevance can be approximated by topical similarity" (2) "... a single set of judgments for a topic is representative of the user population" and (3) " ... the lists of relevant documents for each topic is complete (all relevant documents are known)" [37]. She also notes that these assumptions are usually violated. Relevance is more complex than just topical similarity and user populations are diverse. Except in the smallest test collections, determining the relevance of every document is not feasible. However, despite widespread awareness of the limitations of the Cranfield paradigm, researchers

continue to find it useful for evaluating IR systems. The Cranfield approach has some potent advantages over alternative approaches. By keeping all other elements the same, a change to a single component can be evaluated in relative isolation from the other parts of the system [37]. Furthermore, test collections are reusable, allowing many experimenters to profit from the effort invested in creating a single test collection.

The most well-known examples of experiments using the Cranfield paradigm are the annual Text REtrieval Conferences (TREC) that are sponsored by the National Institute of Standards and Technology (NIST) [38, 39]. Participating research groups, from both academia and industry, can choose to work on one or more of the tasks that are available in the different tracks that run in a given year. All of the tracks run on an annual cycle that involves distribution of document collections and topics (descriptions of information needs) pertinent to each task. NIST provides the infrastructure needed for creating large scale test collections, particularly the organization of thousands of expert relevance judgments, and sponsors an annual conference where participating research groups compare and discuss their results. Most of the document collections used in TREC contain millions of documents and so only a fraction of the documents are judged for relevance. Documents are chosen to undergo human relevance assessment using the technique of *pooling*. The pool of documents to be judged is formed from the top X documents (where often $X = 100$) in the ranked results submitted by each participating research group for a given query. Because of overlaps in the documents returned by different groups, a pool is usually

only about one third as large as the theoretical maximum size of the pool (number of groups * *X*). For most of the TREC experiments, the relevance judgments are binary and pertain only to whether a document addresses the subject of the given topic. Voorhees states "the assessors are told to assume that they are writing a report on the subject of the topic statement. If they would use any information contained in the document in the report, then the (entire) document should be marked relevant, otherwise it should be marked irrelevant" [39].

In the Cranfield paradigm, the output of an IR system can be evaluated using a variety of metrics. The choice of metrics depends on characteristics of the IR system, characteristics of the test collection, and the goals of the evaluation. When the output is a set of documents that match a Boolean query, the most commonly used metrics are *recall* and *precision*, where

$$recall = \frac{number\ of\ documents\ \ retrieved\ \ and\ relevant}{number\ of\ documents\ \ relevant} \tag{1}$$

and

$$precision = \frac{number\ of\ documents\ \ retrieved\ and\ relevant}{number\ of\ documents\ retrieved} \tag{2}$$

When the output is a ranked set of documents, as is usually the case for natural language queries (or any non Boolean query), metrics that assess the quality of ranking are more appropriate than set-based recall or precision. One method to evaluate ranked results is to calculate values for precision at standard levels of recall, plotting precision as a function of recall either with or without interpolation between

known values. Another method to evaluate ranked results is to calculate a single summary value that can be averaged over a set of queries to reflect system performance and to facilitate comparisons between systems. We briefly introduce five popular metrics: precision@X, Mean Average Precision, Reciprocal Rank, bpref, and Normalized Discounted Cumulated Gain (and the related cumulated-gain metrics).

Precision@X is the precision value for only the top *X* documents in a ranked output where *X* is a variable chosen by the evaluator, such as 1, 5, 10, or 20. In other words,

precision@X = num of top X documents that are relevant / X (3)

The advantages of precision@X (also referred to as precision at document cutoff values) are that it is easy to calculate, its interpretation is intuitive, and it reflects the well-known phenomenon that searchers rarely look beyond the first few pages of ranked search-engine output. The disadvantage of precision@X is that it does not average well across a set of queries because it fails to account for variability among queries with respect to the number of relevant documents. Suppose an IR system returns one relevant document among the top ten documents for each of two queries. And, suppose that the first query has no other relevant documents, but the second query has twenty relevant documents. The precision@10 is 0.1 for both queries, but common sense indicates that the system has performed much better for the first query than for the second query.

Average Precision (AP) is a measure of the "goodness" of the ranking for a single query and reflects both precision and recall for the entire list of ranked results. To calculate AP one first calculates the precision@r for the rank at which each relevant document *r* is returned, and then averages the values that are obtained. For relevant documents that are not returned in the query results, precision is 0. Therefore,

$$AP = (\sum_{r \in R} precision@r) / |R| \quad (4)$$

where R is the set of relevant documents for the query, *r* is a relevant document, precision@r is the precision achieved at the rank of *r*, and precision@r = 0 if *r* was not retrieved.[11] To illustrate average precision, consider a query for which four relevant documents are known to exist and the system returns three of them, ranked 1, 4, and 5. The average precision is calculated as (1/1 + 2/4 + 3/5 + 0)/4 = 0.525. After the first relevant document is retrieved, the precision is 1.0. After the third relevant document is retrieved, 3 of 5 documents are relevant, so the precision is 0.6. An ideal average precision is 1.0, meaning that all *n* relevant documents are retrieved and appear in the first *n* positions in the ranked list. *Mean average precision (MAP)* is the average of the AP values for a set of queries (such as the queries in a test collection). MAP is one of the most commonly used metrics in IR evaluation and has been found to be "stable and discriminating" in a study of IR evaluation measures [41]. A disadvantage is that

[11] This formula for AP is the one used by TREC. It is slightly different from *Average Precision at Seen Relevant Documents* [40], which ignores the failure to retrieve known relevant documents.

it lacks an analogue in real-life experience and therefore the values it yields are not intuitive to interpret.

Reciprocal Rank (RR) is the precision achieved when the first relevant document is returned, that is RR = $1/\text{rank}_d$ where rank_d is the rank at which the system returned the first relevant document. Although of limited usefulness for evaluating the full range of IR system capabilities, reciprocal rank can be useful if one is interested in how well the first relevant document is ranked. When there is only one relevant document for a query, reciprocal rank is equivalent to average precision.

The use of pooling means that relevance judgments are incomplete, which could affect the assessment of various experimental IR systems. The *bpref* metric was introduced as an alternative that does not rely on the assumption of complete relevance judgments [42] and has become popular for evaluating experiments using the TREC collections. Bpref is calculated as:

$$bpref = \frac{1}{R}\sum_r \left(1 - \frac{|\, n \text{ ranked higher than } r \,|}{\min(R, N)}\right) \tag{5}$$

where R is the number of judged relevant documents, N is the number of judged nonrelevant documents, r is a (judged) relevant retrieved document, and n is a member of the set of the first R (judged) nonrelevant retrieved documents. Bpref has been shown to correlate well with MAP when complete relevance judgments are available and to be more robust than MAP when relevance judgments are incomplete. However bpref does require that nonrelevant documents have as much chance of being

explicitly judged as relevant documents [42]. Although this assumption holds for the TREC methodology, it may not be valid in other types of experiments.

Normalized Discounted Cumulated Gain (nDCG) is one of four related metrics that were introduced by Järvelin and Kekäläinen for evaluating ranked output when relevance judgments are graded instead of binary [43]. The four cumulated gain metrics are most easily described together. For all four metrics, the initial step is to assign a value for each of the graded relevance scores (such as *0, 1, 2, 3* or *0, 1, 10, 100* for a scale with four levels of relevance). The simplest of the metrics is Cumulated Gain (CG). CG is calculated by summing the values for the relevance scores of each document in the order at which they were returned by the system:

$$CG[i] = \sum_{j=1}^{i} G[j] \tag{6}$$

where $CG[i]$ is the cumulated gain at the ith document and $G[j]$ is the value assigned to the relevance score given to the jth. One can write the CG values as a vector or plot them on a graph to compare the CG performance of two systems on the same query (or to compare average CG performance for a set of queries). One can also compare the CG values at any document cutoff value. Discounted Cumulated Gain (DCG) is similar to CG except that it also applies a discounting function to each document so that documents returned earlier (higher on a results list) are valued more than documents returned later (lower on the results list). Discounting reflects the assumption that relevant documents appearing earlier in the results are more valuable to the searcher than relevant documents appearing later. The discounting function uses a logarithmic function in which the logarithm base is a variable set by the

evaluator. The original formula for DCG (equation 7) only discounted documents retrieved at a rank lower than the value chosen for the logarithm base. Thus, if the logarithm base were 10, the gain values for the first nine documents would not be discounted, but the values for the tenth document, and for all subsequent documents, would be discounted. The original version of DCG is

$$DCG[i] = \begin{cases} CG[i] & \text{if } i < b \\ DCG[i-1] + \dfrac{G[i]}{\log_b i} & \text{if } i \geq b \end{cases} \tag{7}$$

where i is a document rank and b is a variable representing the logarithm base, which allows adjusting the degree of discounting applied to late arriving documents.

DCG was recently modified to discount all documents returned after the first document [44]. The modification (suggested by the author) was part of our development of a session-based metric for evaluating a sequence of queries, which is discussed in Chapter 8. The modified version results in a smoother accumulated gain as documents are returned as compared to the original version, which exhibits a transition from accumulating undiscounted gains from earlier documents to accumulating discounted gains from later documents. The newly modified version of DCG (equation 8) is:

$$DCG[i] = \sum_{j=1}^{i} \frac{G[j]}{(1 + \log_b i)} \tag{8}$$

Because there can be different numbers of relevant documents for different information needs, a metric that normalizes the results for each query can be useful.

By reflecting how closely each query result matches the best possible result, a normalized metric allows comparing results across multiple information needs. Both CG and DCG can be normalized by first constructing a vector containing the ideal results, then dividing the CG or DCG vectors by the ideal CG or ideal DCG vector to produce Normalized Cumulated Gain (nCG) and Normalized Discounted Cumulated Gain (nDCG) values, respectively. Figure 2.1 illustrates each of these four metrics, CG, DCG, nCG, and nDCG.

Suppose an IR system returns a list of ten documents that have the following relevance values (using a four point scale) for a query known to have two highly relevant documents, three partially relevant documents, and two marginally relevant documents.

Rank	Relevance Judgment	Relevance Score
1	Not relevant	0
2	Partially relevant	2
3	Highly relevant	3
4	Not relevant	0
5	Marginally relevant	1
6	Highly relevant	3
7	Not relevant	0
8	Partially relevant	2
9	Marginally relevant	1
10	Partially relevant	2

The vectors for the various metrics would be as follows (using 2 as the log base for discounting):
gain vector: <0, 2, 3, 0, 1, 3, 0, 2, 1, 2> ,
cumulated gain vector: <0, 2, 5, 5, 6, 9, 9, 11, 12, 14>,
discounted gain vector (original): < 0, 2, 3.90, 3.90, 4.33, 5.49, 5.49, 6.16, 6.47, 7.08>
discounted gain vector (new): <0, 1, 2.16, 2.16, 2.45, 3.3, 3.3, 3.8, 4.04, 4.5>
ideal vector: < 3, 3, 2, 2, 2, 1, 1, 0, 0, 0 >
ideal CG vector: <3, 6, 8, 10, 12, 13, 14, 14, 14, 14>
ideal DCG vector(original): <3, 6, 7.26, 8.26, 9.12, 9.51, 9.87, 9.87, 9.87, 9.87>
nCG vector: <0, 0.33, 0.62, 0.5, 0.5, 0.69, 0.64, 0.79, 0.86, 1.0>
nDCG vector (original): <0, 0.33, 0.54, 0.47, 0.47, 0.58, 0.56, 0.63, 0.66, 0.72>

Figure 2.1 Cumulated gain metrics

Many authors have argued for more realistic evaluations than those attainable using test collections and the Cranfield paradigm. Here we note a few examples of specific critiques and specific proposals to introduce more realism.

Using graded relevance judgments is a more realistic reflection of the user experience than using binary relevance judgments. Sormunen [45] reassessed a subset of TREC documents using a four-point relevance scale. Sormunen noted that about 50% of the documents assessed as relevant by TREC assessors were rated as only marginally relevant when reassessed by other assessors using a graded scale, calling into question how well TREC comparisons of IR systems would generalize to more realistic settings.

Saracevic [46] considered a broad spectrum of issues related to IR evaluation and argued for the importance of evaluating systems in interactive mode, not just in batch mode.[12] Saracevic also noted that both system-centered and user-centered evaluations are important.

Borlund [47] proposed a framework for evaluating interactive retrieval systems that seeks to retain some of the controls present in the Cranfield model while introducing human searchers into the evaluations. She emphasized the use of

---

[12] In TREC and other Cranfield paradigm evaluations, researchers typically submit a batch of queries, one query for each topic, to the IR system and then evaluate the system's search performance based on mean performance across the queries. (By search performance we mean the quality of document ranking, as opposed to speed or other system performance measures). Users, on the other hand, interact directly with a system, submitting queries and usually reacting to the results, possibly deciding to reformulate the query. Interactive evaluations introduce additional parameters that influence system search performance, such as the usability of the search interface and the step of translating an information need into a query that conforms to the query language of the IR system.

simulated work tasks to provide a more realistic information need to the human searcher, who then formulates and submits queries to a system. The work task descriptions also provide a basis for making relevance judgments that relate to a given situation instead of indicating only that a document is about the same topic as the information need.

Järvelin [48] and Ingwersen and Järvelin [49] described IR research and evaluation as occurring in two frameworks: (1) the system-oriented or laboratory model and (2) the cognitive framework, which takes a more user-oriented viewpoint. They criticized past emphasis on the laboratory model and argued for the cognitive framework, which provides a more holistic and contextual view of information retrieval research.

2.2. Documents and Subdocuments

In this dissertation, we use *document* to refer to the text content that is indexed and retrieved by an IR system. A document can be: the full text of an article, a section of an article, a book chapter, or other unit of text; an abstract of a longer piece of text; a bibliographic record that contains information (metadata) about a piece of text; or some combination of these elements. We use *subdocument* to refer to a subset of a document, without regard for how the subset is selected. A subdocument might or might not correspond to structural elements, such as sections, within the containing document. A subdocument also need not be contiguous text. Each semantic component instance corresponds to a subdocument.

Most IR research is based on returning a whole document, although matching documents to queries might be based on a more concise representation of the document, such as an abstract in a bibliographic database or a set of keywords. However, some areas of IR research that index and retrieve whole documents use subdocuments as part of the retrieval process or investigate tasks that occur primarily at the subdocument level. As illustration, we highlight a few examples of subdocument use that are related to semantic components.

2.2.1. XML

The eXtensible Markup Language (XML) is an example of a method for representing subdocuments within a containing document. XML provides an explicit mechanism for defining hierarchical elements within documents. XML-based retrieval systems can index the content of each element separately and can return elements at any level of the hierarchy, from a leaf node up to the root. XML is particularly useful for representing structural document organization, where XML elements represent structural constructs. This representation of both content and structure makes XML highly suited for structured document retrieval, which uses both content and structure to retrieve documents. Users can submit content-only or content-and-structure queries. The Initiative for the Evaluation of XML Retrieval (INEX) is a large-scale effort to study XML retrieval using a test collection and annual evaluation campaigns similar to the TREC conferences. Query results from XML retrieval might consist of elements at various levels of the hierarchy, allowing the user

to access whatever amount of content is desired. Reid and colleagues have studied the *best entry points* to structured (XML) documents, considering both users' browsing and querying behavior. They define a best entry point as "a document component from which the user can obtain optimal access, by browsing, to relevant document components" [50, 51].

XML represents the structure of documents. It can also be used to represent the semantic organization of documents, but the hierarchical nature of XML limits its usefulness for representing purely semantic organization because semantic content is not necessarily neatly organized in a hierarchical fashion. Authors often weave strands of content throughout a document, or may use a different hierarchy for organizing content than is used in other documents in a collection. XML could be a useful method for representing semantic components in documents if the documents are written so that they conform to an existing XML schema that represents a semantic component schema. If structural elements reflect the same organization as the content, then the issue of overlapping semantic component instances would not arise. Documents that were written before a semantic component schema existed could be represented with XML if the documents share a well-defined structure that would be useful for searching. In such cases, the semantic component schema could be created to correspond with the existing document structure. Documents created using a template, in which the structural elements correlate with useful semantic components, would be particularly amenable to an XML representation that reflects a semantic component schema. When semantic component schemas are developed for collections

of more heterogeneous documents, XML is unlikely to be suitable for representing semantic components because XML would not allow semantic component segments to overlap with each other (unless one semantic component instance is completely nested inside another semantic component instance).

2.2.2. Other Subdocument Manipulations

Several tasks that are related to information retrieval involve analyzing and retrieving text at subdocument granularity and are related to semantic component indexing. We briefly discuss five such tasks: content analysis, text segmentation, passage retrieval, novelty detection, and information extraction. These tasks differ with respect to the importance of detecting the location and boundaries of subdocument text, the importance of characterizing or labeling the content in the subdocument text, and the purpose of the task (that is, how the results of the task are used).

Content analysis [52] is arguably the task most closely related to semantic component indexing. Content analysis, frequently used in social science research, is the systematic evaluation of the content of various forms of communication. It typically involves coding (labeling) units of information within a message. Content analysis can also be applied to other information types, such as audio and video, not just to text. The coding scheme might be predefined or might be developed as part of the research. For example, a study of the effects of television on children might require coding the content of various television shows. While the underlying medium

can have logical units, such as words or video frames, coding generally results in segmenting the message into variable-length pieces corresponding to the analysis. Both the assigned code and the location of various coded segments, including the assigned boundaries, are important. When demarcation and labeling of segments using a defined coding scheme is applied to text, the task is almost identical to semantic component indexing. One difference between the two tasks is the purpose: Content analysis is a research technique whereas semantic component indexing is intended to enhance information retrieval. Another difference is the model. Semantic component indexing occurs in the context of a model containing document classes and semantic components whereas the coding scheme will vary across different research projects and might or might not involve classification of the document to be coded. Comparing two coding instances and comparing two instances of semantic component indexing (such as to establish the reliability of a coding scheme or a semantic component schema or to establish the reliability of a coder or an indexer) both involve comparing the labels assigned to text (either codes or semantic component names) and comparing the similarity of the locations that have been labeled. We explore this similarity in Chapter 6 when we develop evaluation techniques for semantic component indexing.

Text segmentation is the task of dividing text into sections based on changes in topic or subtopic. It has been studied in the context of several problems: dividing previously undifferentiated streams of text, such as concatenated news stories, into their components [53], possibly as part of a topic detection and tracking effort [54];

dividing documents into sections corresponding to subtopics to aid in information retrieval [55] or display of retrieval results [56]; and preprocessing text in a summarization system [57]. Text segmentation can be linear or hierarchical. Linear segmentation typically assigns each unit of text to exactly one contiguous segment. The task is to correctly find segment boundaries, and evaluations have been focused on measuring the correctness of automatically-placed boundaries. Semantic component indexing is similar in that we try to find sections of documents that pertain to specific aspects of the main topic. But, unlike most text segmentation tasks, the set of aspects of interest (semantic components) is defined in advance based on the document class. Semantic component instances within a single document can be discontiguous and also can overlap with other semantic component instances, unlike segments resulting from text segmentation tasks. A given unit of text can belong to zero, one, or many semantic component instances as opposed to just a single segment in text segmentation.

Using text segmentation for information retrieval is one example of a broader group of passage-retrieval techniques, in which documents are split into a set of passages (subdocuments) and similarity to the query is computed for each passage instead of for whole documents. Liu and Croft classify approaches to splitting documents into passages as structural, semantic, window-based, and arbitrary [58]. Semantic component instances can be considered a form of semantic passages (meaning that passages are defined by their semantic content), although not all document text is necessarily included in any of the semantic component instances. A

more significant difference between our approach and passage retrieval is that we propose to use information about semantic component instances to supplement, not replace, whole-document retrieval techniques.

Novelty detection is similar to text segmentation in that the goal is to find instances of different subtopics, but the focus of novelty detection is on the different subtopics, not on their locations within documents. The TREC novelty task focused on finding sentences that were both relevant to a topic and novel, given the sentences that have already been seen [59]. Semantic component indexing differs from novelty detection because the aspects of interest, semantic components, are defined in advance and because the locations of the semantic component instances are important.

Information extraction (IE) is a somewhat different subdocument-level task. Information extraction systems identify certain types of information in unstructured text, such as entities, facts, and events. IE systems then extract the information into databases or templates. IE can be part of a question answering system, a specialized form of information retrieval that returns a fact, an entity, or a short answer that contains the answer to the question. The segments extracted by IE systems are generally quite short. Cardie points out that IE is inherently domain-specific since systems typically identify domain-specific relations among entities in the text [60]. Some of the semantic components we have identified in Decision Notices, such as *Responsible Official* and *Date*, are discrete, fact-oriented bits of information that would be suitable for extraction. Instances of other semantic components, such as *Issues* in Decision Notices or *Management* in documents about Clinical Problems tend

to be more diffusely distributed in the text and less amenable to identification using IE pattern-matching techniques. (See Chapter 4 for a discussion of the Decision Notice and Clinical Problem document classes and the semantic components we identified in those document classes).

2.3. Genre

A number of authors, such as Crowston and Kwasnik [61], Rauber and Müller-Kögler [62], and Freund and colleagues [63], have suggested using document genre to improve information retrieval. The term *genre*, traditionally used to describe literary and artistic works, has also been used to describe categories of organizational communications [64], documents in digital libraries [62], and web pages [65], although there does not seem to be a precise and universally accepted definition of genre. Orlikowski and Yates describe genres of organization communication (such as business letters and annual reports) as being "characterized by a socially recognized communicative purpose and common aspects of form" [64]. Attempts to automatically classify document collections on the basis of genre [62, 66, 67] generally rely on identifying attributes, or facets, that can be used to create a genre-classification system. Documents are assigned to genres based on the values for those attributes. The document classes in our model are akin to genres. For some familiar genres, it is easy to suggest semantic components whereas for others it is not so easy. For example, recipes typically have ingredients and cooking instructions, but what about letters or emails? All three documents have an identifiable form and purpose.

The difference seems to be that recipes are both specific to a domain (cooking) and have a predictable topic type (a dish), but knowing that a document is a personal letter or an email gives us little clue about the types of information likely to be present.

Turner and colleagues [68] created a model in the public health domain in which genre was one component. They used content analysis and a study with expert users to identify key elements in public health gray literature[13] that could serve as document representations in a searchable database. The key elements consisted of metadata that could be automatically extracted using natural language processing and brief, automatically-generated summaries of particular kinds of information in the documents. Document type (such as newsletters, guidelines, and data sets) was just one of the key elements. Some of the other key elements in the proposed document representation, such as *description of the problem*, *description of the intervention*, and *target population*, are similar to semantic components but were not linked to particular document types.

2.4. Concept Relations in Information Retrieval

Complex information needs often include multiple concepts that are related in a specific way. For example, consider this question from a database of physician questions [69]. "What is the best antibiotic to use for subacute bacterial endocarditis

[13] Gray literature consists of documents, such as reports, meeting notes, and policy documents, that do not appear in peer-reviewed or commercial publications and are often not indexed in databases used by professionals for information retrieval.

prophylaxis in penicillin-allergic patients?" The question is not just about antibiotics, or prevention, or bacterial endocarditis, it is about preventing endocarditis (an infection of the lining of the heart) with antibiotics. Modern IR systems can retrieve documents based on a query that includes multiple concepts, but they do not restrict the retrieval to documents about a specific relationship between concepts. The query may include a word or a phrase that expresses the desired relationship between two concepts, but that query term is treated like any other query term, independent of the concepts it relates. Thus, an IR system can retrieve a document that contains the desired concepts, but that is not relevant because the concepts are not related in the document (they might appear in completely unrelated sections of text) or because they are related differently than in the user's information need. In a detailed failure analysis of an early IR system, Lancaster termed these precision failures "false coordination" and "incorrect term relationships," respectively [70].

Information needs can be modeled as relations, where a *relation* is an ordered pair of concepts or terms that are related to each other in a particular way and has a label that describes how the concepts are related (the *relationship*). For example, the relation *treats("penicillin", "endocarditis")* expresses the notion that penicillin is used to treat endocarditis. Here penicillin is the intervention and endocarditis is the condition being treated. The relationship is important because penicillin can be used both to prevent and to treat endocarditis.

Concept relations abound everywhere, not just in medicine. For example, a business analyst might be interested in companies and products. Companies

manufacture, sell, and buy products. Companies also buy and sell each other. Products can be used to manufacture other products, or products can be bought and sold together. Finding information about specific business relations might be facilitated by explicit matching of relations in query and text. The relation *buys("McDonalds","potatoes")* expresses more of the meaning of the question "How many tons of potatoes does McDonalds buy?" than the query *McDonalds buys potatoes*, which could also represent "Who buys potatoes at McDonalds?".

Current IR systems do not represent relations explicitly and may even interfere with retrieving documents that contain the relations. Relationships are often expressed as prepositional phrases or as verbs. In full-text indexing, prepositions often appear on stopword lists and are discarded. When a relationship is expressed by a commonly occurring verb, the verb will appear in the index but the verb may have little effect on retrieval. Controlled vocabularies usually include noun forms of terms that represent relationships, such as "treatment" or "etiology". Manual indexers can choose relationship terms if the relationship represents a main focus of the document, but including a relationship term as a keyword does not indicate which concepts participate in the relationship. In both full text and controlled vocabulary indexes, relationship terms are usually treated the same as the concept terms they relate, with no structure to represent the relation itself.

Users sometimes represent relations implicitly through coordination, the combining of terms that represent different concepts to represent a new or more complex concept. Terms can be combined by entering a multiword query, by using a

logical AND operator, or by including a phrase in the query (usually by enclosing the phrase in quotation marks) if phrases are allowed by the IR system's query language. IR systems that use full text indexing are typically *postcoordinate*, meaning that descriptors are simple terms that are combined when a search is processed instead of during indexing. The combination can either be set based (such as for the Boolean query "cat" AND "platypus") or can be implemented with ranked similarity (such as for the natural language query "cat platypus"). Phrase matching can be implemented by indexing phrases or by finding documents containing all the words in the phrase and then checking for adjacency and order.

Systems that use controlled vocabularies, such as the Medical Subject Headings (MeSH) [71] usually have some degree of *precoordination*. A precoordinated term is a multiword term, such as *Heart Surgery*, that describes a complex concept. Some precoordinated terms represent relations. For example, a specialization of the broader term *Endocarditis* is the term *Endocarditis, Bacterial* that represents *causes*("bacteria", "endocarditis"). In addition, indexers and searchers can use role indicators (also called qualifiers or subheadings) that are available in some controlled vocabularies to indicate a particular aspect of a topic. An article about complications of endocarditis could be indexed with *Endocarditis/complications*, a MeSH descriptor/qualifier pair [72] that represents a partial relation. In some cases, a pair of complementary qualifiers can be used to specify a full relation, such as using *Endocarditis/drug therapy* and *Penicillin/therapeutic use* to represent the relation

treats("Penicillin", "Endocarditis"). Lancaster [22] discusses other precoordinate indexing systems that also use role indicators.

One way to express queries is as a set of relations that should appear in retrieved documents. In previous work [73], we modeled relations in three types of queries to an IR system. First, when the user is interested in two concepts related in a particular way, we have a *full relation*. Second, if a user wants to know everything about a topic, we can represent the relationship and the other concept in the relation with variables in what we call an *open relation*. Third, when only one of the concepts is specified and the other can be represented as a variable, we have a *partial relation*. The term partial relation was coined by Khoo and Myaeng [74]. Queries can also be composed of combinations of full and partial relations. Figure 2.2 shows examples of these query types. The letters *X* and *Y* in Figure 2.2 represent variables, meaning that the slot can be filled by any concept or relationship.

Query	**Type**	**Relation**
endocarditis	open relation	X("endocarditis", Y) or X(Y, "endocarditis")
What causes endocarditis?	partial relation	causes(X, "endocarditis")
How does S *aureus* cause endocarditis?	full relation	causes("S *aureus*", "endocarditis")

Figure 2.2 Relational representations of three query types

Using relational representations of information needs, we studied whether relation matching could improve document ranking in a small test collection of medical

documents that we created [73]. From the ClinicalQuestions Collection, an online repository of almost 3000 questions collected by researchers studying the information needs of physicians in clinical settings [69], we chose 24 questions that were about either causation or treatment and that contained a single full or partial relation. We used an extended query language to express the relations in our queries to our experimental search system and developed collections of regular expressions to automatically identify instances of *treats* and *causes* relations in text documents. We achieved better ranking of search results by explicitly modeling the relations in the information need, and searching for documents containing the same relation, than when using more traditional methods to express relationships in the information need. The three comparison methods were: (1) using a single word in a natural language query (either "cause" or "treat"), (2) using the single word plus several synonyms, and (3) using a proximity operator to ensure that the words for the concepts and the words for the relationships occurred near each other in the text.

Although relations might be useful constructs to model queries and the expression of facts and ideas in text documents, identifying all of the relations that occur in a document seems impractical. In a given domain, certain relationships are particularly significant and will appear in a large number of queries. We were reasonably successful at identifying instances of the *causes* and *treats* relationships in a set of documents from a single source, but developing comprehensive sets of regular expressions to identify even a limited number of important relationships is challenging. The semantic components model is an alternative approach for using

relations in information needs to enhance information retrieval. Semantic components are not as fine-grained as regular expressions, and not all concepts mentioned in a semantic component instance participate in the relationship indicated by the semantic component label. Furthermore, semantic components can cover a broader information type than a single relationship. So, while using semantic components may be less specific than using regular expressions to capture concept relations, semantic components are also more flexible than regular expressions and may be more scalable than trying to identify the exact text that participates in a relation.

Others have used relation matching to try to improve retrieval performance, with modest results. For example, Khoo and colleagues [75] studied a single relationship they called *cause-effect* in a subset of documents (Wall Street Journal articles) and queries from the TREC test collections. They used a broader notion of the *causes* relation than we did in our work. For example, they constructed the relation *[The antitrust investigation]*$_{effect}$ *must be a result of [a complaint]*$_{cause}$ (their notation) for the information need statement "Document discusses a pending antitrust case." They used both partial matches and full matches to match a query relation to document relations. They did not find any improvement with relation matching except when they used a weighted combination of text word matching, relation matching, and proximity matching, and first optimized the weightings for each query individually in a training set.

Wendlandt and Driscoll [76] and Liu [77], investigated the use of thematic roles to enhance IR. These authors viewed terms in a sentence as taking on thematic roles,

such as *recipient* or *consequence*, based on the relationships expressed in the sentence. Their focus was on the role each concept plays in a relation, instead of on the relationship itself. As noted by Khoo and Myaeng [74], this approach can be viewed as partial relation matching.

Wendlandt and Driscoll [76] developed a lexicon of words, called triggers, that indicate the presence of a thematic role or an entity attribute, which they collectively referred to as categories. Since triggers are often general words, such as prepositions, the authors assigned probabilities that a trigger word indicates a particular category. Although they did not use regular expressions, their use of prepositions was similar to our use of prepositional phrases in regular expressions. Their two-step approach used word frequencies to retrieve a collection of possibly relevant documents and then used category information to rerank the initial *n* top-ranked documents.

Liu [77] built on the work of Wendlandt and Driscoll by incorporating thematic role categories into a Semantic Vector Space Model (SVSM) that integrated data about thematic role categories directly into the vector space model (VSM) for retrieval. In the SVSM, each term had both a weight, based on the term's frequency, and a case weight for each of the 33 possible thematic roles that represents the probability that the term will trigger that particular thematic role. Liu used more intensive text analysis than Wendlandt and Driscoll. He assigned case weights based on part of speech and syntactic type and assigned cases to prepositional phrases based on a manually-built *prepositional case realization* (PCR) dictionary.

These two studies were domain independent. The authors investigated a broad selection of general relationships that one might find in any document collection. Their thematic roles corresponded to partial relations; they did not explore a notion of full relations. Their results were lukewarm. Wendlandt and Driscoll found improvement in a very limited evaluation [76]. Liu found improvement only for longer queries that consisted of research paper abstracts [77].

Farradane used the "psychology of thinking" to identify nine relationships that he claimed were necessary and sufficient to express concept relations in all subject fields [78, 79]. He proposed a system of relational indexing that would use a controlled vocabulary plus these nine relationships, which were expressed using a shorthand of typographical symbols. Because some concepts participate in more than one relation, some of his indexing diagrams were quite complex, containing rings and other two-dimensional structures. Farradane envisioned that relational indexing of queries as well as documents would be performed by trained indexers (a more reasonable vision in 1980 than in 2008) but that the indexing products could be encoded and stored electronically and that queries could be matched to documents using computer systems. Although his work focused on full relations, he also alluded to the potential usefulness of allowing queries to contain partial relations. The human effort required for relational indexing limited evaluation to small document collections. In one evaluation, most terms had been used to index so few documents that Boolean keyword searches alone were quite effective [80]. Relational indexing was never commercially implemented and the potential usefulness of relational indexing for real-

life searching has never been fully investigated [81]. Like Wendlandt and Liu, Farradane focused on general relationships that could be applied in any domain instead of taking a domain-specific approach as we have with semantic components.

2.5. Facets and Faceted Browsing

Classification systems and controlled vocabularies for organizing information can be either enumerative or faceted [82]. Enumerative systems enumerate all the subjects of interest (which may be organized hierarchically) whereas faceted systems define the properties of interest. Classifying or indexing with a faceted system involves identifying the applicable facets and their values. The Art & Architecture Thesaurus [83] is an example of a faceted scheme that allows description of objects or concepts related to art using terms representing various facets, such as *physical attributes*, *styles and periods*, *agents* (people and organizations), and *materials*.

Faceted schemes can also be used to organize websites and to facilitate browsing access to information. Hearst and colleagues [84] use a representation called *hierarchical faceted categories* to create a user interface framework, Flamenco, that mixes elements of searching and browsing to enable exploration of search results. Flamenco uses metadata values for each facet to give searchers a browsing-like view of search results (such as viewing recipes by "Dish Type").

Development of a semantic component schema is not the same as performing facet analysis for a topic because the purpose is different. Traditional facet analysis involves such principles as exhaustivity and mutual exclusivity [85], which we do not

try to achieve. Some semantic components correspond quite naturally to facets. For example, *diagnosis* and *treatment* can be facets of *diseases*. However, semantic components can also contain information that might not be considered a topic facet, such as the scheduling instructions for a given hospital within a *practical information* component for documents about surgical operations. In some cases a semantic component schema might combine two or more concepts that are different facets of a topic into a single semantic component, such as combining *epidemiology* and *natural history* of a disease into *general information*. Because semantic components are intended to facilitate retrieval, not to describe the domain, knowing either the contents of a particular document collection or the frequently occurring information needs among users of the collection can lead to selecting semantic components with varying degrees of specificity to represent document content. Unlike Flamenco, in which labels are assigned from the faceted metadata categories, the "value" for a semantic component is the text that pertains to a semantic component. Our approach allows the semantic components for each document class to be chosen at varying levels of specificity. More general components encompass more text, provide more exhaustive indexing, and can cover a wider range of queries while more specific components can support greater precision. This flexibility makes semantic components applicable to a wide range of documents.

2.6. Discourse Models

Discourse analysis is a research area that typically studies both content and structure and that focuses on units of text (or other communication media) larger than a sentence. Modeling the structure of documents at the discourse level has been used as a way to analyze and use the semantic content of documents for various information tasks, including automatic generation of natural language text, document detection and information extraction, automated abstraction, and automated summarization as well as for document retrieval. We loosely group various discourse models into three categories: (1) models based on the rhetorical/argumentation function of text segments, (2) models based on the communicative function of the document type, and (3) models based on the semantic roles of domain-specific entities and relationships in the document. We briefly discuss a selection of research in this area to compare and illustrate some models in each of these categories.

Mann and Thompson [86] and Teufel and Moens [87] worked with models based on the rhetorical and argumentation function of text segments. Mann and Thompson [86] introduced Rhetorical Structure Theory (RST) as a way to analyze existing text and as a theoretical basis for the planning phase of automatic generation of large scale texts. For example, if the goal is to produce a message that will have a particular effect on the reader, RST is a way to build a rhetorical structure so that the message will have its intended effect. RST describes text using a tree structure, where each subtree is an instance of an RST schema. Each RST schema is defined by a relation (or sometimes two relations) between a nucleus (a span of text) and one or more

satellites (other text spans). Each relation describes a rhetorical move or relationship, such as *evidence*, *justify*, *motivation*, and *restatement*. The leaf nodes of the RST tree are usually independent clauses, resulting in a fine-grained description of the text. Teufel and Moens [87] developed an annotation schema for classifying sentences in scientific articles to build a human-annotated training set and to serve as the basis of an automated summarization system. Like Mann and Thompson, their schema was domain-independent, but it was intended only for scientific articles and consisted of seven nonhierarchical categories: *aim*, *textual*, *own*, *background*, *contrast*, *basis*, and *other*. The basic text unit is larger (sentence instead of clause) than in RST and the categories refer to the role of the sentence in the argument of the entire paper instead of the rhetorical role within a smaller section of adjacent text.

A more common approach is to develop models based on elements being communicated by various document types, also referred to by Paice [88] and by Liddy and colleagues [89] (both citing van Dijk), as the document's *superstructure*. We could also consider this approach to be genre-based. Liddy [90] worked with professional abstractors to develop a hierarchical schema for the discourse structure of empirical abstracts. The schema can be used to indicate the semantic roles (such as *subjects* and *data collection* under *methodology* and *significance of results* and *practical applications* under *conclusions*) of concepts appearing in the abstract text. She also analyzed a sample of abstracts that had been marked up using her representation to identify "clue-words" that could aid automatic analysis of abstracts and facilitate automatic filling of slots in the frame-like structure used to represent the

schema. She suggested possible applications for a discourse-level representation of abstracts, including information retrieval and automated extraction of information. In other work, Liddy and colleagues [89] used a discourse model for newspaper articles and tested an application that classified sentences into the various components of the model (such as *circumstance*, *consequence*, and *main event*). This application was a module in a larger system, called DR-LINK, for retrieving and extracting information related to specific topics. DR-LINK was developed and evaluated as part of the Defense Advanced Research Projects Agency (DARPA) TIPSTER Text program. When the researchers investigated in detail the 26 Topic Statements (information need descriptions) in the TIPSTER collection for which DR-LINK performed well, they found that in 12 cases the discourse-level data was useful. Those 12 Topic Statements included a requirement that the information about an entity match another dimension, such as a temporal relation, or provide information about an aspect of the entity, such as the impact of an event [91]. Purcell et al. [92] developed context models of three types of medical research articles that could be used to represent documents in a retrieval system. A context model is basically an outline of the types of information that appear in a particular type of document, such as *case presentation* and *case discussion* in a case report and *methods* and *results* in a clinical research article. The contexts they identified were closely tied to the document organization that is characteristic in each type of medical research article. Conrad and Dabney [93] developed a schema to describe judicial opinions that could support development of

new search tools. Their schema included components such as *concurring opinions*, *dissenting opinions*, *historical facts*, and *disputed issues*.

Paice took a similar, but more domain-specific and less structural, approach to identifying important concepts in research papers that would be useful for both abstracting and indexing [94]. First he used manual analysis to identify the important semantic roles played by concepts in the domain (crop agriculture) and identified characteristic "context patterns" that signaled their occurrence. One example of a context pattern was "effect of INFLUENCE on PROPERTY of/in SPECIES" where the capitalized words were important concepts in the domain. A computer program identified candidate strings to represent the concepts based on the occurrence of the strings in the context patterns. The candidate strings were weighted according to the strength of the evidence provided by the context patterns in which they occurred. Although framed as a discourse-level approach, the result is similar to identifying concepts by the relations they participate in. Paice also reviewed the literature on automated abstract construction and noted the difference between using domain-specific semantic schemas as frameworks for representing text content and using superstructures that describe discourse structure typical of certain document types, such as research papers and abstracts of research papers [88].

Not all discourse models fall neatly into our three categories. For example, we described the work of Conrad and Dabney [93] as an example of the superstructure approach because their components have a specific role in the document type (judicial opinions) in which they occur. However, their model also has features of the other

two categories. Their components are specific to the legal domain as well as to the document type and their model could be characterized as representing the structure of an argument. Above, we described the work of Turner and colleagues [68] as an example of using document genre, but their approach is mixed. They did not try to identify the superstructure of each genre, but they did identify discourse elements (such as such as *description of the problem* and *description of the intervention*) in addition to document types.

Semantic components can be considered a form of discourse analysis that is related to both the superstructure and the domain-specific approach. Our analyses begin with a specific collection of documents, and the nature of the document collection determines the nature of the discourse analysis. As will be discussed in Chapter 4, when analyzing documents from sundhed.dk, which did not have a well-defined, homogeneous superstructure, our schema reflected important domain-specific concepts and relationships. When analyzing natural resource management documents, which had a superstructure that is mandated by law, our schema reflected elements of the superstructure.

2.7. Summary

In this chapter we presented a brief introduction to information retrieval. We discussed both the basic functional components of IR systems and important issues related to evaluating IR systems. We then discussed some areas of research that are related to semantic components:

- analysis and use of text at a subdocument level
- classification of documents by genre
- identification of concept relations in documents and the use of relations to improve document retrieval
- identification and use of topic facets
- construction of discourse models of documents

We showed how the semantic components model uses ideas from each of these areas and briefly compared and contrasted semantic components to existing work.

Chapter 3 The Semantic Components Model

The semantic components model uses additional information about the semantic content of documents to match documents to information needs. Instead of providing a single semantic component schema that is applicable to all document collections, or to all documents in a particular domain, we take a divide-and-conquer approach. Each semantic component schema is tailored to a domain and to a particular document collection. A semantic component schema does not try to represent all the concepts or relationships in a domain (or in a collection). Instead, a semantic component schema provides a set of semantic components that are important to the users of a collection and that can help differentiate types of information needs. Semantic components provide information to supplement, not replace, existing retrieval methods.

Traditional IR systems using natural language queries or keyword queries primarily support topical requests. In other words, the IR system returns documents "about" the topic that is represented by the word(s) in a query. As was discussed in Chapter 2, natural language queries, and full text indexes, sometimes contain words that represent relationships or facets of concepts. Controlled vocabularies also contain some terms that represent facets of concepts. Some controlled vocabularies (such as the Art & Architecture Thesaurus) have an explicitly faceted structure. Such faceted vocabularies allow some representation of the relationships between concepts in a query. Structured queries that accept query terms restricted to specific metadata fields (such as *title*, *author*, and *publication date*) can represent additional, non-topical facets

of the information need. Semantic components can supplement natural language queries, full-text indexes, faceted vocabularies, and structured queries to provide an orthogonal method of representing semantic relationships in queries and documents.

This chapter describes the semantic components model in detail. We begin by describing a prototype indexing application in Section 3.1. We then provide a formal description of the semantic components model in Section 3.2. In Section 3.3 we discuss semantic components as a specialized form of superimposed information. In Section 3.4 we provide a detailed overview of our approach to investigating the feasibility and potential benefits of semantic components to enhance information retrieval. We summarize this chapter in Section 3.5.

3.1. Indexing Prototype

To use semantic components for retrieving documents, semantic component instances must be identified in documents, a process that we call *semantic component indexing*. To further clarify and illustrate what we mean by semantic component indexing, we describe a prototype indexing application that we implemented to demonstrate indexing feasibility and to automatically record indexing decisions. When building the prototype, our goal was to allow an indexer to record indexing decisions with a minimum of effort. The work of indexing should be the intellectual effort of understanding and analyzing the document, not coping with mechanisms to record data.

The prototype is written in Java and records data from each indexing instance in a text file. A configuration file specifies the semantic component schema for the document collection being indexed. This specification allows the application to offer the indexer a menu of document classes appropriate to the collection. The application interface uses two side-by-side panes. The left pane displays the document associated with the URL that is entered by the user. Initially the right pane displays a menu of document classes. Figure 3.1 shows the indexing prototype interface while a document class is being selected for a document (in Danish) that was used in the indexing study (described in Chapter 7). The upper panel in Figure 3.1 shows a screenshot of the entire interface. The lower panel shows the same screenshot, enlarged and cropped for better readability.

After the user selects a document class, the appropriate menu of semantic component labels for that class becomes available. Figure 3.2 shows the same document as Figure 3.1 while semantic components are being indexed by highlighting and right-clicking on a menu. The right pane displays a vertical series of smaller panes, one for each semantic component in the document class. To index a document, the indexer uses the mouse to highlight a segment of the text appearing in the left pane. Right-clicking the mouse causes a menu of semantic component labels to appear, each label in a different color. Clicking one of the labels causes the highlighting to change to the color associated with the chosen label and also copies the highlighted text into the small pane associated with the label on the right. The copied text retains the colored highlighting for easy visual identification of text belonging to

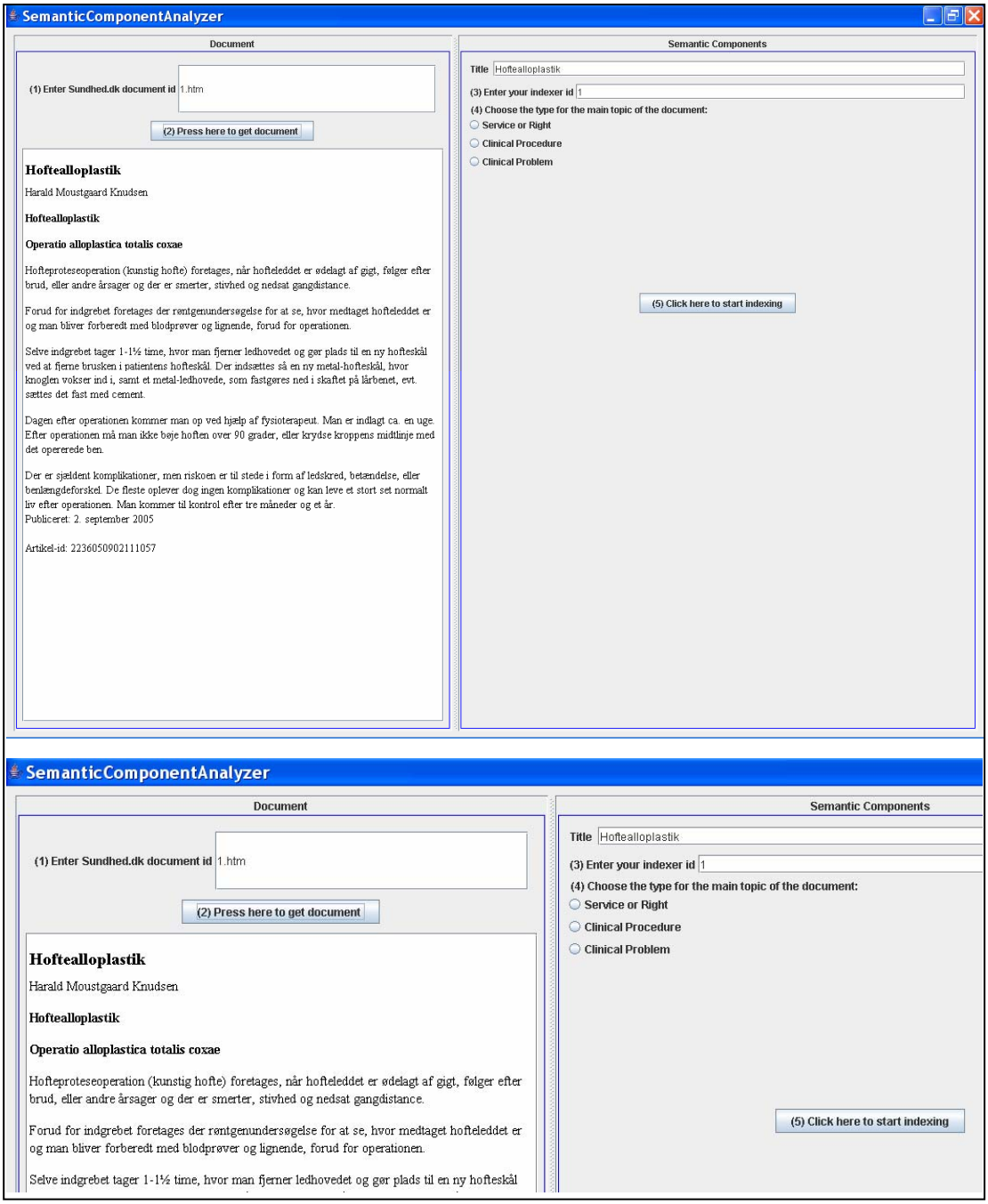


Figure 3.1 Screen shot of the prototype indexing application: choosing the document class. The lower panel is a magnification of the top part of the upper panel.

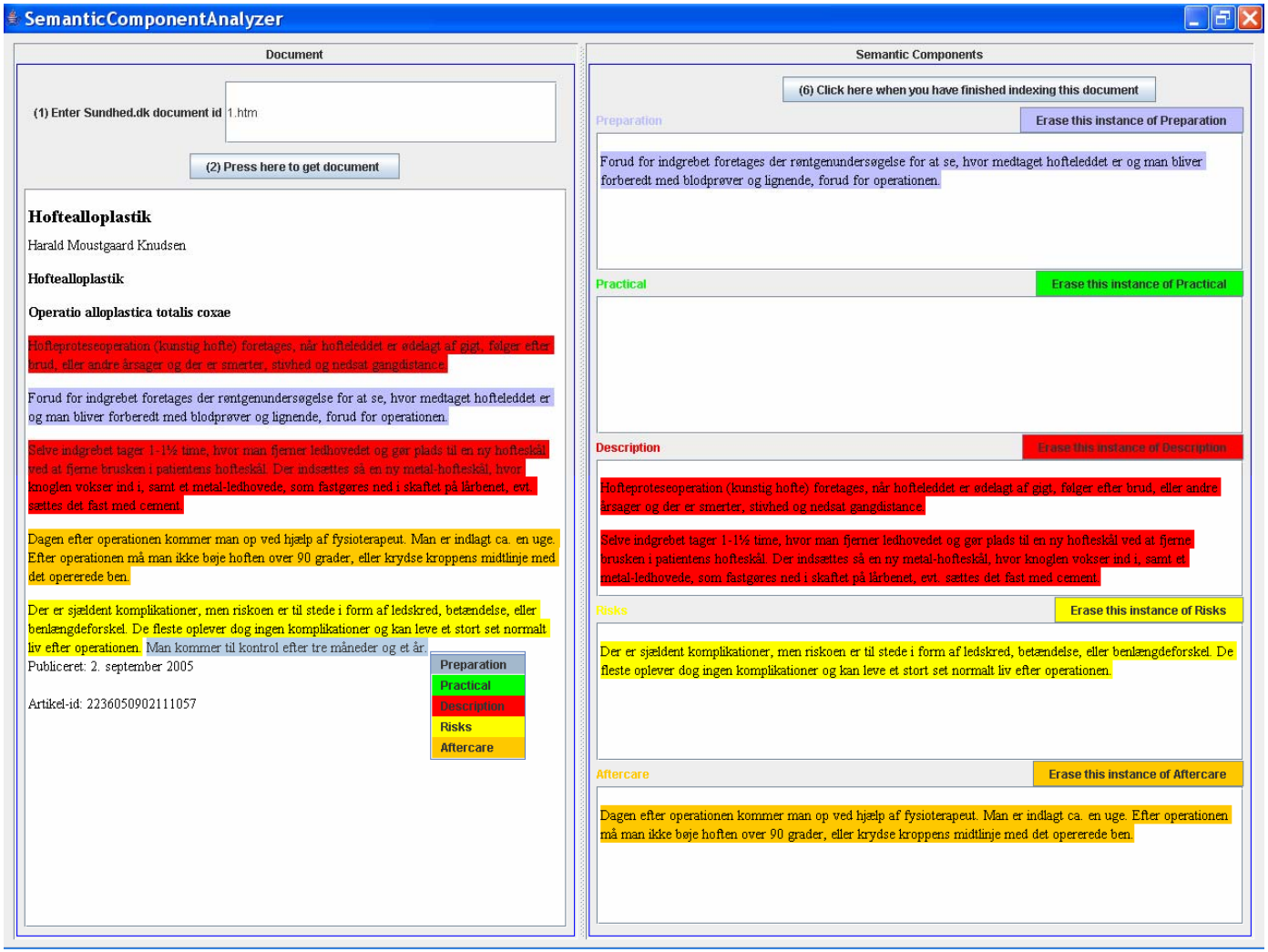


Figure 3.2 Screen shot of the prototype indexing application: marking semantic components

each semantic component. Additional text segments can be added to an existing component by highlighting a new segment of text and repeating the sequence of right-clicking and choosing from the displayed menu. Text that has already been highlighted and assigned to one semantic component can be re-highlighted and assigned to another semantic component as well. Errors can be undone by clicking a button to remove the text assigned to a semantic component. That now-empty component can then be re-indexed. The indexing application records the indexer's id, a timestamp, the document title and URL, the assigned document class, character offsets for the beginning and end of each segment, and the text in the semantic component instance. The application automatically adjusts the boundaries if overlapping or redundant segments are added to a given semantic component instance.

3.2. A Formal Description of the Semantic Components Model

A document collection that has been (at least partially) indexed in the semantic components framework, an **SCI** collection, is a triple (**D**, **M**, **I**) where:

- **D** is a nonempty set of documents.
- **M** is a nonempty set (possibly a singleton) of semantic component schemas (defined below).
- **I** is a nonempty set of indexing instances (defined below).

Definition: A **semantic component schema** *m* is a triple (**C**, **S**, **R**) where:

- **C** is a set of document classes.
- **S** is a set of semantic components.

- **R** is a relation that represents the relationship between a document class and the semantic components that represent the types of information found in documents that belong to the class. **R** is a subset of **C** × **S**, that is **R** = $\{(c, s) \mid c \in \mathbf{C}, s \in \mathbf{S}$, and s is a semantic component for document class $c\}$.

A document class is formed based on a concept that represents the shared similarity among the documents assigned to that class; the concept is represented by a symbol that serves to label the class. Similarly, a semantic component is a concept that is an important kind of information for a class of documents; the concept is represented by a symbol that serves as a label for the semantic component.

Definition: A text **unit** u_d is an occurrence of a minimum unit of text in document d.

Definition: The text **universe** for a document, $U_{\boldsymbol{d}}$, is the set of all text units u_d in document d.

For the purpose of discussion, we consider each occurrence of a character as a text unit. Although we focus on text, many documents that are primarily textual also contain images and graphics. We consider images and graphical elements to be units as well. Depending on the document format and level of preprocessing available to delimit units, an implementation could also treat words, sentences or other elements that can be individually selected as units.

We assume that a document is a linear sequence of text units. Two units are *consecutive* (or *contiguous*) if they occur in the same document and they are adjacent

(no other unit occurs between the units). The same character in a different position is a different text unit.

Definition: A **semantic component instance** $t_{s,d}$ is all of the text in document d that has been labeled with the symbol for semantic component s. $t_{s,d} = \{ u_d \in \mathbf{U}_d \mid g_s(u) = true\}$ where $g_s()$ is the characteristic function for semantic component s, i.e., it is a function that maps a unit to either *true* or *false* for s.

Definition: An **indexing instance** $i \in \mathbf{I}$ is a quadruple $(d, c, m, \{t_{s1,d}, t_{s2,d}, \ldots t_{sn,d}\})$ where

- $d \in \mathbf{D}$ is a document.
- $c \in \mathbf{C}$ is the document class assignment for document d, which can result from either automated classification or human decision, and c is a document class in schema m.
- $m \in \mathbf{M}$ is the semantic component schema used to index d in indexing instance i.
- $\{t_{s1,d}, t_{s2,d}, \ldots t_{sn,d}\}$ is a set of semantic component instances $t_{si,d}$ in document d and each $t_{si,d}$ is an instance of a semantic component s_i such that $(c, s_i) \in \mathbf{R}$ in semantic component schema m.

Implicit in the definition of an indexing instance is the notion of an indexing session that links the semantic component instances for a document that are parts of a single output by a particular indexer. An indexing instance can represent the intellectual effort of a human indexer or the results of an automated indexing application.

The set of indexing instances **I** in the SCI collection is the set of all indexing instances *i* in that collection. A given document can have multiple indexing instances that result from different indexing sessions and not every document in an SCI must have an indexing instance. Two indexing instances for a particular document may or may not have been assigned the same document class, and may or may not have been indexed using the same schema.

Definition: A **segment** of a semantic component instance *t* for document *d* is a maximal set of consecutive text units (and is therefore a subset of $\mathbf{U}_d$).

Note that two different segments of a semantic component instance *s* are non-overlapping because, by definition, a segment is a maximal set of contiguous units. In other words, $y \cap z = \emptyset$ for distinct segments *y* and *z* in *s*.

Note that these definitions do not require that a document belong to only one class or that it have only one indexing instance. Such a restriction can be imposed if desired; the experiments we describe allowed for at most one indexing instance per document.

3.3. Semantic Components as Superimposed Information

Superimposed information is information that is "placed over" existing information and can serve a variety of functions, such as organizing, linking, annotating, supplementing, or even just highlighting a subset of the information present in the existing base layer [95, 96]. Superimposed applications are applications that provide facilities to create and manipulate superimposed information. *Marks* are

encapsulated addresses that allow a superimposed application to reference a subset (not necessarily a proper subset) of the information in a base layer document.

Identification of semantic components in a document (semantic component indexing) supplies a layer of additional, superimposed, information about the text in the document. The semantic component labels are specialized annotations, describing the content of some portion of the document. Semantic component indexing can be implemented as a form of superimposed information, by using marks to indicate the location of each segment belonging to a semantic component instance and by using the semantic component name as an annotation for the segment of text. Two characteristics distinguish semantic component indexing from general superimposed information.

- Semantic component indexing has a specialized purpose and usage model. Superimposed information is any information that is superimposed on a base document, and can serve a variety of purposes. On the other hand, semantic component indexing serves a particular purpose. It provides an additional description of the content of a portion of the document (a subdocument) that can be used to enhance information retrieval. Instances of semantic component indexing can be stored in an index and used to filter or rank search results and to provide additional information about each document that is represented in a display of search results.
- Semantic component indexing conforms to a semantic component schema that specifies document classes and associated semantic components.

Superimposed information in general has no such requirements. Superimposed information does not require that marked text (the excerpt of text referenced by a mark) have a label or any other kind of annotation. If an annotation is present, its value is not restricted to a set of semantic component labels. Document classes are not part of the superimposed information model, although document classification can be implemented by annotating a mark that references the whole document with a label that represents the class name.

If semantic component indexing is implemented using marks to identify segments belonging to semantic component instances, a few additional mechanisms are needed that are not ordinarily present in superimposed applications. A superimposed application used to identify, manipulate, or store semantic component indexing must have mechanisms to:

- link segments that are part of the same semantic component instance
- link semantic component instances that are part of the same semantic component indexing instance
- ensure that semantic component indexing conforms to the semantic component schema

3.4. Studying the Feasibility and Potential Benefits of Semantic Components

Before semantic component indexing can be performed, a semantic component schema must be developed that is tailored to the particular document collection. For semantic component indexing to be useful for searching, users' information requests

must be expressed in such a way that semantic component information can be used to match documents to requests. And finally, the search system must be configured to use the additional information in a user's query that specifies one or more semantic components and possibly specifies search terms that should appear in the specified components. To fully assess the potential usefulness of semantic components there are four general questions to consider:

1. Can document classes and semantic components be identified for particular domain-specific document collections?
2. Can searchers express information needs using document classes and semantic components?
3. How easily can semantic components be identified in documents?
4. Are semantic components useful for retrieving documents?

We explored each of these four areas by investigating more specific versions of these questions in a particular domain and a particular setting in order to establish preliminary evidence regarding the feasibility and usefulness of semantic components for indexing and searching in domain-specific digital libraries. This section provides an extended overview of the remainder of the dissertation and describes how we addressed each of these areas in the research that is presented in Chapters 4 – 8. The details of our methodology appears in the individual chapters. Most of the research has been done in the healthcare domain, but we have also done some preliminary work investigating documents produced and used by natural resource managers, particularly documents mandated by the National Environmental Protection Act (NEPA). Table

3.1 summarizes our investigations in each of these four general areas. We elaborate on the information that is presented in the four rows of the table in the following four subsections.

3.4.1. Identifying Document Classes And Semantic Components In Document Collections

We addressed the first question by investigating two related issues:

- *What methods are available and useful for analyzing a document collection for the purpose of identifying useful document classes and semantic components?*
- *What lessons can be learned from preliminary efforts to analyze a collection and use the analysis?*

Table 3.1 Overview of methods to investigate the feasibility and usefulness of semantic components

General question	**Methods of investigation**
Can useful document classes and semantic components be identified for particular domain-specific document collections?	Preliminary analysis of two collections in two domains
	Use of one of the preliminary analyses plus an analysis of a second collection in the same domain in a study that mapped information needs to the schemas for two appropriate collections
	Use of analyses of sundhed.dk documents for indexing and searching studies
Can searchers express information needs using document classes and semantic components?	Mapping of an existing clinical questions taxonomy to the schemas for two medical document collections
How easily can semantic components be identified in documents?	User study of semantic component and keyword indexing
	Document indexing to support a searching study
Are semantic components useful for retrieving documents?	Interactive searching study

In Chapter 4 we describe our experiences analyzing three document collections in two different domains: medicine and public land management. We discuss two main

approaches to identifying a set of document classes: (1) analyzing a document collection based on document sampling, and (2) re-using existing document types. We also discuss two stages of the analysis with respect to semantic components: (1) an initial analysis of the collection itself, and (2) refinement based on the expected characteristics of information needs, search tasks, and users.

In the medical domain, our first experience consisted of an initial exploration of the documents available from the national Danish health portal, sundhed.dk. We subsequently used sundhed.dk documents for three experiments and each time we refined our schema for sundhed.dk. Chapter 4 reports on our experiences and the lessons we learned from iteratively developing and using analyses.

3.4.2. Expressing Information Needs With Semantic Components

To better understand how information needs can be expressed using a query language extended with semantic components, we mapped a published taxonomy of questions asked by primary care physicians [8, 9] to the document types and semantic components that we found in two collections of documents intended for physicians. The information needs represented in the taxonomy have already been abstracted into generic questions (such as "What is the cause of symptom X?"). We analyzed how well the generic questions in the taxonomy could be expressed using the semantic component schemas for the two document collections in order to answer the question:

- *What proportion of clinical questions and clinical question categories can be expressed using document classes and semantic components identified in two*

collections of documents that are intended to serve the clinical information needs of physicians?

We describe the study in detail and report the results in Chapter 5.

3.4.3. Indexing Semantic Components In Documents

The success of the semantic components model for enhancing domain-specific searching will be dependent on successful indexing. Successful indexing has two requirements:

- Indexing must be of high quality. This means that the indexing must faithfully apply to the individual documents the schema that has been developed for the document collection. (Here we assume that there is an existing schema that reflects the actual document types in the collection and their contents). Document classifications must consistently reflect the intent of the schema, and instances of the corresponding semantic components must be correctly identified in each document. If indexing does not adequately reflect the intended schema and the corresponding expectations of searchers, then using semantic components to express queries and retrieve documents is unlikely to be useful.
- Indexing must be feasible with respect to the time and intellectual effort required of indexers and with respect to expectations regarding the quality of indexing. Although it might ultimately be possible to automate (or semi-automate) indexing, we should first study the effects of manual indexing. We

assume that manual indexing will be of higher quality than automated indexing, because we also believe that manual indexing will better help us to understand the potential benefits and limitations of semantic components than would automated indexing of variable and unknown quality. Furthermore, in order to develop automated indexing systems it will be necessary to (1) produce a substantial volume of manual indexing in order to more thoroughly understand the requirements for an automated indexing system, and (2) have a reference standard for assessing the output of automated systems.

Semantic component indexing is intended to supplement other forms of indexing that support topical queries, typically full-text indexing or a combination of full-text and keyword indexing. We decided to compare semantic component indexing to keyword indexing for three reasons:

- Manual keyword indexing is the "gold standard" for supplementing full-text indexing and is still being used on a large scale in a variety of settings.
- There is a long history of keyword indexing, so its nature and limitations are well understood. It provides an established standard for comparing the time and intellectual effort required for semantic component indexing, and for comparing the quality of indexing.
- We are studying semantic component indexing in a setting (sundhed.dk, the national Danish health portal) where manual keyword indexing already exists. Sundhed.dk has already evinced a commitment for investing human resources into indexing to improve the quality of search results. Furthermore, we had

access to experienced indexers, allowing us to compare semantic component indexing to keyword indexing performed by indexers who provide indexing for an established, operational system.

Our investigation of semantic component indexing consisted of two parts, with the goal of assessing the accuracy, consistency, speed, and perceived difficulty of semantic component and keyword indexing. The first investigation compared manual semantic component indexing to manual keyword indexing. Each indexer indexed half of the documents using semantic components and half of the documents using keywords. The second investigation is an analysis of the time required for semantic component indexing of 371 documents (for use in our searching study) by seven indexers who used the prototype indexing application shown in Figure 3.1. We describe both of these investigations in Chapter 7.

Evaluating the accuracy and consistency of semantic component indexing requires appropriate metrics. Because semantic component indexing is new, we had to determine which metrics are appropriate for evaluating the indexing in our study. In Chapter 6 we develop an evaluation framework for semantic component indexing. We also include the evaluation of keyword indexing in the framework to facilitate interpreting the accuracy and consistency data for both kinds of indexing. Direct comparison of measurement data is not possible because the different characteristics of the two types of indexing require different units of measurement. We first analyze the nature of both indexing tasks and propose a set of criteria that describe the desirable properties of evaluation metrics for measuring and comparing the accuracy

and consistency of indexing. We also discuss related tasks and the metrics that are commonly used to assess the performance of those tasks. We then propose a framework of evaluation tasks and appropriate metrics for each evaluation. We use the evaluation framework to evaluate the results of the indexing study that we describe in Chapter 7.

3.4.4. Using Semantic Components For Retrieval

Evaluating the usefulness of the semantic component model for searching is arguably the most important piece of this work. Clearly, for the model to attain widespread use it needs to provide some benefit to the searcher. We were interested in knowing:

- *Does the use of semantic components facilitate more successful searching?*
- *Does the use of semantic components facilitate faster searching?*
- *Does the use of semantic components for searching result in better document ranking?*

Evaluation of the semantic components approach is challenging. Information retrieval research is often accomplished by using existing test collections that consist of a set of documents, a set of queries, and a set of relevance judgments that indicate which documents in the collection are relevant to each query. No existing test collection has the minimum requirements for a retrieval study that can evaluate semantic components: a defined semantic component schema appropriate to the document collection, queries expressed using semantic components, documents with

semantic component indexing, and appropriate relevance judgments. Not only would building such a test collection be time-consuming and expensive, but a fixed collection would never be entirely satisfactory for evaluating the potential usefulness of the semantic components model for information retrieval because using test collections neglects the role of the user. Test collections typically include one or more descriptions of the information need (often called the *topic*) in varying levels of detail that are most often used directly as the queries to the IR system. How well users can express information needs using semantic components is an important aspect of the semantic components model that needs to be evaluated. Doing such an evaluation requires the participation of study subjects from the target group for which the document collection is intended. If our intent is to leverage the knowledge of domain experts about the domain and about the organization of domain-specific documents, study subjects must be domain experts.

Because of these constraints, we chose to conduct an interactive searching study to evaluate the potential usefulness of semantic components. This allowed us to study not only whether semantic components could improve document ranking, but also to study how searchers used semantic components and whether the searchers found semantic components to be a sensible way to express queries.

To investigate these general questions, we conducted a searching study in which thirty Danish family practice physicians searched for documents using two search systems. Both search systems used the documents in sundhed.dk plus the existing keyword indexing and full text indexing for each document. One search system was a

basic system without semantic components and the other was a search system identical to the first except that it also provided the ability to use semantic components to specify the search and to match documents to queries. We collected both subjective and objective data, including data from questionnaires and from log files.

We evaluated search system performance in the study from two distinct perspectives: (1) the user perspective, using only a searcher's own relevance assessments for the documents returned by his or her queries; and (2) the system perspective, considering all relevant documents returned by a system, where relevance is determined by the reference standard regardless of user assessment. The two perspectives do not necessarily give the same results. Although good system performance may be necessary for good user performance, improved system performance does not guarantee improved user performance.

Hersh et al. [97], and Turpin and Hersh [98], reported experiments from the TREC Interactive Track comparing batch (system) and user retrieval evaluations. They used the description part of the interactive topics from previous years of the TREC test collection as queries and submitted the queries to an IR system in batch mode, as is commonly done in TREC experiments. They calculated retrieval results for two well-known methods for calculating the similarity between documents and queries, confirming that an improved method outperformed a baseline method. The researchers then created two IR systems with identical interfaces, one using the baseline method and one using the improved method for calculating similarity. They randomized the participants in the interactive experiments to use either the baseline or

the improved system. The participants were given new search tasks to perform interactively that consisted of identifying documents that contained instances of the answers to questions. After the interactive experiment, the researchers verified that the two systems had again performed differently in batch mode when the text of the topics (the description of the search problem that was given to the participants) were used as queries. Even when the queries issued by the participants were evaluated against the expert relevance judgments in the test collection, the improved system outperformed the baseline system. However, there was no significance difference between the task performance (either finding documents with instances of answers to questions or finding correct answers to questions) of the users who searched using the baseline system and the task performance of users who searched using the improved system.

Turpin and Scholer further investigated this performance disparity by having searchers find as many relevant documents as possible in five minutes for each of 50 queries from another TREC collection. Unknown to the users, the IR system returned hit lists with predetermined levels of MAP, from 55% to 95%, regardless of the query entered by the user. The researchers measured recall (number of documents the searcher indicated as being relevant and that were relevant according to the TREC relevance judgments / total number of relevant documents in TREC relevance judgments) and the time to find the first relevant document. These measures of performance had very little relationship to the known system performance as determined by MAP [99].

The papers that reported these studies expressed the experimental results as the performance of the human searching participants and measured the searchers' performance using a reference standard that consisted of relevance judgments (judging which document contained answers to the questions in the topics) by judges who did not participate in the experiment. Alternatively, the experiment can be viewed as an evaluation of the performance of the two systems using the relevance judgments made by the searchers. When the searchers identified documents as containing answers to the information needs in the topic descriptions, they were indicating that the documents were relevant. If we view the experiment as an evaluation of system performance, using the relevance judgments made by the users, we can conclude that the "improved" system does not outperform the baseline system. These experiments highlight the different results one can obtain when user behavior is incorporated into the experiment. Users do not always notice relevant documents and they do not always agree with a reference standard as to which documents are relevant. We believe that both system-oriented and user-oriented evaluation perspectives are valuable; we evaluated semantic components from both the system-oriented perspective and the user-oriented perspective.

Our experiments involve elements of two approaches to IR experiments recently analyzed by Järvelin [48]. He considered the frameworks, models and study designs characteristic of two general approaches in IR that he referred to as the "lab IR" approach and "Ingwersen's cognitive IR" approach. On the one hand, our experiment follows the lab IR tradition of comparing two systems under controlled conditions.

Although our experiments employed real users, we tried to control as many experimental variables as possible, such as by using fixed search scenarios and by developing a reference standard of judgments asserting which documents are relevant to each scenario. On the other hand, our experiment also has features more characteristic of the cognitive approach. We designed the scenarios to reflect realistic tasks for the searchers of interest (family physicians) and we used a much stricter (and more realistic) standard of relevance than the topical relevance used in most lab IR studies. Objects of interest in our study included the documents themselves (we studied the use of document classes and the semantic structures within the documents) and the information requests (we studied a novel extension to traditional query languages). In addition to analyzing search system performance using the reference standard of relevance judgments, we also looked at how well the system helped each user find documents that he or she thought were relevant (contained the information needed to satisfy the scenario).

Measuring and interpreting the results in an interactive searching study is, itself, challenging. Test collections have a single statement to represent each topic (although the statement can include representations at multiple levels of detail). IR studies using a test collection submit a single query per topic to the IR system and evaluate retrieval performance using metrics that assume a single query per topic. But in real life, searchers often submit multiple queries with different representations of an information need until they either find information that satisfies the need or until they give up. We refer to the collection of queries submitted while searching to satisfy a

particular information need as a *session*. Existing IR evaluation metrics are designed to evaluate the results of a single query, not the results of an interactive searching session. Previous interactive searching studies have tended to avoid this issue by using searching tasks that can be evaluated without considering the performance of individual queries that occur in a sequence. For example, the TREC Interactive Track used question-answering tasks that could be evaluated using the fraction of topics for which the correct answer was found (for factoid questions) or instance recall and instance precision (for questions requiring lists as answers) [100]. The number of queries issued, and the quality of document ranking for sequences of queries, were not included in the evaluation. We are interested in supporting search tasks in settings where searching time is limited, so we investigated semantic components from both a single-query perspective and a session-based perspective. Our efforts to evaluate semantic components from a session-based perspective exposed interesting research problems. We wanted to evaluate the performance of sequences of queries, combining information about document ranking with the iteration number of each query in the sequence. Surprisingly, no metrics existed for evaluating retrieval results from a session perspective. An unanticipated result of this study was a collaboration with Dr. Kalervo Järvelin to develop a new session-based metric [44] that is described in Chapter 8.

3.5. Summary

At the beginning of this chapter we provided an introduction to semantic component indexing. We then provided a formal description of the semantic components model and a description of how semantic component indexing is a specialized form of superimposed information. Lastly, we provided an overview of the research activities that comprise the major contributions of this dissertation. The descriptions of how we analyzed document collections to derive semantic component schemas (Chapter 4) and our work to map a taxonomy of clinical questions to two semantic component schemas (Chapter 5) serve as foundations for the indexing and searching studies (Chapters 7 and 8, respectively). The ability to describe document collections with semantic component schemas is a prerequisite for semantic component indexing and the ability to express information needs using semantic components is a prerequisite for using semantic components for searching. Finally, a prerequisite for studying semantic component indexing and comparing it to keyword indexing is an appropriate evaluation framework (Chapter 6).

Chapter 4 Developing Semantic Component Schemas

Developing a semantic component schema is the first step in using semantic components to find information in a particular document collection. Because our goal is to support the information searching activities of domain experts who have specialized knowledge and needs, we develop semantic component schemas that are tailored to a particular document collection and to the experts who use the collection. A given document collection might serve multiple user groups and a diverse set of tasks. In such a setting, having different schemas for different user groups or task types might be useful. So far, we have only studied describing a given document collection with a single schema. In this chapter we discuss our experiences with developing one semantic component schema per collection. In the final chapter, we consider how multiple schemas might be implemented and used, but we save their study for future work.

We discuss and illustrate two approaches to collection analysis and schema development through case studies in two domains, medicine and natural resource management. The first method is a bottom-up approach that focuses primarily on determining the kinds of documents and the kinds of information that are present in the document collection. The second method is a more top-down, domain-centered approach that focuses primarily on the known purposes for the documents and begins by identifying any existing document types or templates. Iterative refinement, based on knowledge of user characteristics and common work tasks, can be applied to the

initial product of either approach. Because we developed semantic component schemas to support specific experiments that we performed in the context of specific user groups (as described in Chapters 5, 7, and 8), we did not try to describe all the document classes that can be identified in the document collections we studied.

In the following sections we first describe some specific experiences with respect to developing semantic component schemas, then we discuss some topics related to defining and using semantic component schemas. We discuss our initial analyses of medical document collections and of documents related to natural resource management in Sections 4.1 and 4.2, respectively. In Section 4.3 we describe how we refined the schemas. In Section 4.4 we suggest that, in some cases, individual documents share properties of multiple document classes and that allowing membership in more than one document class might be appropriate. In Section 4.5 we compare semantic component schemas to other knowledge structures, we note characteristics of document collections that can facilitate creating schemas, and we discuss issues regarding evaluation of semantic component schemas. We summarize in Section 4.6.

4.1. Analyses Of Medical Document Collections

We developed semantic component schemas for two collections of medical documents. The first collection is the documents (written in Danish) that are hosted by sundhed.dk, the Danish health portal. Our semantic component schema for sundhed.dk has undergone several cycles of refinement in the course of using it for

three experiments. The second is a collection of documents written (in English) for healthcare professionals. In this section we describe the processes we used to develop the initial schemas for each collection. In Section 4.3 we discuss iteratively refining the sundhed.dk schema.

4.1.1. The Sundhed.dk Documents

When we began this research, the sundhed.dk collection consisted of nearly 22,000 documents about health, medicine, and the Danish healthcare system.[14] The operational portal uses several classification methods to aid in information retrieval. The first is a set of document types used to classify documents. Although sundhed.dk has templates for use by authors when they are preparing new documents, conformance to the templates is not required and appears to be uncommon. Furthermore, the document type labels that are used in document metadata (*informationskategori*) do not have a one-to-one correspondence with the labels (*informationstype*) offered to searchers in the advanced search interface to the portal. The informationstype menu in the searching interface allows the searcher to select a filter so that only documents of that type will be returned in the search.[15] Table 4.1 shows correspondences between the document types. We determined the correspondences shown in Table 4.1 empirically, by choosing a document type filter

---

[14] By July 2006 the collection had grown to almost 25,000 documents.
[15] We noted on December 17, 2007 that the *informationstype* filter was no longer available in the sundhed.dk advanced search interface.

for various searches and then examining the metadata tags in the documents that were returned by the search. Table 4.1 shows the Danish label followed by the English translation in parentheses. One of the labels, *forløbsbeskrivelse*, is a compound word that literally translates as "course description." These are documents written for healthcare professionals that provide a comprehensive description of a disease,

Table 4.1 Existing document types in sundhed.dk

Informationstype (Document types available as filters in the advanced search interface)	**Informationskategori** (Document types present in document metatags)
Forløbsbeskrivelser (course descriptions)	*Forløbsbeskrivelse (komplet)* (course description)
Generel information (general information)	*Information* (information)
Henvisningsvejledninger (referral guidelines)	*Henvisningsvejledning* (referral guideline)
Lægemidler (drugs)	*Behandling og anvendelse* (treatment and use)
	Præparat- og produktbeskrivelse (preparation and product description)
Nyheder (news)	*Nyhed* (news item)
Patientinformation (patient information)	*Patientinformation (komplet)* (complete patient information)
	Undersøgelse (examination/investigation)
	Behandling (treatment)
	Sygdomsbeskrivelse (disease/condition description)
Sundhed og forebyggelse *(health and prevention)*	*Sundhed og forebyggelse* (health and prevention)
	Sygdom (disease)
	Behandling (treatment)

including its natural history, how it should be diagnosed, and how it should be managed at various stages of severity and progression.

The second classification method used by sundhed.dk is the association of keywords with documents. Many, but not all, of the sundhed.dk documents have undergone manual keyword indexing. Indexers can choose any number of keywords from any of three controlled vocabularies. They can also choose "free" keywords, which can be any words or phrases that the indexer deems appropriate. The three controlled vocabularies are:

- *ICPC*: The International Classification of Primary Care (ICPC) is used primarily by family practitioners and other primary care providers to code health care encounters and contains about 700 terms [101]. Sundhed.dk uses a Danish translation of the international classification system.
- *ICD-10*: The International Classification of Diseases, 10th Revision (ICD-10) is used primarily to classify diagnoses and diseases and contains about 20,000 terms [102]. Sundhed.dk uses a Danish translation of ICD-10.
- *Almen thesaurus*: The Almen Thesaurus was created specifically by sundhed.dk to index content that is written for the general public (versus healthcare professionals) and contains about 1400 Danish terms. It is based on a classification system used in Danish public libraries.

For the initial analysis of the sundhed.dk documents, we chose to ignore the existing classifications and take a bottom-up approach to understanding what kinds of documents and what kinds of information are common in the collection. To do this,

we selected a sample of 72 documents using a modified random sampling approach. Sundhed.dk documents have unique identification numbers that are generated at the time the document is uploaded into the system. Uniqueness is ensured by generating the id using a combination of document characteristics, including the author's id and the time the document enters the system. Because the numbers are not generated sequentially, the ids in use are only a tiny subset of all possible valid ids. When we performed the initial collection analysis, we had neither copies of the documents nor a list of document ids. We therefore used document searches to select a random sample of documents. We also designed our methodology to ensure that our sample included documents intended for both health professionals and for the general public and that our sample included documents for all the regions of Denmark.

To select our sample, we executed 72 searches using the advanced search interface for the health portal. To sample documents for both health professionals and the general public, we designed the searches to ensure that at least 20 documents had been indexed using at least one term from ICPC and at least 20 documents had been indexed using at least one term from the Almen thesaurus. We obtained the remaining documents through the free-text search interface. Our sampling occurred in two stages. First, we sampled 42 documents, of which ten were indexed with an ICPC term, ten were indexed with an Almen thesaurus term, and 22 were chosen from free-text searches, independent of any indexing terms that might have been assigned. We used the first sample for a preliminary analysis. We then selected another 30 documents to supplement and validate our initial analysis, using the same

methodology to ensure that ten of the new documents had been indexed with an ICPC term, and ten had been indexed with an Almen thesaurus term. The final ten documents were found with free-text searches.

Selecting documents for the sample required two stages: (1) selecting a search term, and (2) selecting a link in the search result. In the advanced search interface, the user can choose to do a free-text search or to begin a search by browsing the top level categories of either ICPC or the Almen thesaurus. Selection of a top-level ICPC category yields a result containing all documents indexed with a child term of the category selected. The searcher can then either scan the result list, or search within the subset of documents just returned. In other words, selection of a top-level category acts as a filter. Selection of a top-level Almen thesaurus term offers a similar result, with the additional option of further narrowing the results by selecting a second-level category. To select the documents indexed with the two controlled vocabularies, we chose search terms (categories) by randomly selecting ten categories from the top level of each vocabulary. We selected the top-level terms by assigning numbers to each term and using a pseudo-random number generator to generate numbers in the appropriate range. For the free-text searches, we used common, non domain-specific search terms, such as the Danish words for *and*, *or*, *in*, and *it*, to avoid biasing the search to particular topics. (Unlike some search engines, the sundhed.dk search engine does not eliminate common or "stop" words from searches.) After either selecting a search term (top-level category) from one of the two controlled vocabularies or using one of the ten common words as free-text search terms, we had a

result list consisting of links to documents. The second stage was to select a link, which then provided access to a document. We selected among the list of links by using a pseudo-random number generating program to generate a number, *i*, between one and the total number of links in the result. We then downloaded the document for the *i*th link in the result list.

Healthcare in Denmark is largely organized by region and many of the sundhed.dk documents provide practical information that is applicable only to a particular region. For some information needs, only documents for a searcher's own region are useful, so the sundhed.dk advanced search interface provides the option of limiting a search to documents applicable to a particular region. We wanted to ensure a broad geographical representation of documents, so we randomly assigned some of the first 42 searches to be limited by region, independent of the search term used. For each of the 17 regions, we limited one search to documents from that region.[16] On the day of the study, searches limited to documents from three of the regions all yielded only the same single document, contributed by a national organization. We included that document in the sample. The searches that we limited to each of the other 14 regions all yielded multiple documents. We used the pseudo-random number generating program to select one of the search result documents as described above.

[16] Since this work was performed, the regional governments in Denmark have been reorganized. In January 2007 the 17 regions were consolidated into five regions.

After selecting the sample, we then read the documents.[17] We made a brief outline of each document to summarize its content and a created a preliminary list of the types of information present. We did not use a predetermined set of information types but, instead, iteratively defined and refined the types based on what we saw in the documents. Three classification axes emerged from the analysis:

- intended audience (health care professionals versus patients)
- domain orientation (about clinical issues versus about organizational or personnel issues)
- region specificity (useful primarily for a particular region versus having national applicability)

For most of the health documents, the intended audience was clearly either health professionals (such as physicians) or patients. Documents we judged as written for health professionals contained more technical medical terms, contained guidelines regarding patient or specimen management, and tended to be written in the passive voice (the patient *is discharged*, the test *is performed*). We further subclassified the documents for health professionals according to their primary focus:

- a clinical problem (such as a disease or symptom)
- a test or procedure (such as a laboratory test)

---

[17] All the document were written in Danish. After a month of intensive language study, the author of this dissertation was able to translate the documents with the help of two Danish-English dictionaries, one general and one medical.

Documents we judged as written for patients were more likely to contain lay terminology, to contain both technical and nontechnical terms (often listed as synonyms), to contain information about self care or when to contact a doctor, to contain information about what to expect during a clinical encounter, and to address the reader using "du," the familiar (as opposed to formal) word for *you* in Danish. We further subdivided these documents according to their primary focus:

- a clinical problem
- a test or procedure
- a health-maintenance or wellness activity (such as smoking-cessation services or the benefits of exercise)

For some documents, the intended audience was less clear, or was not differentiated. Such documents were more likely to be nonclinical or to address public health issues instead of the care of an individual.

We classified documents as *clinical* if they described such things as diseases, symptoms, laboratory tests, diagnostic procedures, and public health issues. Many of the nonclinical documents contained organizational information, such as information about personnel or about services offered by a particular hospital or department.

We classified documents as *region-specific* if they were unlikely to be useful outside the region of origin. We classified them as general (not region-specific) if at least some of the information could be useful outside the region. Organizational information was often limited to a single unit and had no applicability outside the associated region. Many of the clinical documents contained guidelines that were

Table 4.2 Semantic components for documents about a clinical problem written for a health professional audience

Component	Description
Evaluation	Tests or procedures for diagnosis, screening, monitoring, or staging
Therapy	Invasive procedures, medications or other therapies
Management guidelines	Guidelines for managing the clinical problem
Referral guidelines	Guidelines about when a patient should be referred to a specialist, what evaluation or therapy should be administered before referral, and what reports should accompany the referral
Prevention	Strategies for preventing the clinical problem or for preventing or minimizing associated complications
Risk factors	Factors that increase the risk of developing the clinical problem or increase the risk of complications
Prognosis	The expected course, or natural history, of the problem
Etiology	Information about causation
Associated conditions	Information about co-occurring, or complicating conditions, or common resulting conditions
Epidemiology	Populations statistics, such as incidence and prevalence

created for use within a single region but could be useful to patients or practitioners in other regions.

We performed additional analyses to develop sets of semantic components for two groups of documents from the Health Portal: (1) clinical documents about clinical problems that were written for health professionals (12 documents) and (2) clinical documents describing tests and procedures that were written for patients (four documents). We chose these two groups of documents because they are useful to family physicians but support different tasks: informing the physician and assisting with patient education. We were interested in semantic components that could help family physicians find useful documents so we did not develop document classes and semantic components to exhaustively describe the sundhed.dk document collection.

Documents in both classes had a readily identifiable primary focus, either a clinical problem or a test or procedure. Occasionally the focus was a group of related clinical problems or procedures. The semantic components we identified for each document class are shown in Tables 4.2 and 4.3, respectively. Structural elements to aid identification of components were present in some cases but often were absent. Interestingly, two types of information that commonly appear in other medical information sources were not present in our sample: information about diagnosis (we found guidelines for evaluation but not comprehensive discussions of differential diagnosis) and information about physiology and pathophysiology associated with clinical problems. We believe this absence is because sundhed.dk is intended to support patient care, especially in the setting where the family practitioner has a gatekeeper function with respect to referrals to medical specialists. Sundhed.dk is not intended to serve as a medical textbook with lengthy explanations regarding mechanisms of disease.

Table 4.3 Semantic components for documents written for patients about a clinical test or procedure

Component	**Description**
Preparation	How the patient is prepared, or should prepare himself, for a procedure (e.g. diet, shaving, medications)
Practical details	For example, where and when to report
Description of procedure	What will be done; what should the patient expect
Risks and complications	Possible risks, side effects, complications of the procedure
Aftercare	What to expect in terms of hospitalization, discharge, activities, follow up appointments
Where to direct questions	Who to contact if the patient has questions

We found that within each class, neither the presence (or absence) of a particular semantic component nor the location of a semantic component within a document was predictable. Instead, the semantic components present in a given document could be described as a subset drawn from a limited and predictable superset of semantic components. We also analyzed the relative sizes of the semantic components (calculated as the number of characters in each component) across the sample of 12 documents about clinical problems. Figure 4.1 shows the cumulative size of each of the semantic components in all 12 documents. This figure provides an interesting view of the types of information being communicated and, if the proportions hold across all the members of this class of documents, might reflect the purpose and information priorities of the portal itself. Figure 4.2 shows the relative proportion of each semantic component in the individual documents. Clearly, the documents are not at all uniform with respect to the types of information they contain.

4.1.2. The UpToDate® Documents

The second collection we analyzed is from UpToDate® [103], a commercially produced resource that is popular with physicians in the United States. In a previous study we used 100 topics (the term for documents in UpToDate®), mostly related to obstetrics and gynecology, to investigate concept relations for IR [73]. In the work reported here, we initially analyzed 20 of these documents, using a pseudo-random number generating program to select the sample, then added five additional documents to have a larger sample for one of the document classes.

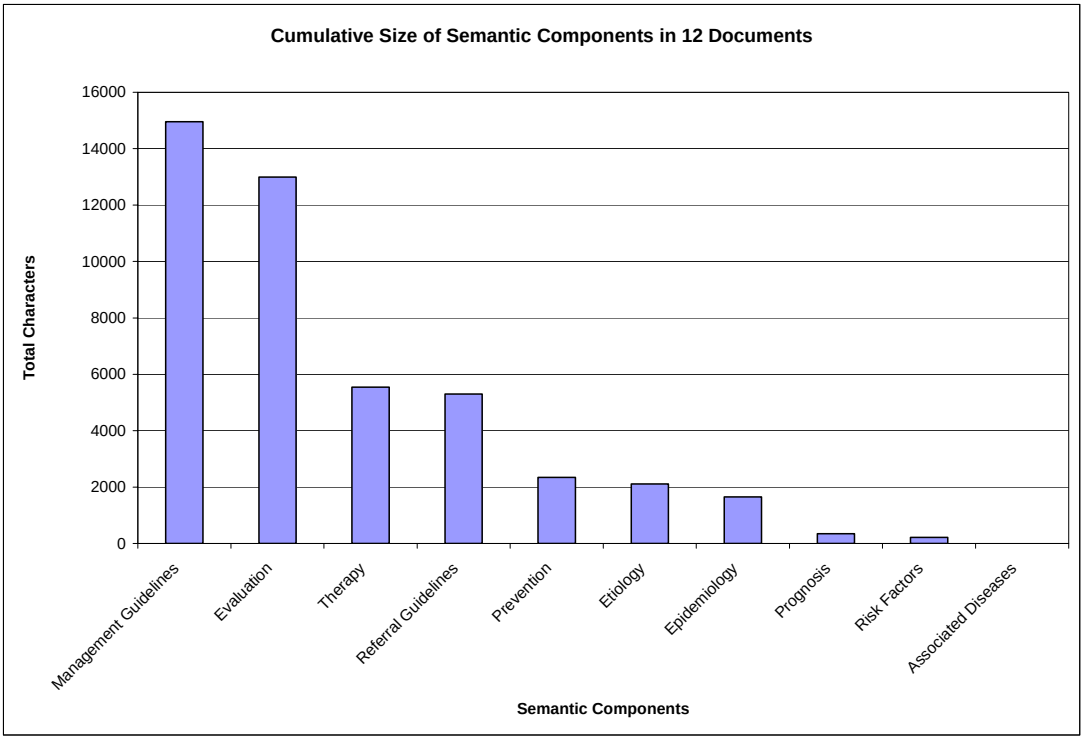


Figure 4.1 Relative size contribution of semantic components in documents about clinical problems

We used the same analytic procedure as for the sundhed.dk documents, although the UpToDate® documents were not as heterogeneous as the sundhed.dk documents. For all 25 documents, we outlined the information in each document using brief descriptions of information content in natural language. As with the sundhed.dk documents, we did not base these descriptions on any pre-existing list or classification because we wanted to describe the kinds of information that appear in these particular documents, not the medical domain in general. We also identified the primary focus of each document and derived document classes from the semantic types of the primary foci. We then considered the kinds of information we found in the documents assigned to each class. We constructed a list of semantic components to represent the commonly occurring information types (aspects or facets of the main topic) by

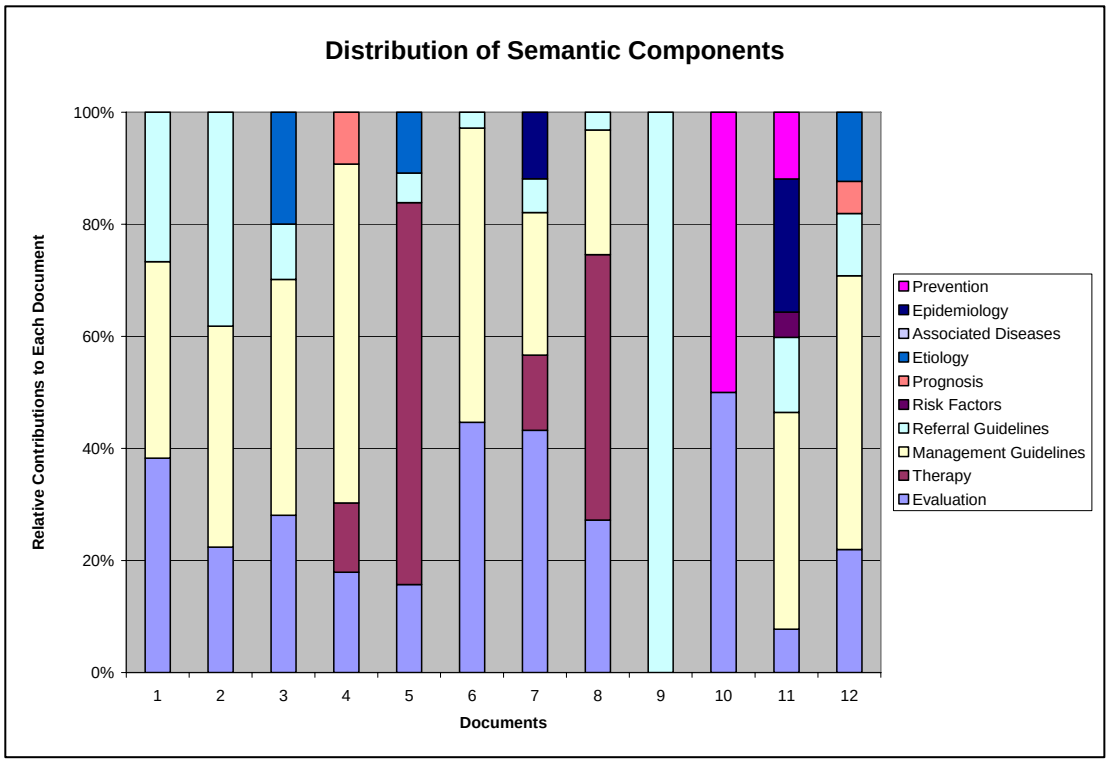


Figure 4.2 Proportion of text belonging to semantic components in individual documents about clinical problems

manually clustering similar natural language descriptions of the information types into groups and assigning meaningful labels to each group. Table 4.4 displays the results of our analysis of the UpToDate® document sample. Each of the three columns contains the semantic components for a different document class.

The semantic components we identified in the two medical document collections were similar, but not identical. Documents about medications were more common in UpToDate® than in sundhed.dk. Sundhed.dk may have fewer documents about drugs because sundhed.dk provides links to external resources that contain medication information instead of producing their own documents about drugs.

Table 4.4 Three document classes and their semantic components in UpToDate® documents

Clinical Problem	Test or Procedure	Medication
Epidemiology	General description	Pharmacologic category(s)
Diagnosis and workup	Indications	Administration
Pathogenesis	Pre-procedure preparation	Indications, use and effects
Treatment	Procedure	Use in pregnancy and other special conditions
Associated conditions	Complications, risks, and pitfalls	Adverse effects
Complications and sequelae	Post-procedure care	Contraindications and precautions
Prevention	Outcome	Interactions
Prognosis		Toxicity
		Cost information
		Alternatives
		Brand names
		Patient education and storage instructions

4.2. Leveraging Existing Document Types For Natural Resource Management

The National Environmental Protection Act (NEPA) mandates processes that public land managers must follow for all major projects. Among those processes are the creation of various types of documents to record the decisions and rationale at each stage of a decision process for an individual project. Most residents of the United States have heard of *Environmental Impact Statements*, which are required before major land development projects can be undertaken. Environmental Impact Statements are one of the document types specified by NEPA. A number of other document types are also mandated by NEPA. Documents of the same type, created for different projects, contain the same types of information, such as the *purpose and need* for a proposed project and the main *issues* that are considered when making a decision regarding the project.

We analyzed two NEPA document types, Environmental Analysis (EA) and Decision Notice (DN). These two types represent a common dichotomy among NEPA documents. The EAs are typically created by a multidisciplinary team, which analyzes a variety of issues. The EA provides a record of their analysis. The EA is written for, and used by, the responsible official who makes a land management decision. The DN, on the other hand, is typically written by a single person to document a decision for the Forest Service. The DN provides a synthesis of the alternatives considered in the EA and the rationale for choosing a course of action. The DN also communicates to the public what decision was made and why a particular alternative was chosen.

Documents of each type are available online at the websites for various national forests. We analyzed a random sample of EAs and DNs from the 140 EAs and 93 DNs available on the Web for the 13 national forests that are at least partially in the state of Oregon. We read each document, outlined the content, and made a list of the information types present. We did not use a predetermined set of information types but instead developed lists of what we found in the documents and then compared our findings to documentation available on Forest Service websites.

The NEPA documents have well-defined types, as we expected. The documents we analyzed followed the format as prescribed in instructions for preparing the documents [104] and in templates available online [105], although they varied in length and detail depending on the scope of the proposed project. Location elements

(such as *ranger district*, *national forest*, *county*, and *state*) appear in all the NEPA documents.

For the NEPA documents, the named project is the main focus of the document. Like the health documents, the focus is usually evident from the title of the document. The NEPA guidelines and document templates specify content elements that are essentially semantic components. The semantic components we identified for EAs and DNs, based on the documents we examined and on the templates, are shown in Table 4.5 and Table 4.6, respectively.[18]

Table 4.5 Initial semantic components for Environmental Analyses

Component	**Description**
Non-discrimination statement	Standard wording of USDA policy and who to contact if violation is suspected
Summary	Administrative unit making decision; rationale for action; description of proposed action and alternatives; rationale for decision
Introduction	For lengthier documents the document structure is described by table of contents or in paragraph form; history leading to proposal; purpose and need for action; proposed action; decision framework; public involvement; significant and non-significant issues and concerns
Alternatives	Description of each alternative, including mitigation; comparison of environmental costs and benefits of the alternatives
Environmental consequences	Description of the bio-physical, social and economic effects raised by public as issues or by the Interdisciplinary Team as concerns.
Interdisciplinary team	Names of the team members and their job titles
Agencies	Federal, state and local agencies contacted about this project
Tribes	Tribes contacted; sometimes including who, how and when
Individuals and groups	People and groups contacted, sometimes including when and purposes of contact
Appendices	Special consultation efforts and analysis efforts.

[18] The semantic component names, and their descriptions, were produced by Timothy Tolle.

Table 4.6 Initial semantic components for Decision Notices

Component	Description
Background	Purpose and need for action including issues, concerns, and direction
Decision	Actions to be implemented, including mitigation measures, site specific maps, drawings
Alternatives	Alternatives considered and environmental effects of each
Rationale	Reasons for decision
Mitigation measures	Measures to render effects less, if part of decision
Public involvement	People and groups contacted, specific means to provide public access to decision process
Findings required by other laws	Consistency with forest plan direction, Endangered Species Act (ESA), plans by other governments and agencies
Implementation date	Specific date or conditions which must be met in order to implement the action
Responsible official	Name and title of person who made decision
Admin. review	Whether or not a party can appeal the decision and how, if the party can appeal.
Contact person	Who and how to contact that person responsible for answering questions.

For the NEPA documents, with well-defined document types and instructions about what types of information must appear in the documents, using existing document types and information types to construct the initial semantic component schema was a useful and efficient approach. However, even for the NEPA documents, which were easily described with a semantic component schema that parallels the recommended document template, the semantic component schema benefited from a refinement based on domain knowledge about common information tasks and about the documents themselves.

4.3. Iterative Refinement Of Initial Schemas

We refined our initial semantic component schemas for the sundhed.dk and NEPA documents after considering how the semantic components were likely to be useful for realistic searching tasks. First, we used the contents of the documents themselves to indicate the kinds of information tasks a semantic component schema should support. For example, Figure 4.1 suggests that sundhed.dk probably contains more information about how to evaluate and manage patients, and how to refer patients to specialists, than information about risk factors and prognosis. Either sundhed.dk users are less likely to want information about risk factors and prognosis, and therefore have not requested more information of those types, or they are likely to have become accustomed to using other resources to answer questions about risk factors and prognosis. In either case, *risk factor* and *prognosis* semantic components are less likely to be useful for searching than *evaluation* and *management guidelines* semantic components. For the NEPA documents, some information types that are mandated by law contain "boilerplate" language and are identical (or nearly identical) in all documents belonging to the class. For example, DNs must contain information about administrative review but the content is stereotypical and similar across documents. The words within *administrative review* instances are unlikely to discriminate one DN from other DNs and searchers are unlikely to want to search within an *administrative review* semantic component.

Next, we consulted domain experts about how users were likely to use semantic components. After discussions with two physician users of sundhed.dk and with three

employees, who are involved in the indexing and editorial processes and who have extensive contact with users, we concluded that a simple schema, with only a few semantic components, would be more useful than a larger set of semantic components. We also confirmed our impression regarding the importance of referral guidelines for Danish family physicians. Our consultant from the forestry domain, Dr. Tolle, emphasized to us the importance of particular information elements in the NEPA documents, such as the specific project that a document relates to and the issues being analyzed.

Table 4.7 shows a refined version of the semantic components for sundhed.dk documents about clinical problems that we used in the indexing study described in Chapter 7. We consolidated *treatment* and *management guidelines* into a single *management* component. We also consolidated the semantic components that we believed would be less useful into a single semantic component, *about*.

Tables 4.8 and 4.9 show revised semantic components for DNs and EAs, respectively. In Table 4.8 we show the original and revised semantic component sets for DNs side-by-side for easier comparison. *Project name* and *location* are new semantic components. Because its anticipated usefulness is so high, *issues* has been made a distinct semantic component instead of including it in *background*. *Purpose*

Table 4.7 Document classes and semantic components used in the indexing study

Document Type	**Semantic Components**
Documents about a Clinical Problem or Condition	*Evaluation:* How to diagnose or evaluate the problem
	Management: How to treat, manage or control the problem
	Referral: How to refer a patient with the problem to a specialist or special service
	About: About the problem

Table 4.8 Semantic components for Decision Notices, initial (left) and revised (right)

Component	Description
Background	Purpose and need for action including issues, concerns, and direction
Decision	Actions to be implemented, including mitigation measures, site specific maps, drawings
Alternatives	Alternatives considered and environmental effects of each
Rationale	Reasons for decision
Mitigation measures	Measures to render effects less, if part of decision
Public involvement	People and groups contacted, specific means to provide public access to decision process
Findings required by other laws	Consistency with forest plan direction, ESA, plans by other governments and agencies
Implementation date	Specific date or conditions which must be met in order to implement the action
Responsible official	Name and title of person who made decision
Admin. review	Whether or not a party can appeal the decision and how, if the party can appeal.
Contact person	Who and how to contact that person responsible for answering questions.

Component	Description
Project name	Official name of proposed project
Location	Location of proposed project
Purpose and need	Purpose and need for action
Decision	Actions to be implemented
Rationale	Reasons for decision
Mitigation measures	Measures to render effects less, if part of decision
Issues	Significant issues considered in making the decision
Public involvement	People and groups contacted, specific means to provide public access to decision process
Date	Date the decision notice was signed
Responsible official	Name and title of person who made the decision

and need has both a narrower specification and a more descriptive name than *background* where it was formerly subsumed. *Findings required by other laws* and *administrative review* have been eliminated because they contain stereotypical text and are unlikely to be useful.

Table 4.9 Revised semantic components for Environmental Analyses

Component	Description
Administrative unit	The administrative unit responsible for making the decision
Year	Year the EA is completed
Project name	Official name of proposed project
Purpose and need	Purpose and need for action
Issues	Significant issues considered in making the decision
Proposed action	Description of the proposed action, such as activities, monitoring, maps, and mitigation
Decision framework	How the decision was made
Public involvement	People and groups contacted, specific means to provide public access to decision process

Although we did not formally assess the semantic component schemas for NEPA documents, three members of the research team performed semantic component indexing for seven DN documents using the revised semantic component schema. Our impression was that the schema was appropriate for the documents and easy to understand. Some semantic components were easy to identify, especially when their location corresponded to structural elements (Section headings) with the same, or similar, names. Other components, such as *issues*, appeared in multiple locations in the documents.

We encountered some interesting issues in the indexing study that led to yet another refinement of the semantic component schema for sundhed.dk before the searching study. Table 4.10 shows the schema for three document classes as the schema appeared in the indexing study and in the searching study. (The complete schema that was used in the searching study appears in Table 8.1). Although we provided the indexers with descriptions and examples of each document class, and

Table 4.10 Document classes and semantic components used in the two user studies

Indexing study		**Searching study**	
Document class name	**Semantic Components**	**Document class name**	**Semantic Components**
Clinical Problem	*Evaluation:* How to diagnose or evaluate the problem	*Klinisk problem* (Clinical problem)	*Diagnosticering* (diagnosis, evaluation)
	Management: How to treat, manage or control the problem		*Behandling* (treatment)
	Referral: How to refer a patient with the problem to a specialist or special service		*Henvisning* (referral)
	About: About the problem		*Generel information* (general information)
Procedure	*Preparation:* How to prepare for the procedure	*Klinisk Metode* (Clinical method)	
	Practical: Practical details		*Praktisk information* (practical information)
	Description: Description of the procedure		*Generel information* (general information)
	Risks: Risks of the procedure		*Risici* (risks)
	Aftercare: What to expect after the procedure		*Efterbehandling* (aftercare)
			Henvisning (referral)
			Forventet resultat (expected results)
Services	*Service or right:* Information about the service or right	*Services* (services)	*Generel information* (general information)
	Inclusion criteria: The indication or conditions that the patient should fulfill to get the service		
	Sequence: the course of events, the sequence of actions		
			Praktisk information (practical information)
			Henvisning (referral)

also of each semantic component for a given class, we nevertheless encountered some confusion about what kinds of documents belong in each class. The information that

was given to the indexing study participants about the *clinical problem* class and its semantic components is shown in Figures 4.3 and 4.4, respectively. It appeared that, despite the explanations and examples, the indexers tended to interpret the "meaning" of the document classes based on the class names, and that at least some of the confusion was related to terminology. In Chapter 7 we discuss in some detail why the name "procedure" might have caused problems in the indexing study. Briefly, both translation (between English and Danish) and different word senses (common usage versus medical jargon) might have contributed to confusion. For the searching study, we supplied Danish versions of both the semantic component schema and the accompanying descriptions and examples to the participants. We also used the name "clinical method" instead of "procedure." In addition, we tried to reduce the cognitive load on the searchers by using the same name for similar types of information in different classes, such as "general information", "practical information", and "referral". We did not study whether such use of the same names actually helped the searchers. It is possible that using the same name for information that might have subtle differences is actually confusing. Clearly the names for document classes and

Document Type	**Short Name**	**Description**
Documents about a Clinical Problem or Condition	Clinical problem	Documents that are primarily about a particular clinical problem such as a disease, a symptom, or other clinical condition. Examples: - a normal condition, such as pregnancy - an abnormal condition, such as malnutrition or injury - a disease, such as diabetes - a group of related diseases or problems, such as knee injuries (could include information about several specific injuries) - a symptom, such as chest pain

Figure 4.3 Information about the Clinical Problem class, as supplied to indexing study participants

Documents about a Clinical Problem or Condition	
Name	**Description**
Evaluation	**How to diagnose or evaluate the problem.**
	Information about how to evaluate a patient who has, or might have, the clinical problem. Examples: - how to diagnose the disease - how to determine its severity or clinical stage - the differential diagnosis of a symptom (what diseases could cause this symptom) - what screening tests are appropriate - what tests should be performed in patients who have this problem.
Management	**How to manage or control the problem.**
	Information about how to treat or manage a patient who has the clinical problem. Examples: - formal disease management guidelines - how to prevent complications - how to reduce the severity or impact of the disease on the patient - how to monitor progression of a disease - recommended diet, education, or counseling - what medications or procedures are appropriate - what doses of a medications to give
Referral	**How to refer a patient with the problem to a specialist or special service.**
	Information about how and when the family practitioner should refer a patient for specialist care. Examples: - criteria for referral (such as severity of disease, presence of certain complications) - how to make a referral (what number to call, where to mail documents) - what tests to do before the referral - what records to send to the specialist or special clinic
About	**About the problem.**
	General information about the condition, not necessarily for care of a particular patient. Examples: - natural history of a disease if not treated - the usual clinical course of patients with this problem - population statistics about how frequently the problem occurs - common co-occurring conditions or complications of the problem - etiology (causation) of the disease or condition.

Figure 4.4 Information supplied to the indexing study participants about semantic components for the Clinical Problems document class

semantic components are important, but we do not yet know what the best naming strategy is.

4.4. Multiple Schemas and Multiple Indexing Instances

We have iteratively refined some of the schemas as described above, but during any particular use of the schema, for either indexing or searching, we have allowed a

document collection to have only a single schema. Similarly, for the mapping study (Chapter 5) and the searching study (Chapter 8) we assumed that each document would have at most a single indexing instance. When we formalized the semantic components model (Chapter 3) we ensured that the definitions do not preclude multiple schemas and multiple indexing instances. We did so because we can imagine that a given document collection might be useful for a diverse set of information searching tasks or for a diverse set of users. It is possible that describing a document collection with more than one schema might be more useful than creating a single schema for all searches. Furthermore, it is possible that some documents have elements of more than one document class, and should be indexed accordingly, with multiple class labels and with instances of semantic components from multiple classes.

We have not yet investigated allowing multiple schemas and multiple indexing instances. We discuss some possibilities for future work along such lines in Chapter 9. For now, we offer one example to illustrate a document that might benefit from being indexed as a member of two document classes. One of the documents that was indexed for the searching study was titled "B-vitaminer: Behandling ved mangelsygdomme" (B-vitamins: Treatment of deficiency conditions). The document has several sections that each address who should have extra amounts of a particular B vitamin. Each section has two or three paragraphs that discuss the conditions resulting from deficiency of that vitamin (such as beriberi as a result of vitamin B-1 deficiency), common causes of such deficiency (such as chronic alcoholism), symptoms (such as encephalopathy), and treatment with supplemental vitamins. Is this a document about

clinical conditions (vitamin deficiency syndromes)? The document has information about evaluation and treatment. Or, is it a document about medications (B-vitamins)? The document has information about indications and dosage. One of the scenarios in the searching study concerns giving supplemental folic acid (a B vitamin) to pregnant women. Searches for that scenario included queries that used semantic components from the clinical problem class and queries that used semantic components from the medication class. Allowing the document to have two indexing instances, one that considered the document to be about clinical problems and one that considered the document to be about medications, might have been useful to the searchers.

4.5. Discussion

Semantic component schemas share features with other knowledge organization structures. Document class names can be thought of as keywords that describe something about the document. Assigning keywords from controlled vocabularies is a method of classifying documents that allows documents to belong to multiple classes simultaneously (see Chapter 6). Although we define semantic component indexing as associating semantic component names with segments of documents (subdocuments), the semantic component names could be treated as keyword assignments that describe some aspect of document content. Semantic component schemas can be thought of as small hierarchical controlled vocabularies, although the vocabulary terms have somewhat different uses in the semantic component model than simply indicating what the document is about. A semantic component schema is not a thesaurus; it lacks

the broader term/narrower term, synonymy, and related term relationships that are characteristic of thesauri [106].

A semantic component schema also is not an ontology, although it shares some features of ontologies (in the computer science sense, not the philosophical sense, of the word). Like an ontology, it provides some abstractions (document classes and semantic components) for representing a domain. However, a semantic component schema does not represent the entire domain, only a view of the domain as it is represented in a particular document collection. The domain representation is impoverished, representing only a small subset of relationships in the domain. It represents only those relationships that the schema creator deemed as being both sufficiently important to searchers and sufficiently well-represented in the document collection. Furthermore, relationships are represented by the schema in an imprecise and indirect fashion compared to an ontology. The presence of a *treatment* semantic component in a class of documents about diseases suggests that the domain represented by the collection has an abstract relationship *treats (X, Disease)* where *X* is a variable. Concrete relationships in the domain are represented using a combination of words in document text, semantic component instances, document classifications, and human inferences about the text. For example, the relationship *treats (penicillin, pneumonia)* could be represented by the word "penicillin" appearing in a *treatment* semantic component of a document about a clinical problem. That the problem being treated is pneumonia is represented by words in the text (and possibly the document title) that indicate that the document is about pneumonia. However, the

final determination that the document actually asserts that penicillin treats pneumonia requires human interpretation of the text.

Semantic component schemas are a type of discourse model for the classes of documents in the collection being described. Based on our experiences, we identified two characteristics of document collections that facilitate creation of a semantic component schema and that are likely to contribute to the usefulness of semantic components for searching. These two characteristics correspond to two of the approaches to discourse models that we identified in Chapter 2, the domain-specific approach and the superstructure approach.

The first characteristic is homogeneity of a collection, with respect to having documents that pertain to the same well-defined domain. If the main topics of most documents can be identified as instances of common entities in the domain (such as a disease or a therapeutic procedure in medicine or a designated project in natural resource management), the documents are more likely to share an identifiable set of information types from which one can select a set of useful semantic components. Semantically homogeneous document collections are well-suited to a domain-specific approach to developing a discourse model. A collection that covers multiple domains might also be amenable to description with a semantic component schema if multiple subcollections, each pertaining to a domain, are readily identifiable and the subcollections are relatively homogeneous.

The second characteristic is pre-existing structures. Well-defined document types, such as those specified by NEPA, make identifying document classes easier because

they share a common superstructure. Templates, manuals that prescribe how to prepare instances of various document types, or even customs to which authors tend to conform, can facilitate identifying candidate semantic components. Common structural elements within documents of the same class, such as identical section headings, can be useful if they correspond to the identified semantic components.

Both characteristics, homogeneity of domain and pre-existing structures, are more likely to be found if documents are created for a particular collection, usually by the same organization or team of authors, or if the documents are explicitly selected for inclusion in the collection by human intellectual effort. The same characteristics that facilitate schema creation are also likely to facilitate semantic component indexing, whether manual or automated. Although we believe these characteristics assist the process of semantic component schema creation, they are not necessarily required. Additional work would be required to quantitatively assess the importance of these characteristics or to determine how heterogeneous a collection can be and still benefit from semantic components.

Validation of the correctness of a semantic component schema is not a realistic goal. A semantic component schema is not intended to provide a sound or complete representation of a domain. Instead, a semantic component schema is intended to help searchers to find documents more easily in a particular document collection. Therefore, we suggest three methods for assessing a semantic component schema.

The first method is an experimental evaluation of the relative usefulness of a semantic component schema. An empirical study could determine if using the schema

for searching is better than no schema. A study could also determine if using the schema of interest is better than using another schema. However, empirical studies are likely to be quite expensive. Each document collection would need a tailored set of information needs and corresponding relevance judgments. In addition, the documents would have to be indexed using each new schema to be assessed.

The second method is to compare it to other classifications or knowledge structures in the domain. Because semantic component schemas are collection-specific, we do not expect them to be identical to knowledge structures intended to represent the entire domain. But informal comparisons can provide qualitative answers to the question "Does the schema make sense?".

The third method is to assess the reliability of indexing using the schema. If different indexers are consistent in the way they apply a schema to documents in a collection, resulting in a high inter-indexer consistency, then the schema is likely to be a good reflection of the documents in the collection. Studies of indexing consistency could compare both alternative schemas and alternative names for semantic components in a given schema.

For example, in Chapter 8 we provide a detailed description of an experiment that compared searching using the schema for sundhed.dk documents to searching without a schema. It is easy to imagine similar experiments in which both experimental systems used semantic components, but with different schemas. However, doing such experiments seems impractical, especially if the experiment uses manual semantic component indexing and interactive searching. To confirm that the schema is

reasonable, we note that the semantic components we identified are compatible with other knowledge structures in the medical domain, such as: qualifiers available in MeSH for precoordinated indexing and retrieval of medical journal articles [72]; relationships in the Unified Medical Language System (UMLS) semantic network [107]; relationships expressed in generic questions appearing in a taxonomy of clinical questions collected during observational studies of physicians [8, 9]; a list of query types for which search expressions were developed to filter retrieval of medical journal articles to those articles that report research using sound methodologies [108]; and information about drugs that is provided in the Physicians Desk Reference (PDR), which contains Food and Drug Administration (FDA)-approved labeling and other prescription information provided by manufacturers [109].

4.6. Summary

In this chapter we addressed two related questions:

- What methods are available and useful for analyzing a document collection for the purpose of identifying useful document classes and semantic components?
- What lessons can be learned from preliminary efforts to analyze a collection and use the analysis?

We proposed two methods of identifying the document classes and semantic components that comprise a semantic component schema: (1) document sampling to analyze the contents of a document collection, and (2) reusing existing document types and templates or prescriptions for document creation. We discussed these two

methods in the context of our experiences with creating semantic component schemas in two different domains. We also discussed how, and why, we refined the schemas and some lessons we learned in the process. We concluded that describing document collections with semantic component schemas is feasible, but not necessarily easy and straightforward. We found that feedback from potential users of the schema was valuable and that careful consideration should be given to the names assigned to document classes and semantic components in a schema. We also furnished an example to illustrate why allowing a document to belong to multiple classes might be useful.

In addition, we compared semantic component schemas to other domain-centered knowledge structures, observing that the elements of a schema comprise a simple controlled vocabulary and noting similarities and differences with thesauri and ontologies. We discussed how certain characteristics of a document collection can assist in schema creation and might predict the effectiveness of semantic components for searching the collection. Finally, we considered how a semantic component schema can be evaluated, noting some challenges and limitations to trying to validate a schema for a particular document collection.

Chapter 5 Expressing Information Needs with Semantic Components

Representing information needs with semantic components is an important part of using the semantic components model to retrieve documents. As a preliminary assessment of the feasibility of using semantic components to assist searching, we investigated using the semantic components model to represent information needs in a domain (clinical medicine) using the elements of semantic component schemas for two appropriate document collections. We manually mapped generic questions from a taxonomy of clinical questions to the document collections using the document classes and semantic components that we identified for each collection.

5.1. Methods

In this section we briefly review the two document collections, sundhed.dk and UpToDate®, and the schemas that we developed using the semantic components model. Then we describe the taxonomy of clinical questions and how we mapped the categories in the taxonomy to semantic components in the schemas.

5.1.1. Document Analysis

In our initial analysis of the sundhed.dk documents, we classified 72 documents according to intended audience (health professionals or patients) and orientation (clinical or nonclinical). For this study, because we were focused on the clinical information needs of healthcare professionals, we considered only the 25 documents

that we judged to be about clinical content and to be written primarily for healthcare professionals. From these documents, we identified four classes according to the semantic type of the primary topic of the document: *clinical problem* (such as a disease or symptom), *test or procedure* (such as a laboratory test or diagnostic procedure), *drug* (or class of drugs), and *clinical service* (such as information about a local specialty clinic). In the UpToDate® documents we also defined four document classes according to primary focus: *clinical problem*, *test or procedure*, *drug*, and *normal processes*.

Table 5.1 summarizes the document classes in the schemas for the two document collections. Table 5.2 shows the semantic components for documents about clinical problems in the sundhed.dk schema. Table 5.3 shows a list of semantic components for three UpToDate® document classes. Note that the tables in this chapter reflect the versions of the schemas that we used for this mapping study. Tables 5.2 and 5.3 also show which semantic components we used when mapping the categories in the question taxonomy to the two document collections. An x indicates that the semantic component was mapped to at least one clinical question category in the taxonomy.

5.1.2. The Clinical Questions Taxonomy

Ely and colleagues collected 1101 questions from Iowa family practice physicians and developed a classification scheme [8]. In a subsequent study, 295 questions collected from Oregon physicians were added and used to modify the taxonomy [9]. The researchers grouped questions with a similar structure and created generic questions

Table 5.1 Document classes in the two schemas used for the mapping study

Class	Description
	Sundhed.dk
Clinical problem	About a disease, condition, or finding
Test or procedure	About a laboratory test, or a significant diagnostic or therapeutic procedure
Drug	About a drug, or class of drugs, with respect to treating one or more clinical problems
Clinical service	About a clinical service, such as a specialty clinic
	UpToDate
Clinical problem	About a disease, condition, or finding
Test or procedure	About a laboratory test, or a significant diagnostic or therapeutic procedure
Drug	About a drug, or class of drugs, with respect to treating one or more clinical problems
Normal processes	About normal bodily processes, such as the menstrual cycle, or maternal adaptations to pregnancy

Table 5.2 Semantic components for documents about a clinical problem in the schema for sundhed.dk used in the mapping study

Component	Description	Mapped
Evaluation	Tests or procedures for diagnosis, screening, monitoring, or staging	x
Therapy	Invasive procedures, medications or other therapies	x
Management guidelines	Guidelines for managing the clinical problem	x
Referral guidelines	Guidelines about when a patient should be referred to a specialist, what evaluation or therapy should be administered before referral, and what reports should accompany the referral	x
Prevention	Strategies for preventing the clinical problem or for preventing or minimizing associated complications	x
Risk factors	Factors that increase the risk of developing the clinical problem or increase the risk of complications	x
Prognosis	The expected course, or natural history, of the problem	
Etiology	Information about causation	x
Associated conditions	Information about co-occurring, or complicating conditions, or common resulting conditions	x
Epidemiology	Populations statistics such as incidence and prevalence	x

Table 5.3 Three document classes and their semantic components in the UpToDate® schema

Clinical Problem		Test or Procedure		Medication	
Epidemiology	x	General description		Pharmacologic category(s)	
Diagnosis and workup	x	Indications	x	Administration	x
Pathogenesis	x	Pre-procedure preparation	x	Indications, use and effects	x
Treatment	x	Procedure	x	Use in pregnancy and other special conditions	x
Associated conditions	x	Complications, risks, and pitfalls	x	Adverse effects	x
Complications and sequelae	x	Post-procedure care		Contraindications and precautions	x
Prevention	x	Outcome	x	Interactions	x
Prognosis	x			Toxicity	
				Cost information	x
				Alternatives	
				Brand names	
				Patient education and storage instructions	

that could represent multiple questions asking for the same type of information, using one or two variables to represent specific concepts in the original questions. For example, questions about drug dosage can be represented by the generic question "What is the dose of drug x?" where *x* is a variable representing a drug name. Questions about which drug to use can be represented by the generic question "What is the drug of choice for condition x?" The investigators further categorized the resulting generic questions into a four-level hierarchy that reflects the type of information being sought. In addition to listing the generic questions in each category, the taxonomy includes the number of questions, from the original 1396 questions asked by physicians, that were abstracted into the generic questions in each category of the taxonomy.

The resulting hierarchical taxonomy contains 64 question categories, each based on one or more related generic questions. The top levels of the taxonomy are: *diagnosis* (18 categories, 525 original questions), *treatment* (23 categories, 611 original questions), *management (not specifying diagnostic or therapeutic)* (8 categories, 126 original questions), *epidemiology* (5 categories, 82 original questions), *nonclinical* (9 categories, 52 original questions), and *unclassified* (1 category, 0 questions) [9]. For this study, we only considered the categories in the first four top levels because we were interested in mapping clinical questions to document collections that provide clinical information. We also eliminated the four "not elsewhere classified" categories (one in each top level of the hierarchy) since there was no information to use for mapping the questions. Instead of listing generic questions, the "not elsewhere classified" categories state that "generic type varies." The comment section describes the categories as "In a broad sense, the question is about X, but it does not fit any other diagnosis category" where *X* is the name of the top level category, such as "diagnosis" or "treatment." Eliminating the *nonclinical*, *unclassified*, and "not elsewhere classified" categories left 50 categories.

5.1.3. Mapping Questions to Semantic Components

For each category, we (manually) tried to identify one or more combinations of a document class plus a semantic component associated with that class in each collection that would be reasonable to express the types of questions represented by the category. We used both the generic questions and the comments associated with

each category to understand the intent and scope of each category in the taxonomy. For example, the first category in the question taxonomy is about *diagnosis* related to a *clinical finding*. The comment describing the category states "you start with a finding and you want to know what condition is causing it. You know what the finding is, you don't know what the condition is." The generic questions associated with this category include "What is the cause of symptom x?" and "What is the likelihood that symptom x is coming from condition y?" One possible way to search for answers to questions in this category is to look for documents about *symptom x* that contain a discussion of the causes of *symptom x*. In the context of the semantic components model, we would do a topical search for documents about *symptom x* and refine the search by looking for documents in the class *documents about clinical problems* that contain the semantic component *etiology* (sundhed.dk) or *pathogenesis* (UpToDate®). We refer to this selection of an appropriate document class and an appropriate semantic component as a *mapping*. For this category, the generic question that asks about *condition y* as a cause of *symptom x* can also be represented by a search for documents about *condition y* that discuss *symptom x* as a manifestation of *condition y*. We could search for *documents about clinical problems* that are about *condition y* that also contain the term for *symptom x* in the *evaluation* (sundhed.dk) or *diagnosis & workup* (UpToDate®) semantic component. We considered a mapping successful if at least one combination of document class and semantic component represented the types of generic questions associated with the category.

5.2. Results

For sundhed.dk, we mapped 34/50 (68%) of the question categories, and for UpToDate®, we mapped 36/50 (72%) of the question categories. The taxonomy also includes the number of original questions (asked by physicians during the observational studies) that the taxonomy authors assigned to each category, so we can calculate the proportion of actual questions that belonged to the mapped and unmapped categories. Based on the question frequency for the categories, over 92% of questions could be mapped for both collections (after eliminating the nonclinical and nonspecific questions as noted above).

If all question categories in the taxonomy are considered, our mappings covered 34/64 (53%) and 36/64 (56%) of the categories. If we eliminate just the category that had no questions (*unclassified*), our mappings covered 34/63 (54%) and 36/63 (57%) of the categories. Based on question frequency, the coverage is over 88% of all questions for both resources.

Some categories contained multiple related generic questions, suggesting multiple related mappings. For example, Category 2.1.2.1 contains ten generic questions, including: "What are the indications for drug x?" and "Is drug x (or drug class x) indicated in situation y or for condition y?" In the UpToDate® collection, we mapped this category to the class of documents about drugs. The first question, *What are the indications for drug x?* could be mapped to the semantic component *indications, use & effects* in documents about *drug x*. The second question, *Is drug x (or drug class x) indicated in situation y or for condition y?* could be mapped either to documents about

drug x in which *condition y* appears in the *indications, use & effects* component, or to documents about *condition y* in which *drug x* appears in the *therapy* component. The second question is an example of what we refer to as a question that can be mapped in two directions, where either

- *x* is the main topic and *y* is searched for in text about an aspect of *x*
- *y* is the main topic and *x* is searched for in text about an aspect of *y*

Nineteen categories contained at least one generic question that could be mapped in two directions. These examples (which can also be represented as full relations) highlight the importance of identifying the topic of the search as well as the semantic component of interest.

Table 5.4 shows our mappings for five categories. The first three categories shown are the three categories that were most frequent in Ely's study. The fourth category is a category with a single type of mapping. The fifth category shown is an example of a category for which we did not identify a mapping. We abbreviated the category description and the example generic questions from the original taxonomy (as presented in the supplement [9] to the paper) for presentation in the table. The variables *x* and *y* indicate values that should correspond to the focus of the document (for instances of the document classes in the Doc. Class columns) or that should be present in instances of the semantic component (in the Semantic Component columns). When no variable is present, then the question does not specify a value for document focus or content of the semantic component instance. Variables shown in

Table 5.4 Example mappings (created manually) for five question categories from the clinical questions taxonomy.

Code	Freq (%)	Category Description	Example generic questions (from supplement to paper)	sundhed.dk Doc. Class	sundhed.dk Semantic Component	UTD Doc. Class	UTD Semantic Component
2.1.2.1	10.7	Treatment: drug prescribing: efficacy/ indications/drug of choice: treatment	Is drug x (or drug class x) indicated in situation y or for condition y? OR What are the indications for drug x? OR …	Drug (*x*)	target pop. (*y*)	Drug (*x*)	indications, use, and effects (*y*)
				Drug (x)	benefits (y)		
				Problem (*y*)	treatment (*x*)	Problem (*y*)	therapy (*x*)
1.1.1.1	8.2	Diagnosis: cause/ interpretation of clinical finding: symptom	What is the cause of symptom x? OR What is the differential diagnosis of symptom x? OR Could symptom x be condition y or be a result of condition y? OR …	Problem (*x*)	etiology (*y*)	Problem (*x*)	pathogenesis (*y*)
				Problem (*y*)	evaluation (*x*)	Problem (*y*)	diagnosis & workup (*x*)
1.3.1.1	8.0	Diagnosis: test: indications/ efficacy	Is test x indicated in situation y? OR What test (or evaluation, or work up), if any, is indicated/ appropriate in situation y or with clinical findings x1, x2, . . , xn? OR …	Test/proc. (*x*)	indications(*y*)	Test/proc. (*x*)	indications (*y*)
				Problem (*y*)	evaluation (*x*)	Problem (*y*)	diagnosis & workup (*x*)
4.1.1.1	1.0	Epidemiology: prevalence/ incidence	What is the incidence/ prevalence of condition y (in situation z)? OR Why is the incidence/ prevalence of condition y changing?	Problem *y*	epidemiology	Problem *y*	epidemiology
1.4.1.1	0.6	Diagnosis: name finding: body part on physical exam or imaging study	What is the name of this body part? OR What is the anatomy here?	NO MAPPING		NO MAPPING	

parentheses indicate the presence of multiple generic questions that correspond to mappings with and without variables.

5.3. Analysis

We successfully mapped a substantial majority of questions to document classes and semantic components in both collections. Based on reported frequencies in the taxonomy, we mapped over 92% of the questions we considered. We report coverage based on question frequency to illustrate that the most common types of questions map easily in our model. This does not mean that the answers to all instances of the questions can be found, only that one or more semantic components can easily be identified as most likely to satisfy the information need. For example, the generic question "What is the preparation for test x?" can be mapped to *documents about a test or procedure* and the semantic component *pre-procedure preparation*, but the specific question "What is the preparation for a sigmoidoscopy?", in which the variable *x* is instantiated, cannot be answered if the collection does not contain a document about sigmoidoscopy.

Table 5.5 shows the frequencies of the categories for which we found only partial or possible mappings or that we did not map at all. We classified several categories as possibly or partially mapped for two main reasons. Either the question lacked sufficient detail or we thought only some questions in the category could be mapped. An example of a question with insufficient detail is the generic question "Why did

Table 5.5 Analysis of unsuccessful, or partially successful mappings.

Results	Sundhed.dk		UpToDate	
	# categories	% questions	# categories	% questions
Possibly/partially mapped	5	2.1	5	2.8
Not mapped: no semantic component in the question	2	0.4	2	0.4
Not mapped: other reasons	9	5.1	7	4.1

provider x treat the patient this way?". This question could represent either a question with a clear scientific answer or a rhetorical question that is not answerable from medical literature. An example of a category for which we thought some of the questions might be answerable in the collection, and were thus mapped, and other questions might not be answerable and were not mapped is Category 2.1.12.1: *treatment – drug prescribing – availability* that has two generic questions. An answer to the generic question "Is drug x available over-the-counter?" might appear in an *administration* semantic component but the answer to the other question in the same category "Is drug x available yet?" probably would not appear in the *administration* component.

The two questions we did not map because there was no semantic component suggested in the question were the two general questions: "What is condition x?" and "What is test x?". These are the only "aboutness" questions, and are examples of the type of questions that are usually answered well by simple topical queries in existing systems.

We did not map the remaining questions (for "other reasons") because the two document collections were not appropriate for these questions. Some were name-finding questions, such as "What is the name of that condition?". A clever query

might remind the user of a name, but we could not assume that a searcher might think of such a query and we did not try to predict how it might be phrased. We therefore could not determine how the query should be mapped. The other cases consisted of requests for information types that we did not observe in the sample documents. Our experience mapping queries suggests that providing a list of document classes with corresponding semantic components could help a searcher quickly decide whether to search a given collection. If the information need is not a simple topical question, and does not map to a combination of document class and semantic component in the collection, the search might be better directed elsewhere.

The semantic components in the two collections had substantial overlap, which is not surprising. The semantic components reflect common physician works tasks (such as *diagnose*, *manage*, *treat*, *refer*) and important clinical issues (such as *drug interactions*). We also noticed some differences between the collections that reflect differences with respect to the audiences and the practice milieus for which the collections are intended. These differences are highlighted when developing semantic component schemas. The sundhed.dk documents are intended primarily for family physicians, in large part to promote integrated care between family physicians and specialists. The Danish healthcare system is managed largely at the regional level, and most of the documents are produced by, and written for, users in a particular region. Furthermore, the "gatekeeper" role of the family practitioner is more prominent than in the U.S. As a result, *referral guidelines* are an important semantic component in the sundhed.dk documents, and reflect regional practices. Similarly, the *test or*

procedure documents contained a component, *practical information*, that provided locally-tailored instructions for handling laboratory samples. The UpToDate® collection does not address issues specific to local or regional practices.

We performed the study in the medical domain because we had access to document collections and to the taxonomy of information needs. Few other domains have been studied as thoroughly as the medical domain with respect to information needs and the resources that can satisfy those needs. As a result, few other domains have such rich resources for studying domain-specific information retrieval. However, the semantic components model is not limited to any particular domain. If an appropriate taxonomy of information needs were available, it would be possible to do a similar study in a different domain.

5.4. Related Work

The aspects we identified in the two collections of documents about medicine are similar to qualifiers in the Medical Subjects Heading (MeSH) vocabulary [72] used to index and search MEDLINE documents. Both semantic components and MeSH qualifiers can be used to add specificity either to an index or to a search. There are three fundamental differences however. First, MeSH contains the notion of aspects, but they are not associated with classes of documents. For individual documents (or searches) the indexer (or searcher) can associate a qualifier (that can represent an aspect of the concept represented by a term) with a specific indexing (or search) term. Second, MeSH is a vocabulary that was designed specifically for indexing and

searching a particular collection of documents. The semantic components model is intended as a framework that allows the development of a set of document classes and semantic components appropriate for any given document collection and is not restricted to use in a particular domain. Third, MeSH qualifiers are associated with an entire document. Semantic component instances are subdocuments, allowing the searcher to restrict the search for certain terms to occurrences *within* selected semantic components. In effect, the ability to search for a term within a labeled subdocument allows the searcher to specify a search for a full relation, not just the partial relation that is represented by a MeSH descriptor/qualifier pair.

Several interesting retrieval systems in the medical domain use query models based on generic queries that incorporate relationships between concepts or aspects of topics.

ELBook is a system for retrieving very fine-grained information from medical documents such as textbooks [110]. The text is indexed by associating queries with text segments, which can be as small as sentences or cells within a table, that answer those queries. The query model consists of generic queries populated with concepts from the UMLS. Instead of searching the text, the user identifies the query that will point him directly to the answer. This query model offers very precise, but possibility limited, searching. The quality of the searching experience is likely dependent on how well the indexer predicted the user's information need. We are not aware of an evaluation of the searching performance for ELBook.

DynaCat uses the UMLS to dynamically categorize search results [111]. The user issues a traditional query and also selects from a set of nine query types. The query types are similar to our semantic components in that they consist of a topic type (problem, symptoms, treatment) associated with an aspect (such as preventive actions, risk-factors, treatments). Interestingly, two of the types are reciprocal: problem-treatments and treatment-problems; this reciprocity is similar to our mapping queries in two directions. DynaCat uses the query types, not to refine the query, but to organize the documents into appropriate categories (in conjunction with indexing keywords and their semantic types) for presentation to the user.

Cimino and his colleagues have done extensive work linking the electronic patient record to medical knowledge resources using InfoButtons [112-115]. The researchers use information from the patient record to populate generic queries with specific concepts. The generic queries are mapped to automatic search strategies; when the user chooses a generic query, the system uses the query and specific concepts extracted from the patient record to compose a search strategy appropriate to whichever knowledge resource is appropriate. For some MEDLINE queries, the relationship in the generic query is used to select a MeSH qualifier. For example, if the user chooses the generic query "What is the treatment for <disease>?", the system uses the qualifier *Drug Therapy* in its search strategy. The InfoButton system does not otherwise appear to exploit aspects or relationships, and the queries are limited by the underlying indexing and query languages of the resources being searched.

All three of these systems make use of existing terminologies, such as the UMLS [107] and the Medical Entities Dictionary (MED) [114]. The UMLS Metathesaurus and Semantic Network provide extensive coverage of the concepts and relationships in medicine. In general, ontologies and other knowledge organization models intended for use by computer applications specify the concepts in a domain plus relationships between the concepts and can be quite comprehensive. With semantic components, our goal is different. We want to express only those relationships or aspects of concepts that are important in a particular document collection and we want to express an information need in a way that helps users to find desired documents.

5.5. Summary and Conclusions

We used a taxonomy of generic questions to represent information needs and used semantic component schemas to describe two document collections that were appropriate to the information needs in the taxonomy. We had developed the schemas based on the contents of the document collections, not based on the information needs represented in the taxonomy. We investigated what proportion of the clinical questions and clinical question categories could be expressed using the semantic component schemas. We found that a large proportion of clinical questions can be expressed using the document classes and semantic components we identified in the two collections of documents that provide information to clinicians.

We conclude that the semantic components model is capable of representing information needs. The ability to represent information needs is necessary, but not

sufficient, for demonstrating the potential usefulness of the semantic components model for searching. In Chapters 7 and 8 we investigate the process of semantic component indexing and the usefulness of semantic components for searching, respectively.

Chapter 6 Evaluation of Semantic Component and Keyword Indexing

Ultimately, we want semantic component indexing to improve a user's ability to retrieve desired documents quickly and easily. Directly measuring indexing effectiveness (whether an indexed document is correctly retrieved every time it is relevant to a query [116]) is not feasible, but searching studies can evaluate indexing effectiveness in combination with other factors that may influence retrieval and relevance judgments. In Chapter 8 we report the results of a searching study that compares two experimental search systems, one with conventional indexing (in this case, full text indexing supplemented with manual keyword indexing) and one with semantic component indexing that supplements the conventional indexing. Comparing the search results achieved by each system is one way to assess the effectiveness of semantic component indexing.

As part of assessing the feasibility and potential usefulness of the semantic components model, we must assess the feasibility of semantic component indexing. Assessing the feasibility of indexing includes determining how much time is required for indexing, assessing how the indexers perceive the difficulty of the task, and measuring the quality of the indexing. In Chapter 7, we report our findings from a study that compared manual semantic component indexing to manual keyword indexing with respect to time required to index documents, perceived difficulty, and the quality of the indexing performed. First, we must consider how to evaluate the quality of indexing. In this chapter, we focus on identifying specific qualities of

indexing that may contribute to its effectiveness, that are easier to measure than effectiveness, and that can serve as surrogates for predicting effectiveness. We discuss and critique candidate methods for measuring such qualities and propose methods for evaluating instances of semantic component indexing. We also discuss evaluation of these qualities for keyword indexing. In Chapter 7 we use the methods discussed in this chapter to evaluate data from the indexing study.

The two qualities of interest that we will consider are accuracy and consistency. We use *accuracy* (or correctness) to refer to how well an instance of semantic component indexing represents document content in the framework of document types and semantic components that have been defined for a particular document collection. This definition implies comparison to an ideal indexing instance. We use *consistency* (sometimes called reliability, reproducibility, or inter-indexer consistency) to refer to the similarity among indexing instances when different indexers index the same document. A related concept is *stability*, which refers to the similarity between different indexing instances produced by the same indexer at different times, which can be considered a special case of consistency. The same metric that is used to measure similarity between indexing instances produced by different indexers can also be used to measure stability, so we do not further discuss stability.

We want to be able to measure the accuracy and consistency of indexing instances. The two measurements reflect the similarity between indexing instances, and many of the criteria for a good metric will be the same for both accuracy and consistency. We

use the term *agreement* for the similarity that is being assessed between two or more indexing instances. Agreement can apply to either accuracy or consistency.

Assessing accuracy by comparing an indexing instance to an ideal indexing instance, or gold standard, is not actually feasible. The semantic content of a document is open to subjective interpretation and any attempt to represent that content is unlikely to be agreed upon by all observers as a completely correct, or "gold," standard. We can, however, establish an acceptable reference standard by using a commonly accepted technique, such as expert opinion or consensus formation. We can use the reference standard for evaluating an indexing instance, while recognizing the limitations of what the standard actually represents. The "goldness" required can vary depending on the context and goal of the evaluation. The effort invested in creating a reference standard, and its resulting quality (its "goldness"), is an orthogonal issue to methods used for comparing an indexing instance to the standard. For example, the same measurement of accuracy that is used to compare the indexing of a trainee human indexer to an expert-generated reference standard can also be used to compare an automatically generated indexing instance to a manually produced indexing instance that is assumed to be the reference standard (regardless of the formal training of the human indexer). In both cases, the comparison is binary and asymmetric. One indexing instance is the instance being evaluated, the other is the reference standard.

Assessing consistency may involve two or more instances and is symmetric, that is, none of the indexing instances is assumed to be preferable to any other instance. In

other words, for measuring consistency there is no gold standard. Although indexing consistency does not directly reflect correctness, because indexing can be consistently incorrect, consistency is easier to measure than accuracy because it does not require constructing a reference standard. Consistency is likely to be at least somewhat predictive of accuracy, and perhaps effectiveness, because consistency among indexers is likely to reflect the ability of the indexers to understand the task and perform it well. Writing about consistency with respect to keyword indexing, Rolling [116] stated that "... since the selection of indexing terms by an indexer reflects his judgment regarding the information contained in the document and its representation, indexing consistency is essentially a measure of the similarity of reaction of different human beings processing the same information." We argue that the same is true of semantic component indexing, which is another method for an indexer to record his judgment about information contained in a document. A judgment about document content that is similar to other judgments about the document's content is more likely to be a faithful representation of document content than a judgment that is dissimilar to other judgments. Consistently incorrect semantic component indexing may result from a different, but consistent, interpretation of a document class or semantic component label than what was intended. Frequent users of a system are likely to adjust their searching behavior more easily to accommodate imperfect, but consistent, indexing than to accommodate inconsistent indexing.

If we accept that indexing consistency reflects similarity of judgments about document content and its representation, then consistency among indexers may predict

that searchers will make similar judgments. A searcher issues queries based on his expectations about how desired content is likely to be represented in a particular system. A successful search outcome is more likely if there is a good match between the searcher and the indexer with respect to interpretation of document content and representation of content using the indexing language (such as a controlled vocabulary or semantic component schema).

Indexing consistency might also reflect the quality of an indexing language itself. If a semantic component schema does not reflect the content and organization of a document collection, it will be difficult for indexers and searchers to use the schema consistently. Similarly, interpretation and use of an indexing language will be hampered by poor choices with regard to the names used to represent concepts or indexing entities (such as document classes, semantic components, or controlled keywords), the descriptions of appropriate usage associated with each name, the degree of specificity or generality of terms, and the coherence of hierarchies. If indexers frequently have different interpretations of either semantic components or controlled keywords, then it is likely that searchers will have different interpretations as well. If indexers and searchers have disparate interpretations of the elements used for indexing, then search success is likely to be degraded.

Methods to evaluate semantic component indexing are important not only for assessing indexing quality in a particular setting, but are also important as part of assessing the costs of scaling semantic component indexing to larger and more diverse collections. For example, we may want to: (1) assess the expertise (and therefore cost)

required to extend manual semantic component indexing to larger or different collections without loss of indexing quality, or (2) assess the degradation of indexing quality if manual semantic component indexing is replaced with a different (cheaper) approach, such as automated or semi-automated indexing.

The remainder of this chapter is organized as follows. In Section 6.1 we discuss the properties of semantic component and keyword indexing and propose criteria for evaluation of indexing instances. In Section 6.2 we compare semantic component indexing to related tasks and discuss candidate metrics for evaluating indexing instances. In Section 6.3 we analyze a family of metrics, Krippendorff's Alpha, and describe implementation of three of its forms. In Section 6.4 we offer recommendations for evaluating indexing. We summarize in Section 6.5.

6.1. Properties and Criteria for Evaluation Metrics

We propose two desirable properties for consistency metrics that apply to all of the indexing tasks we are about to discuss:

1. A consistency metric should indicate the extent of agreement that exceeds the agreement we would expect by chance alone. Agreement by chance is particularly likely to occur when the indexers are choosing among a small number of alternatives. For example, with only two alternatives, random choice would result in agreement in 50% of instances.
2. When measuring consistency, a metric should be able to compare the judgments of any number of indexers. This criterion specifies that a metric ideally should be

capable of a global comparison of multiple indexing instances. Averaging multiple pair-wise comparisons muddles the meaning of a consistency measure and can be unwieldy when many indexing instances are available for comparison.

6.1.1. Characteristics of Semantic Component Indexing that Affect Measures of Agreement

In Chapter 1 we proposed three ways that semantic components can be useful for searching. In addition to a traditional query (using natural language or a controlled vocabulary), a searcher using an IR system with semantic components: (1) can indicate terms that should appear in specified semantic component instances; (2) can specify one or more semantic components that he desires to be present in retrieved documents (without specifying particular terms that should appear in the semantic component instances); and (3) can view, in the results display, a list of the semantic components instances present, and their relative sizes, in each retrieved document. When considering appropriate metrics for evaluating semantic component indexing, we focus only on the first use of semantic components. We expect that any measure of semantic component indexing quality that pertains to the first way of using semantic component indexing will also be relevant to the other uses as well. In this section we discuss the nature of semantic component indexing and propose criteria for good measures of the accuracy and consistency of indexing instances.

Two related tasks comprise semantic component indexing of a document: (1) designation of the document class, and (2) identification of semantic component

instances in the text. Designation of document class consists of selecting a document class from a list of labels. Identification of semantic component instances consists of several related tasks: identifying text that contains information pertaining to a semantic component, indicating the boundaries of such text, and labeling the text with the appropriate semantic component name. Designation of document class and identification of semantic component indexing are distinct tasks that, although related, should be evaluated separately. The basic unit for indexing evaluation is an indexing instance. Each indexing instance has a single document type and a single instance of semantic component labeling. An instance of semantic component labeling can include multiple semantic component instances and each semantic component instance can consist of one or more semantic component segments.

6.1.1.1. Assigning Document Class

Document class assignment is a nominal categorization task. Each document is placed into exactly one category chosen from multiple unordered categories. We propose two criteria for useful metrics for comparing instances of document class assignment: (1) a metric should reflect whether indexers actually agree on the category chosen, and (2) a metric should be able to handle any number of categories.

The first criterion specifies that agreement means that indexers chose the same category. Here we are making a distinction between agreement and correlation. Systematic classification decisions, such as tending to choose a particular class more frequently than other classes or always choosing one class for a subgroup of

documents that other indexers would place in a different class, can result in one indexer's decisions being different from, but highly correlated with, another indexer's decisions. Choice of document class *s* by indexer *a* might predict choice of class *t* by indexer *b*, resulting in a high correlation coefficient despite disagreement about choice of document classes. We want to identify metrics that reflect agreement, not just correlation.[19] The second criterion, allowing an indexing schema to have an arbitrary number of document classes, ensures flexibility for applying the metric in a variety of situations.

These two criteria are orthogonal to the two criteria for consistency metrics, accounting for agreement by chance and handling an arbitrary number of indexers, put forth at the beginning of Section 6.1. Correlations can occur by chance, and metrics that measure correlation can account for the probability of chance correlations. We have extended the scope of our desired evaluation to encompass an arbitrary number of categories as well as an arbitrary number of indexers.

6.1.1.2. Identifying Semantic Components

Comparing the semantic component labeling portion of two indexing instances involves comparing both the labels and the extents of labeled segments. By *extent*, we mean the range of text included in a given segment. Figure 6.1 shows four instances

---

[19] Explicitly measuring correlation might be useful in some circumstances, such as for determining the "confusability" of two document classes when developing or evaluating a semantic component schema.

of semantic component indexing for a short snippet of text.[20] Text highlighted in color is an instance of the semantic component with the label in the callout box of the same color. The extent of the *epidemiology* semantic component in Instance 1 is the nine-word sentence highlighted in yellow. If a given section of text, such as a word, is labeled with the same semantic component name in two indexing instances, then it is easy to conceive of the evaluation task as comparing the boundaries of the two semantic component instances that include the word. For example, the word "frequency" appears in Instance 1 and Instance 2 of the *epidemiology* component in Figure 6.1. But what if the word is labeled with different semantic components in the two indexing instances? In Instance 2 the word "age" appears in the *epidemiology* component, but in Instance 3 it appears in the *diagnosis* component. Do we compare the labels? Or do we compare the label and the extent? What if one indexing instance has multiple overlapping semantic component instances and all instances include the same word, such as the word "frequency" in Instance 4? The text highlighted in green in Instance 4 is part of two components, one highlighted in yellow (labeled epidemiology) and one highlighted in blue (labeled diagnosis).

Semantic component indexing is intended to represent the meaning of text by grouping segments of text in a document that contain information about the same aspect of the main topic of the document. Semantic component schemas that are appropriate to a particular document collection and user group may not match the

---

[20] The snippet is composed of sentences extracted from http://www.emedicine.com/med/topic1816.htm

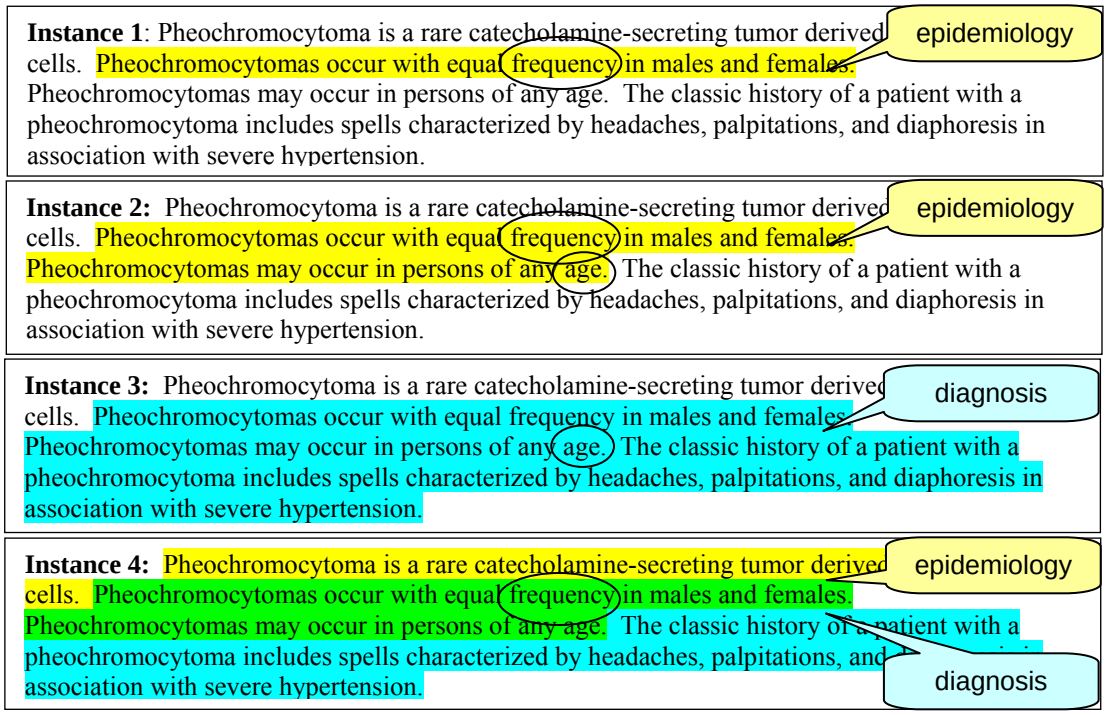


Figure 6.1 Four instances of semantic component indexing for the same document.
Selected words are circled to highlight issues regarding comparison of difference indexing instances, as discussed in Section 6.1.1.2.

document organization used by individual document authors. Therefore, instances of different semantic components in a given document are likely to overlap. We argue that the primary unit of evaluation for a given indexing instance should be a semantic component, and that each semantic component should be evaluated separately. We make this argument for two reasons. First, the semantic component is a basic unit for influencing document retrieval and ranking. Second, if we evaluate each semantic component separately, then allowing a given fragment of text to be in more than one semantic component does not complicate the evaluation. By analyzing each component separately, we can also gain more insight into the quality of indexing for

each component. If a particular semantic component has low agreement, we may be able to improve indexing quality by interventions such as enhancing indexer training, improving documentation, or reconsidering the semantic component schema itself. Although aggregating the evaluation results for all of the semantic components in a document, or document class, may also be of interest, the initial task should be evaluation of each semantic component, whether the evaluation is of consistency or accuracy.

Next we define six relationships that can occur between semantic component segments and semantic component instances that belong to different indexing instances: identity, independence, overlap, nesting, containment (the inverse of nesting), and subsumption. Suppose we have two semantic component instances, *i* and *j*. Let *s* be a segment in *i* and *t* be a segment in *j*. Instances *i* and *j* are *identical* if both instances have the same number of segments and segments can be paired so that every pair contains one segment from each indexing instance and the extents of the segments in the pair are equal. Instances *i* and *j* are *independent* if no text that appears in *i* also appears in *j*. Instances *i* and *j overlap* if some text is included in both *i* and *j*, but each instance also includes some text that is not part of the other instance. Segment *s* (in instance *i*) is *nested* in segment *t* (in instance *j*) if segments *s* and *t* are not identical and if all the text in segment *s* is also in segment *t*. In other words, the text in segment *s* is a proper subset of segment *t*. If segment *s* is nested in segment *t*, then segment *t contains* segment *s*. If *s* is the only segment in instance *i*, or if all the segments in *i* are nested in segments belonging to *j*, then instance *j subsumes* instance

i. Observe that, if *s* is nested in *t* and instance *i* also contains at least one segment that is not nested in a segment belonging to *j*, then instances *i* and *j* overlap. In Figure 6.2, each panel illustrates a relationship that can occur between two semantic component segments or instances. In this example, each instance consists of a single text segment and both instances can be assumed to have the same label. One instance is shown outlined by a solid, rectangular blue box, while the other instance is outlined with a dashed, round-edged red box. In panel D, the segment surrounded by the dashed, round-edged red box is nested in the segment surrounded by the solid rectangular blue box, and the segment in the solid box contains the segment in the dashed box. The instance in the solid box also subsumes the instance in the dashed box because both instances each consist of a single segment.

Having decided to measure agreement between instances of each semantic component separately, we still have several issues to consider when determining the criteria for a good metric:

- how to measure length of text segments
- whether the relative nearness of two independent instances should affect the measurement of agreement
- whether the position of a nested segment within the containing segment should affect the measurement of agreement
- whether a difference in the number of segments within different instances should affect measurement (assuming the amount of overlap is the same)

We discuss each of these properties in turn.

Figure 6.2 Relationships between semantic component instances

The first issue we consider is the appropriate unit for measuring position (location) and length of a semantic component instance in a document. A document and a given semantic component instance each has a total size, or length, and that length can be measured in characters, words, sentences, or other units. Semantic component instances can consist of zero or more discontiguous segments, each of which also has a length. Each segment also has a position, indicated by its boundaries, which are the beginning and end position in the text. (Position can be indicated equivalently using one boundary and a length). Several candidate units are worthy of consideration.

Because we are interested in the semantics of text, we might consider a sentence, or a clause, as a basic unit for conveying meaning. Both sentences and clauses have drawbacks as candidates units. Sentences boundaries are not always easy to detect accurately in text, and finding clause boundaries would require parsing tools and would incur additional computational costs. Furthermore, if a human indexer placed a boundary across a clause or sentence boundary, comparing that instance to other instances that adhered to the clause or sentence boundary would require a decision to either include or exclude partial clauses or partial sentences in the semantic component instance. Another candidate unit is a word. Words are the most common unit for full text indexing and for matching natural language queries to text. Automatically detecting word boundaries is relatively easy and accurate but still requires some preprocessing that could introduce inaccuracies, particularly with respect to punctuation and formatting characters. Manually marked indexing instance boundaries could also fall within words, although they are less likely to fall within words than within sentences. Neither sentences nor words will allow us to deal with graphics, images, or other multimedia content that can occur in documents. We argue that using the smallest unit that a user can individually select in the base application (used to create or display the document) and for which the base application can return a location, provides high precision and low ambiguity. For text, this is usually the character. This approach also has generality; the same approach would be useful for graphics, images, or other multimedia content. For the remainder of this discussion

we assume that documents consist only of text and that the character is the unit of measurement.[21]

Next we consider relative positions. Should independent but nearby instances ("near misses") be considered more similar than independent instances that are far apart? If our primary interest were to compare the similarity of text interpretation by different indexers, then we would want to know whether near misses reflect more semantic relatedness of text than non-near (far) misses. For IR applications, if we are concerned only with the matching of query terms to document terms (which could be terms in full text, terms in keyword indexing, or terms in semantic component instances) then near misses are the same as far misses. Here we are interested in whether query terms match terms in semantic component instances. Near misses will not help retrieval and, therefore, a good evaluation metric should treat near misses and far misses as equivalent.

We make a similar argument with respect to the position of nested segments. We want our agreement measurement for two semantic component instances (belonging to different indexing instances) to reflect the similarity of their effectiveness. In other words, how similar is the likelihood that each instance will contribute to correct retrieval of the document whenever a query is relevant to the document? Although a

---

[21] However, we note that the nominal and binary categorization metrics discussed in later sections are compatible with units that have positions in multiple dimensions, such as a pixel. The unitization metric (see Section 6.3.3) assumes that position is measured on a linear scale. Application of the unitization metric to units with position data expressed in two or three dimensions would require defining a new difference function for observed disagreement. It is unclear how expected disagreement would be calculated.

nested segment that is centered might be more semantically similar to the containing segment than a nested segment located near a segment boundary, we have no way of knowing in advance whether query terms are more likely to match words in the center of a segment or words near the boundaries. We argue that a good evaluation metric should treat nested segments that are of the same size, but that occur in different locations relative to the containing segment, as having equal agreement.

We extend the same logic to argue that partitioning one segment into multiple segments, while maintaining the same amount of overlap with another segment, should not affect agreement because it does not affect the likelihood of query words occurring in the overlapping, versus nonoverlapping, parts of the text. To illustrate the equivalent agreement between contiguous and discontiguous instances, let *i*, *j*, and *k* be instances of semantic component *c* appearing in three different indexing instances. Suppose also that instance *j* overlaps instance *i* by a given amount, and that both *i* and *j* consist of a single segment. Instance *k* overlaps *i* by the same amount as *j* but the overlap occurs in two discontiguous segments. Should *j* be considered more, or less, similar to *i* than *k*? Whether a query matches *i*, *j*, or *k* depends only on whether a query term matches a term in the instance, not how many segments comprise the instance. We therefore argue that discontinuities are not, by themselves, significant and that the similarity between *j* and *i* and between *k* and *i* is the same. This approach is different from content analysis (discussed in Sections 6.2 and 6.2.2), where the number of segments can be very important. A content analyst might be interested in

the number of occurrences of an event, such as the number of violent acts in a television program or in a story.

In summary, a good metric for measuring the agreement between semantic component indexing instances should meet the following criteria:

- It allows comparing instances of each semantic component separately.
- For segments of a given length, it results in more agreement when the length of the overlap or the length of the nested segment is larger.
- It treats near misses the same as far misses.
- For segments of a given length, it results in the same measured agreement regardless of the position of the nested segment within the containing segment.
- It allows semantic component instances to be discontiguous. It measures overlap between instances and is agnostic regarding whether such overlaps are contiguous and whether the number of segments is the same.

6.1.2. Characteristics of Keyword Indexing that Affect Measures of Agreement

As described in Chapter 2, some document collections are indexed with keywords, usually assigned manually. Keywords have traditionally been applied to whole documents, although they also could be assigned to subdocuments (such as semantic component instances). We consider two types of keyword indexing: (1) keywords that are chosen from a controlled vocabulary, and (2) "free" keywords that are chosen from unrestricted natural language. For both types of keyword indexing, the number of

terms is usually flexible, not fixed, but represents a small proportion of the total universe of possible indexing terms.

Lancaster describes keyword indexing as having two principal steps: conceptual analysis (deciding what a document is about) and translation (deciding what terms to use as representations of the concepts) [22]. Comparing indexing instances with free keywords can be especially challenging because, in the translation step, indexers can choose synonyms to represent the same concept or may choose linguistic variants of the same base word (such as different verb tenses or a gerund instead of a verb). When developing an evaluation procedure, one must decide whether to perform some normalization before determining whether certain keywords are distinct or not. Controlled vocabularies diminish the problem of synonyms and word variants by normalizing the terms used to represent various concepts. In this work we assume that any normalization has already been performed so that our concern is how to measure agreement for a given set of keywords.

The two principal characteristics of keyword indexing that affect how we measure agreement are (1) the number of keywords assigned can vary both by document and by indexer, and (2) the universe of possible keywords is large and possibly unlimited. Keywords can be viewed as categories; assigning a keyword places the document in a category represented by the keyword. The universe of categories is equal to the number of keywords in a controlled vocabulary and is essentially unbounded for indexing with free keywords. Although the universe of "legal" keywords can be large, common sense suggests that human indexers do not consider the entire population of

keywords when they choose keywords to assign to a document. What, then, is an appropriate universe to consider?

If different indexers assign different numbers of keywords to the same document, how many items are we judging for agreement, or disagreement? If each indexer were limited to the same fixed number, *m*, we could conceptualize the indexing process as making *m* decisions or as filling *m* slots. For a flexible number of keywords, an automated keyword indexer could be viewed as making *n* decisions to accept or reject each of *n* keywords in a vocabulary of size *n*. Human indexers do not explicitly consider every keyword in a large vocabulary. How many do they consider?

When we consider how to account for the probability that two indexers will assign the same keyword by chance, the same question arises. We must decide what is an appropriate universe to consider. For semantic component indexing, the indexer chooses document classes from a schema and selects text to include or exclude from a small number of semantic components associated with that document class. The number of choices that could be made at random are limited. For keyword indexing, assuming that every term in a large controlled vocabulary is equally likely to be chosen is not reasonable. Agreement could be trivially increased by simply adding more terms to the vocabulary, regardless of how irrelevant they might be to the document being indexed. Agreement by chance for free keyword indexing would approach zero because the universe of choices is unlimited.

Therefore, in addition to the desirable properties for all indexing consistency metrics, we propose the following criteria for a measure of agreement applied to instances of keyword indexing:

- It allows comparing instances with different numbers of assigned keywords.
- It does not allow the probability of chance agreement to be artificially decreased by merely increasing the universe of possible indexing terms.

6.2. Tasks Related to Indexing and Candidate Metrics for Agreement

In this section, we discuss tasks that are similar to assigning document classes, identifying semantic component instances, and keyword indexing. We discuss metrics that have been used in the literature to evaluate these tasks and analyze the potential usefulness of the metrics for semantic component and keyword indexing.

First, we consider how each of these tasks can be treated as text categorization (Section 6.2.1). The automated text categorization literature discusses evaluation of automatic text categorization systems by comparing categorization results to reference standards, which typically represent human judgments. Another source of relevant literature, and metrics, is content analysis. Content analysis [52] typically involves coding (labeling) units of information within a message (such as text, audio or video). Once the units of information are identified, assigning labels is a form of text categorization. Content analysis is generally performed by human analysts, often as part of social science research. Establishing the reliability of the coding process commonly involves comparison of coding data produced by different coders

(consistency) [52]. Similar tasks, and investigations into interobserver agreement, can be found in literature in various fields, such as behavioral research [117] and linguistics [118, 119].

Next we compare identifying semantic component instances to unitization in content analysis, which is identifying the boundaries of information units within a document when they are not predefined (Section 6.2.2). Then we consider how some other tasks that identify, and sometimes label, subdocuments relate to semantic component indexing. Examples of related subdocument tasks we will consider are: identification of the boundaries in text where the topic changes, recognizing text that represents a novel aspect of a topic compared to those aspects that have already been found, and identifying specific elements of information that can answer a question or fill a slot in an information extraction structure (Section 6.2.3). Finally, we discuss keyword indexing (Section 6.2.4).

6.2.1. Text Categorization

Sebastiani distinguishes *hard* categorization (a binary decision with respect to membership or non-membership in a category) from *ranked* categorization (an estimation of appropriateness of membership in each category). He also distinguishes *single-label* categorization (a document belongs to one category) from *multilabel* categorization (a document can be placed in zero to |C| categories, where C is the set of categories) [120]. Multilabel categorization can also sometimes be viewed as |C| binary classification problems, where C is the set of categories and a document either

belongs to class c_i or its complement (class $\neg c_i$). Treating multilabel categorization as multiple binary classifications is appropriate only if a document's membership in a category is independent of its membership in any of the other categories [120]. Single-label categorization means that membership in each class is not independent and document categorization cannot be treated as a series of binary classification tasks. These distinctions become important when we consider metrics for evaluation of categorization.

Assigning document class is an example of single-label categorization and not multiple binary classifications because, in this chapter, we assume that a document can belong to only one class. (In Chapter 9 we discuss allowing documents to belong to multiple classes). We also assume that the categorization is hard, not ranked.

Keyword indexing is an example of multilabel categorization. Each keyword represents a category to which a document can belong. Here we assume all keywords are equally important, and thus categorization is hard, although some systems do allow keywords of differing importance. Should we treat keyword indexing as multiple binary classification problems? An automated keyword indexing system would algorithmically consider each possible category and could reasonably be treated as multiple binary classifications. Human keyword indexing, from either a large controlled vocabulary or from unrestricted natural language, is somewhat different. When the universe of keywords is large, human indexers do not explicitly consider every possible keyword. Implicit rejection of a category, by not choosing a keyword, should not necessarily be treated as classification equivalent to explicit selection or

rejection. We argue that to do so would imply that all terms in a large keyword universe should be considered possible choices when considering the probability of agreement by chance. In practical terms, we argue that an indexer would not take additional effort to consider every new term if a completely unrelated portion of the indexing vocabulary were (for example) to be doubled in size. Thus, we argue that it is unreasonable to model keyword indexing as involving binary classification with respect to the added terms.

On the other hand, when keyword indexers use a controlled vocabulary it might be reasonable to identify a subset of the controlled vocabulary from which indexing might be treated as multiple binary classification tasks. Lancaster notes that "it seems probable that the greatest consistency would be achieved in the assignment of those terms that might be preprinted on an index form or displayed online ... to remind an indexer that they *must* be used whenever applicable." [22]. Using such limited, well-defined lists of keywords that require explicit consideration could reasonably be treated as independent binary classification tasks.

It might also be reasonable to consider the keywords in a reference standard as terms that the indexer either chose or excluded, even if we do not know whether the indexer explicitly considered each of those terms. A reference standard consists of terms relevant to the document and forms a subset of terms that we can say the indexer should have been considering, even if he did not agree that a particular term should be included. We could extend this same argument to the set of unique terms used by at

least one indexer when comparing consistency, considering it as a universe of terms that indexers either chose or excluded.

Identification of text segments that are instances of a semantic component can also be considered text categorization. Each unit of text identified as belonging to a semantic component instance is placed in a category (i.e., is labeled with the name) representing that semantic component. Because each unit of text can belong to zero, one, or more semantic components, identifying semantic component instances is multilabel categorization. In contrast to keyword indexing, identification of semantic component instances can be viewed confidently as multiple binary categorizations of each text element with respect to the list of semantic component labels. The semantic components comprise a limited number of categories that are explicitly considered by the indexer and for which membership by any given text element is independent from its membership in the other categories.

6.2.1.1. Measuring Accuracy of Single-label or Multilabel Categorization

Automated text categorization is typically evaluated with respect to a reference standard based on expert judgment. The most commonly used performance measures are *recall* and *precision* [120]. Other measures sometimes used are *fallout*, *accuracy*, *error*, and F_1. Figure 6.3 shows a contingency table and uses the contingency table to define performance measures for comparing the decisions of a categorizer to the decisions in a reference standard for a particular category.

	Reference Standard	
Categorization Decision	*Yes*	*No*
Yes	TP	FP
No	FN	TN

Abbreviations:	TP: True positives FN: False negatives	FP: False positives TN: True negatives

Metrics:
Recall = TP / (TP + FN)
Precision = TP / (TP + FP)
Accuracy = (TP + TN) / (TP + FP + FN + TN)
Error = (FP + FN) / (TP + FP + FN + TN)
Fallout = FP/(FP + TN)
F_1 = (2 * recall * precision) / (recall + precision)

Figure 6.3 Calculation of accuracy measures for categorization

Accuracy and error are generally less useful than recall and precision because both accuracy and error have the total number of documents in the denominator. As a result, for rare categories (for which both true positives (TP) and false negatives (FN) are very small), the magnitude of true negatives (TN) can dominate the calculation. Even large changes in TP and FN may have only small effects on accuracy and error, masking differences in performance. Similarly, fallout contains TN in its denominator and is similarly affected when TN is much larger than false positives (FP) [121]. F_1 combines recall and precision and is suitable when the number of categories per document is small compared to the total number of categories [121]. Note that this approach to evaluating categorization results is category-centric. Performance is calculated for each category, where:

$$recall = \frac{number\ of\ times\ a\ document\ is\ correctly\ placed\ in\ the\ category}{number\ of\ documents\ actually\ in\ the\ category} \qquad (1)$$

and

$$precision = \frac{number\ of\ times\ a\ document\ is\ correctly\ placed\ in\ the category}{number\ of\ times\ any\ document\ is\ placed\ in\ the\ category} \qquad (2)$$

The above measures are applicable to either single-label or multilabel categorization, but, by definition, make sense only when comparing a set of categorizations to a reference standard, not when comparing the decisions of "peers" (such as two or more equally trained or equally trusted human indexers).

Recall or precision can be summarized over multiple categories by microaveraging (averaging results of all decisions over all categories) or by macroaveraging (averaging results locally for each category individually then averaging the results for the different categories). Microaveraging weights each document equally whereas macroaveraging weights each category equally, regardless of how many documents are assigned to each category [120, 121]. IR studies most commonly report macroaveraged results, where recall and precision are calculated for each query and then averaged across all queries. Text categorization, on the other hand, is more commonly evaluated using microaveraging [122].

Note that, for single-label categorization, microaveraging will result in the same value for both precision and recall. This result occurs because microaveraging is the sum over all categories of the TP values divided by the sum, over all categories, of either TP + FN (for recall) or TP + FP (for precision):

$$microaveraged\ recall = \frac{\sum_{i}^{|C|} TP_i}{\sum_{i}^{|C|} (TP_i + FN_i)} \tag{3}$$

$$microaveraged\ precision = \frac{\sum_{i}^{|C|} TP_i}{\sum_{i}^{|C|} (TP_i + FP_i)} \tag{4}$$

For both recall and precision, the numerator is identical. The denominator is also the same for microaveraged recall and precision when each document is placed in exactly one category because the sum over all categories of either TP + FN (recall) or TP + FP (precision) is just the total number of categorizations performed. Every FP in one category is a FN in another category.

In the context of evaluating semantic component instances, microaveraging corresponds to weighting each character in each document equally. Macroaveraging corresponds to weighting each semantic component instance (in each document and as indexed by each indexer) equally, regardless of the length of each instance. The choice of microaveraging or macroaveraging depends on the purpose of the data analysis. Because we have no prior experience evaluating semantic component indexing, we are interested in both types of averaging to gain as much understanding of the indexing process as possible.

One issue that is rarely discussed with regard to these calculations is the possibility of having a zero occur in the denominator. Yang includes a caveat in her definitions of these performance measures—that the definition holds if the denominator is greater than zero, otherwise it is undefined—but she does not discuss how to deal with the

undefined values when computing averages [121]. Lewis notes that while an undefined value is unlikely to occur in microaveraging, it may occur when macroaveraging is used. Lewis refers to work by Tague who suggested, in the context of evaluating information retrieval results, that one can either treat 0/0 as 1.0 or throw out the query, and comments that for text categorization one can choose to macroaverage over only those categories for which this situation does not arise, as long as the approach is consistent for all data presented [122].

We expect that not all documents in a class will have instances of every semantic component for that document class. If the reference standard does not have an instance for a particular semantic component, then both TP and FN will equal zero. Calculating recall for that semantic component will result in 0/0, regardless of whether the indexing instance that is being evaluated has text assigned to an instance of the semantic component. In contrast, the calculation of precision does depend on the indexing instance. If the indexer has not included any text in an instance of the semantic component, then FP is also equal to zero and precision = 0/0. If the indexer has included some text in an instance of the semantic component, then FP > 0 for that component and precision = 0.

Macroaveraging can be useful to compare performance among indexers or to use indexing performance to assess the difficulty of indexing particular documents or the difficulty of identifying particular semantic components. When macroaveraging recall values, a value of 0/0 means we have no information about the indexing instance and no information about indexing performance. We propose that the individual semantic

component-document combination should be excluded from the macroaverage calculation when recall equals 0/0.

Macroaveraging precision is different from recall. Not including any text in a semantic component instance, when a semantic component instance does not appear in the reference standard for the same document, is evidence of good indexing performance. Inclusion of text in a semantic component instance, when there is none in the reference standard, is evidence of imperfect indexing performance. This evidence is useful when summarizing data with macroaverages. We therefore propose that precision values of 0/0 (reflecting ideal indexer performance) should be treated as 1.0, but only for macroaveraging. Precision results of 0/FP all resolve to zero, regardless of FP values. Unfortunately, this convention does not allow us to discriminate between errors of different magnitude, but we do not have a better solution.

Recall and precision (and the related measures shown in Figure 6.3) are often used to evaluate automated systems, either for information retrieval or for text categorization. Such systems are deterministic and there is no notion of the systems categorizing or retrieving documents randomly. Only the test parameters, such as queries, documents, and categories, can be treated as samples of a larger population. Reporting tests of statistical significance testing is relatively rare in the IR and text categorization literature, but a few authors have discussed appropriate methods for determining the statistical significance of performance differences between IR systems for some number of queries [123-125] or between text categorization systems for

some number of documents, categories, or document-category pairs [126], where each system's performance has been measured by comparing it to a gold standard.

When measuring the accuracy of human indexers, one might consider two possible approaches to statistical testing. First, one might consider using a statistical test designed for 2 x 2 contingency tables, such as the Chi-Square (X^2) test. X^2 tests whether the proportion of objects in each category is significantly different from the expected proportion of objects in each category. However, for categorization correctness, X^2 uses the same values as the recall and precision calculations (the numbers in the cells of the 2 x 2 contingency table) and compares the numbers in each category, not the correctness, and thus actually provides less information than recall and precision. Two different performances, one with better recall and one with better precision (the same number of TP and TN but swapped values for FP and FN), could yield the same X^2 value. Hence the X^2 test is not particularly useful for evaluating accuracy of semantic component indexing. A second approach is to consider the recall or precision value for each semantic component instance as a sample proportion (recall is the proportion of all characters belonging in a semantic component instance that are correctly indexed) from the population of all documents, all indexers, or all semantic components of interest. This approach allows us to aggregate the individual values for recall and precision using microaveraging and macroaveraging, and to examine the variance of recall and precision values when we macroaverage. We can also use standard statistical tests, such as those suggested by Yang [126], to compare

the recall or precision between two groups, such as the indexing instances for two document types or by two indexers.

6.2.1.2. Measuring Consistency of Single-label Categorization

For measuring consistency of single-label categorization, we consider the content analysis literature and some related literature from computational linguistics. A commonly used approach to estimate intercoder reliability is to measure percentage agreement [127] (number of units on which all coders agree/total number of units), a measure that has been criticized for failing to account for agreement by chance [52, 118, 127-129]. Krippendorff has reviewed a larger selection of coefficients of agreement [52, 130] but we consider only three commonly used coefficients here: Cohen's Kappa (C_κ) [131], Scott's Pi (S_π) [128], and Krippendorff's Alpha (K_α) [52, 130]. All three coefficients can be expressed as:

$$Agreement = \frac{A_o - A_e}{1 - A_e} = \frac{observed\ agreement - expected\ agreement}{1 - expected\ agreement} \qquad (5a)$$

or equivalently as:

$$Agreement = 1 - \frac{D_o}{D_e} = 1 - \frac{observed\ disagreement}{expected\ disagreement} \qquad (5b)$$

Perfect agreement (no disagreement) results in a value of one, whereas agreement that is no better than what can be expected to occur by chance results in a value of zero. Values less than zero indicate systematic disagreement, that is, more disagreement than would be expected to occur by chance. The differences among the three

coefficients arise from the calculation of expected agreement. Expected agreement (agreement by chance) depends on the number of categories, and also on the frequency with which each category is used. Total probability of chance agreement is the sum of the probabilities of chance agreement for each category.

Although various authors have proposed agreement coefficients that treat all categories as equally likely, such as Bennett's S [130], a coefficient that assumes equal distribution of categories is inappropriate for settings in which a nonuniform distribution is expected, as would be the case in a large document collection. Scott points out that minimum chance agreement occurs when each coder uses each category with equal frequency [128]. C_κ determines expected agreement based on the proportion of units that each coder places in each category. Cohen claims that C_κ assumes that the proportional allocation of units to categories (i.e., tendency to prefer certain categories over others) is part of the coders' disagreement [131]. The element of chance merely determines which units are placed in which categories. For example, if coder *A* places 20% of all documents in category *S* and coder *B* places 80% of all documents in category *S*, then the expected agreement for category *S* is .2 * .8 = .16. In other words, the two coders would be expected to agree on category *A* 16% of the time. Krippendorff points out that C_κ measures a correlation between the coders, and that greater disagreement between coders with regard to marginal frequencies (the frequency that an indexer uses each category, which is shown in the marginal column of a contingency table) actually results in a higher coefficient of agreement [52].

When marginal frequencies are the same for all coders, C_κ is identical to S_π, but when marginal frequencies differ, C_κ exceeds S_π [130].

S_π and K_α treat all coders as if they are interchangeable and calculate expected agreement based on an underlying "true" frequency of each category. The true frequency for each category (i.e., the true class for each document) is unknown and must be estimated. S_π and K_α assume that the coders' actual use of each category (the mean frequency calculated from all coding samples) represents an estimate of the true frequency and therefore the probability of a random coder choosing that particular category. K_α is nearly identical to S_π, except that K_α calculates the probability of a coder choosing a category based on sampling without replacement. This difference results in K_α exceeding S_π by an amount that is dependent on the sample size. At large sample sizes, K_α approaches S_π asymptotically. All three measurements are affected by the underlying prevalence of categories [119]. If there are two categories, and nearly all the documents fall into a single category, the probability of agreement by chance is so high that even extremely high levels of observed agreement result in agreement measures near zero.

Figures 6.4 and 6.5 show example data and equations for calculating the three coefficients. For simplicity of illustration we use two observers[22] and three categories. Figure 6.4 shows an agreement table for calculating C_κ and S_π. The cells in the

[22] Measures of agreement are used in various fields to quantify agreement on a variety of categorization tasks, such as coding, rating, diagnosing etc. For simplicity and generality we sometimes use the term *observer* instead of coder or indexer.

agreement table for C_κ and S_π contain the fraction of the observed values that correspond to the values chosen by each observer. For example, the cell labeled

Agreement table for calculating C_κ and S_π:

		Observer B			
	Category	1	2	3	P_A(category)
Observer	1	p(1,1)	p(1,2)	p(1,3)	$p_A(1)$
A	2	p(2,1)	p(2,2)	p(2,3)	$p_A(2)$
	3	p(3,1)	p(3,2)	p(3,3)	$p_A(3)$
	P_B(category)	$p_B(1)$	$p_B(2)$	$p_B(3)$	1.0

Coincidence matrix for calculating K_α:

Category	1	2	3	num in category
1	obs (1-1) pairs	obs (1-2) pairs	obs (1-3) pairs	obs (1-*) pairs
2	obs (1-2) pairs	obs (2-2) pairs	obs (2-3) pairs	obs (2-*)pairs
3	obs (1-3) pairs	obs (2-3) pairs	obs (3-3) pairs	obs (3-*)pairs
num in category	obs (1-*) pairs	obs (2-*) pairs	obs (3-*) pairs	total obs pairs

Example data table:

Item	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Observer															
A	c	a	c	c	a	b	b	a	c	c	b	b	c	c	b
B	c	a	c	b	b	b	c	a	c	c	c	c	c	c	b

Agreement table for calculating C_κ and S_π, populated from the example data:

		Observer B			
	Category	a	b	c	P_A(category)
Observer	a	0.133	0.067	0.000	0.200
A	b	0.000	0.133	0.200	0.333
	c	0.000	0.067	0.400	0.467
	P_B(category)	0.133	0.267	0.600	1.0

Coincidence matrix for calculating K_α, populated from the example data:

Category	a	b	c	num in category
a	4	1	0	5
b	1	4	4	9
c	0	4	12	16
num in category	5	9	16	30

Figure 6.4 An agreement table, a coincidence matrix, and an example data set

$$C_\kappa = \frac{A_o - A_e}{1 - A_e} = \frac{\sum_i p(ii) - \sum_i (p_A i * p_B i)}{1 - \sum_i (p_A i * p_B i)} \quad (6)$$

$$S_\pi = \frac{A_o - A_e}{1 - A_e} = \frac{\sum_i p(ii) - \sum_i (\frac{p_A i + p_B i}{2})^2}{1 - \sum_i (\frac{p_A i + p_B i}{2})} \quad (7)$$

$$K_\alpha = \frac{A_o - A_e}{1 - A_e} = \frac{\sum_i o_{i-i} - \sum_i \frac{(o_{i-i})(o_{i-i} - 1)}{n-1}}{n - \sum_i \frac{(o_{i-i})(o_{i-i} - 1)}{n-1}} \quad (8)$$

For the example data in Figure 6.4:
C_κ = 0.448
S_π = 0.443
K_α = 0.461

Figure 6.5 Equations for calculating C_κ, S_π and K_α

"p(1,2)" contains the proportion of the total items that observer A placed in category 1 and that observer B placed in category 2. Figure 6.4 also shows a coincidence matrix that can be used for calculating K_α. The cells in the coincidence matrix for K_α contain the number of pairs of values for which one of the two observers assigned one category and the other observer assigned the other category. For example, the cell labeled "obs (1-2) pairs" contains the number of observations for which one observer placed the item in category 1 and the other observer placed the item in category 2. Because "obs (1-2) pairs" is the number of items that would be represented in both the p(1,2) cell and the p(2,1) cell in the agreement table, the total number of observed pairs is double the number of observations. Figure 6.4 also shows an example data set, with the categorization for each of 15 items by two observers, and both an

agreement table and a coincidence matrix that have been populated from the example data set. Figure 6.5 shows the equations for calculating each agreement coefficient based on either the agreement table or the coincidence matrix. Figure 5 also shows the values calculated for each of the three coefficients from the example data set. Note that, because the marginal frequencies are unequal, C_κ exceeds S_π. Also note that K_α exceeds the other two values because the data set is small.

C_κ and S_π were both originally proposed for measuring agreement between two observers. Fleiss subsequently generalized C_κ to measure nominal scale agreement between more than two observers when the same number of observers categorize each object [132]. This same generalization to more than two observers is described in a statistics text by Siegel and Castellan as the "kappa statistic," although when applied to a case with only two observers it is actually identical to S_π, not C_κ, in that it assumes the same probability of assignment to a given category for all raters [133]. The similarity in names between C_κ ("Cohen's kappa") and the kappa statistic for multiple observers is a potential source of confusion.

S_π and K_α are conceptually similar and result in similar values. The use of sampling without replacement (instead of sampling with replacement) by K_α results in a small adjustment for sample size, which is reasonable. Although both are metrics are suitable for measuring consistency of single-label categorization, we prefer K_α because of its ability to handle unequal numbers of observers per item being categorized. This means we can handle a different number of indexing instances per document.

6.2.2. Unitization In Content Analysis

As noted at the beginning of Section 6.2, content analysis is the systematic evaluation of the content of various forms of communication and usually involves coding (labeling) units of information. For some tasks, the unit to be labeled is obvious (and predefined), such as documents to be indexed, sentences to be coded, or patients to be diagnosed. For other tasks, identifying the boundaries of an information unit (within a message) is an important part of the analysis that precedes assigning a code, or label, to the unit. For example, an analyst may need to identify text segments that pertain to a research question or to identify episodes or events of interest [52]. Krippendorff calls the process of deciding what is included (or excluded) in each unit *unitization*. For each labeled category, unitization partitions text into *sections*, where each section is either a *unit* or a *gap*.

Assessing the consistency of coding output among two or more coders who unitized the message requires comparing the unitization part of the task as well as the codes applied. The sections produced by unitization have lengths, locations (described by the starting and ending boundaries), and a binary value that indicates whether the section is a unit or a gap. Depending on the nature of the content analysis task, unitization can be performed with respect to one or more categories. Each category is treated separately in the agreement calculations.

Semantic component indexing is analogous to content analysis in that: (1) instances of a semantic component can have zero or more segments; (2) we consider each semantic component separately; and (3) segments produced by semantic

component indexing can be characterized by length, location, and whether the segment is in the instance (is a unit) or not in the instance (is a gap). One difference between content analysis and semantic component indexing is that different segments belonging to a particular semantic component instance are always discontiguous, by definition. If segments belonging to a semantic component instance are adjacent, they are automatically combined to become a single segment because there is no reason to treat them as distinct segments. In content analysis, it can be useful to distinguish separate but adjacent segments. For example, one might want to count the number of episodes of violence in a transcript or recording. Episodes of violence might be distinct but adjacent. Therefore, adjacent segments are permitted and are treated as distinct units when calculating agreement.

Krippendorff has developed an extension of K_α to measure agreement (consistency) with respect to unitization performed by different analysts [52]. We discuss K_α for unitization in more detail in Section 6.3.3. The difference between the two tasks, content analysis and semantic component indexing, with regard to adjacency becomes important when we analyze the behavior of K_α for unitized data.

6.2.3. Other Subdocument Tasks Similar To Semantic Component Indexing

So far, we have considered semantic component indexing as first classifying whole documents (assigning document class) and, then, categorizing each unit of text (such as a character) within a document. In this section we discuss other tasks similar to identifying the text that belongs to a semantic component instance. In this group we

include linear text segmentation, passage retrieval, question answering, novelty detection, and information extraction. All these tasks involve selecting and manipulating text at a subdocument level. We consider the evaluation methods that have been used for these tasks and note that the unit of evaluation is a critical determinant of the evaluation approach. In particular, we compare the character-based classification approach to the evaluation units and metrics traditionally used for these other subdocument tasks.

6.2.3.1. Linear Text Segmentation

As discussed in Chapter 2, text segmentation divides text into sections, placing boundaries to indicate changes in topic or discourse element. Although text segmentation can be linear or hierarchical, we focus on linear segmentation because it is more similar to identifying semantic components in text. In text segmentation, the segments cannot overlap and possible boundaries are sometimes restricted to occurring only between paragraphs, between sentences, or between phrases. We do not impose such restrictions in semantic component indexing. If we consider one semantic component at a time, then identifying the segment(s) of text that constitutes an instance of that semantic component requires finding segment boundaries. Evaluating boundary placements for the segments is similar to evaluating text segmentation efforts, although we allow boundary placement to occur between the smallest resolvable units, such as characters.

Text segmentation research usually compares the results of an automated system to a reference standard, which typically is either a concatenation of news stories or a manually produced standard. In the first case, the automated system tries to determine the story boundaries, which are known to the researchers. In the second case, establishment of an adequate standard may involve evaluating the consistency of human judges or developing a consensus standard. First, we consider how to measure accuracy, that is how to compare an automated system to a standard, however it is derived. Then, we consider methods for considering multiple human judgments for the purpose of making a standard.

One approach to measuring accuracy of text segmentation algorithms is to define standard IR metrics, such as recall and precision, in terms of boundaries instead of documents. Boundaries are placed correctly or incorrectly or they are missed (no boundary is placed where there should be a boundary). This approach is taken by Passonneau and Litman [134] and by Hearst [135]. Ponte and Croft take a similar approach except that, in addition to calculating recall and precision based on exact matches, they also report recall and precision calculated with a partial-match function that gives some credit for near misses [53]. Recall and precision of boundaries is not likely to be useful for assessing the accuracy of semantic component indexing for two reasons. First, recall and precision do not account for chance placement of correct boundaries. Second, if a boundary placement is off by, say, one word, the semantic component instance may have nearly as much usefulness as a completely correct instance because the text included in the instance is almost the same as the text in the

reference standard instance. Near misses should be credited relative to the nearness of the boundary placed by an indexer to the boundary occurring in the reference standard.

Beeferman, Berger, and Lafferty introduce an error metric, P_k that considers both near misses and the probability of random agreement. Briefly, they calculate the probability that two sentences drawn randomly from a corpus are correctly identified as belonging to the same document (segment), or to different documents (segments) [136]. Pevzner and Hearst criticize P_k as penalizing false negatives more than false positives, overpenalizing near misses, and being sensitive to variations in segment size. Pevzner and Hearst propose a modified metric called WindowDiff [137]. Both P_k and WindowDiff are more appropriate for semantic component indexing than recall and precision of boundary placements, but they still have some drawbacks that are related to differences in the tasks being evaluated. P_k requires setting a parameter *k* based on the mean segment length and is sensitive to variability in segment size. Segment sizes can be highly variable in semantic component instances. WindowDiff is less sensitive to segment size, but operates by calculating the number of boundaries between the ends of a fixed length probe.

Passonneau and Litman measured agreement among human subjects performing discourse segmentation by calculating percent agreement, which they defined as "the ratio of observed agreements with the majority opinion to possible agreements with the majority opinion" [134]. In their study with seven human subjects, a majority opinion was the placement (or nonplacement) of a boundary by four or more subjects at each of the possible boundaries (between marked prosodic phrases). Not only does

this measurement not account for chance, but their methodology guarantees at least 4/7 (57%) agreement on all boundaries and 3/7 (43%) agreement on all nonboundaries. Hearst also compared boundary placements by individuals to group decisions (in this case placement of a boundary by only three of seven judges was required to establish a "real" boundary), but she reported the kappa statistic, assuming the overall frequencies for boundaries and nonboundaries as estimates for the probability of agreement by chance [56]. In these two studies, human segmenters were restricted to placing boundaries between prosodic phrases (Passonneau and Litman) or paragraphs (Hearst), limiting the number of possible boundary locations. If boundaries can be placed between characters, the number of nonboundary locations would be so high that the probability of agreement by chance on nonboundaries would be extremely high, making agreement measurements uninterpretable. Calculating only agreement on boundaries would have the opposite problem. The probability of agreement by chance on exact boundaries would be too low to have any meaning.

There are three important and related conceptual differences between segmentation and semantic component indexing:

1. Text segmentation makes an underlying assumption that topics, documents, or discourse elements are distinct if they occur in different segments whereas multiple non-adjacent segments can comprise a single semantic component instance. The presence of multiple boundaries between two units of text does not mean that the units of text are not in the same semantic component instance.

2. The number of boundaries placed during semantic component indexing is not, of itself, important. Boundaries have an important function in semantic component indexing, but they are not the primary unit of interest. What matters is whether a term matching a query is correctly included in a semantic component instance. The occurrence of multiple segments, and therefore multiple boundaries, in that instance is unimportant.
3. Agreement on exact boundaries has less importance whereas agreement on inclusion of particular text elements in an instance is very important for semantic component indexing.

As a result of these differences, none of the metrics described for evaluating text segmentation is likely to be useful for evaluating semantic component indexing. Semantic component indexing is more usefully modeled as classification of each text unit rather than as boundary placement.

6.2.3.2. Passage Retrieval

Passage retrieval, as discussed in Chapter 2, splits documents into subdocuments (using a variety of techniques) and computes the similarity of each passage to the query. Passages can be used for retrieval in two ways. First, the unit retrieved can be a document, with documents being ranked based on the computed relevance scores for passages within the document. Second, the unit retrieved can be an individual passage. Documents are usually split into passages automatically, so evaluations are designed to determine the effect of using passages on retrieval. When the unit

retrieved is a document, standard IR metrics apply and this task does not inform our work with semantic components. When the unit retrieved is a passage, and splitting (segmentation) is semantic, then the effect of generating passages on retrieval performance is estimated by comparing returned passages to a reference standard.

Two tracks from TREC have taken such an approach, the High Accuracy Retrieval from Documents (HARD) Track, and the Genomics Track. The 2004 HARD Track used query metadata and a brief interaction with the user to gather additional data about the information need and experimented with passage retrieval as well as document retrieval [138]. The 2006 Genomics Track required systems to return passages that contained answers to questions [139]. Both tracks evaluated system performance by using variations on existing IR metrics. These variations use the proportion of characters in each returned passage that coincide with characters in the gold standard passages (determined by human judges) for the same topic. The HARD Track used passage-level versions of recall, precision, F score, R-precision and b-pref for evaluation [138]. The Genomics Track developed a passage-level variation on Mean Average Precision (MAP). Because the original passage MAP score can be manipulated by shortening all passages, or by breaking passages in half, a second version, PASSAGE2, concatenates the output of passages, so that each character is treated as a ranked document [139]. This character-based approach is analogous to the character-based approach we have discussed for evaluating the accuracy and consistency of semantic component indexing.

6.2.3.3. Question Answering, Novelty Detection, and Information Extraction

A number of subdocument tasks are related to passage retrieval, and can use text segmentation as an intermediate step, but the target output is more specific than just a passage. In this section we briefly review evaluation methods commonly used for question answering, novelty detection, and information-extraction systems.

Question-answering systems try to return an answer, or parts of the answer. Some question-answering systems return a passage containing an answer; we consider such systems in the passage-retrieval category. Other systems try to return exact answers. Answers can be facts or lists of elements that together constitute an answer. Several TREC tracks (e.g., Question Answering [140], Interactive [100], and Genomics [139]) have had question-answering tasks in which system output was evaluated in three ways: (1) using the fraction of questions for which the answers were judged (by human judges) as correct (for factoid questions), (2) using instance recall and instance precision (for list questions, for which each distinct instance of an answer to a question should be returned), or (3) using an aspect-level version of MAP.

Novelty detection is closely related to question answering. The goal is to return relevant text units (documents or subdocuments) that contain information about the query, but without redundancy [59, 141]. Ideally, each new document in a ranked list should contain information that is relevant and also novel relative to the documents already returned. Conceptually, each ranked element can be treated as containing one or more aspects of the answer. The aspects are then judged for novelty. As with question-answering tasks, the unit being judged is an answer, or part of an answer.

Variations on existing retrieval metrics compare the retrieved elements to a gold standard, assessing the proportion of a complete answer that has been found and the precision of the ranked list of answer elements (which may consider information that is redundant to be equivalent to nonrelevant).

Systems for information extraction identify certain types of information in unstructured text, such as entities, facts, and events, and extract the information into databases or templates. Evaluation is based on the correctness and completeness of elements extracted into slots (compared to a reference standard) and can be reported in terms of recall and precision, or error rates [142, 143].

Common to all three of these tasks is that the evaluation is based on an answer, or part of an answer, which is really a concept, not a segment of text. Comparing the output of these systems to a reference standard is fundamentally different from evaluating the accuracy or consistency of semantic component indexing. Semantic component indexing identifies information pertaining to a semantic component, not a particular question. Metrics based on recall or precision of returning text that represents an answer element are suitable for the three tasks described, but they do not help us decide how to evaluate semantic component indexing. We conclude that methods to evaluate such systems are not likely to help us determine how best to evaluate semantic component indexing because the unit of interest for these tasks is an answer, or part of an answer, not a segment of text (although the answer may be derived from, or composed of, a segment of text).

6.2.4. Keyword Indexing

In this section we consider how to evaluate keyword indexing. The library and information science community has studied both inter-indexer consistency and the quality of automated indexing results compared to human expert indexing. For measuring the accuracy, or correctness, of keyword indexing relative to a reference standard, Rolling suggests formulas that are equivalent to recall and precision [116]. Soergel suggests measuring completeness (equivalent to recall) and purity (the proportion of all terms that should have been rejected that were correctly rejected, or in other words TN/(TN + FP)) [144]. Lancaster describes a (weighted) scoring system that adds points for correctly assigned terms and subtracts points for incorrectly assigned terms (terms not in the reference standard) [22].

Several candidate formulas for measuring consistency appear in the indexing literature, but two predominate. We follow Rolling [116] and represent each formula in terms of a, b, and c, where a is the number of terms used by one indexer, b is the number of terms used by the second indexer, and c is the number of terms used in common by both indexers. The two formulas are:

$$consistency = 2c / (a + b) \tag{9}$$

and

$$consistency = c / (a + b - c) \tag{10}$$

Rolling presents six different formulas for calculating inter-indexer consistency between two indexers [116]. Four of the six formulas represent variations on the theme of percent agreement, consisting of a ratio of *items of agreement*/*all items* and

two formulas consist of a ratio of *items of agreement*/*items of disagreement*. The four formulas based on percent agreement differ with respect to the actual calculations of the numerator and the denominator. Three of the six formulas use c as the number of items of agreement and three formulas use $2c$ as the number of items of agreement. Conceptually, we view these two options as corresponding to: (1) considering the items being agreed upon as *indexing terms*, for which c is the logical numerator; or (2) considering *indexing decisions* as the items being agreed upon, in which case each indexer makes c decisions that agree with those made by the other indexer, for a total of $2c$ decisions made by the pair of indexers. Among the six formulas, three variations occur in the denominator. Two of the variations occur in the four formulas based on percent agreement. One variation (two formulas) uses the total number of terms used, including all duplicates, calculated as $a + b$. Another variation (two formulas) uses the total number of unique terms, calculated as $a + b - c$. The third variation occurs in the two formulas for *items of agreement*/*items of disagreement*. For these two formulas, the denominator is the sum of the terms that are not in agreement, calculated as $a + b - 2c$. We can also view the denominator in this third variation as the number of decisions not in agreement.

Rolling recommends using Formula (9), shown above. We interpret Formula (9) as the ratio of the number of indexing decisions in agreement to the total number of indexing decisions. Lancaster [22] and Soergel [144] both recommend using Formula (10), also shown above. Formula (10) calculates the ratio of the number of terms in agreement to the total number of unique terms. Funk and Reid attributed Formula (10)

to Hooper when they used the formula to report indexing consistency for articles from MEDLINE that were inadvertently indexed more than once [28].

All of the keyword indexing consistency formulas we have discussed measure agreement between pairs of indexers. Lancaster recommends calculating consistency for pairs of indexers, then averaging over all pairs to obtain an overall consistency value for a group of indexers [22]. Rolling points out that generalizing directly to multiple indexers from the formula he recommends would ignore the consistency value of terms assigned by 2 to n-1 indexers and proposes an unwieldy formula to try to account for partial agreement [116]. Rolling [116], Lancaster [22], and Soergel [144] all discuss the use of weighting to account for some indexing terms being more important than others, but that is beyond the scope of our interest here.

None of the above formulas take into account the probability of agreement by chance. To our knowledge there is not an existing measure of consistency that is appropriate for multilabel categorization and that accounts for the probability of agreement by chance. As discussed in Section 6.1.2, it is not obvious what should be the basis for calculating expected agreement. Assuming that all categories in a large vocabulary are equally likely to be chosen is not a sensible choice. If we consider all keywords that were assigned at least once as the universe of keywords for the purposes of calculating expected agreement, we would still have to derive an expected distribution for the number of categories (keywords) assigned per document in addition to an expected distribution for use of the categories (keywords).

One possible approach is to assess consistency in two different ways: (1) calculate consistency based on multiple binary classifications, using all keywords that were assigned at least once as the universe of keywords and applying a measure of agreement appropriate for binary categorization, such as K_α; and (2) calculate one or both of the traditional keyword consistency measures described above in order to compare results with previous studies of indexing consistency. The design and goals of a study might determine whether indexing terms or indexing decisions are the primary unit of consistency and thereby determine a choice between the two keyword indexing metrics (Formulas 9 and 10 above).

6.3. Implementation and Analysis of Krippendorff's Alpha

Krippendorff's alpha (K_α) is a family of related metrics that has been developed by Klaus Krippendorff over a number of years. All the versions of K_α are for assessing reliability of coded data and follow the general form of:

$$\alpha = 1 - \frac{D_o}{D_e} \tag{11}$$

where D_o is the observed disagreement among observers and D_e is the disagreement expected to occur by chance. D_e is calculated by using the data from all observers to estimate a "true" distribution of data in categories that can be randomly sampled [52, 145] . K_α is a generalization of S_π for nominal data, which is data resulting from assignment of objects to categories that do not have a defined ordering, such as gender or literary genre. Krippendorff has extended K_α to handle ordinal, interval, ratio and

specialized kinds of data and to handle incomplete data (missing values) and any number of observers [145]. Krippendorff has also developed a version for comparing unitized data resulting from content analysis. It is the unitized version of K_α that first attracted our attention as a metric that might be useful for comparing instances of semantic component indexing. We have not found any other tasks comparable to semantic component indexing that have suitable metrics for calculating consistency of segment identification that treated the segment as the primary unit for evaluation.

We have implemented the nominal, binary, and unitized versions of K_α as Java programs that read input data files for semantic component indexing instances. In this section we describe the equations and algorithms for calculating K_α in some detail. The equations and algorithms are all based on publications by Krippendorff [52, 130, 145], and we tested the implementations by reproducing the various example results in those papers. In addition, we describe a detailed analysis of the response of the unitized version of K_α to various changes in data characteristics and explain why that analysis caused us to reject the unitized version of K_α for calculating similarity of semantic component indexing instances.

Figure 6.6 summarizes the calculation of K_α and follows the description given by Krippendorff [52]. The difference function δ_{ij}^2 expresses the difference between a pair of values. The prefixed *metric* subscript indicates that its calculation depends on the

Create an *m* by *r* reliability matrix, where m is the number of observers, r is the number of objects being categorized, and each cell c_{ij} in the reliability matrix contains the value assigned by observer i to object j.

Create a *v* by *v* matrix of observed coincidences, where v is the number of distinct values occurring in the reliability matrix. Each cell c_{ij} in the observed coincidences matrix represents occurrences of pairs of values for which one observer assigned value i to an object and another observer assigned value j to the same object. To create the matrix, one first calculates the number of pairs of each v_i-v_j combination for each object being categorized. If all m observers assign value v_i to an object, then there are $m(m-1)$ v_i-v_i pairs. If, instead, p observers assign v_i and q observers assign v_j and $p + q = m$, then there are $p(p-1)$ v_i-v_i pairs, $q(q-1)$ v_j-v_j pairs, $p*q$ v_i-v_j pairs, and $p*q$ v_j-v_i pairs. The observed coincidence matrix should contain one entry contributed by each value (not one entry contributed by each pair) and therefore the contribution of each collection of v_i-v_i pairs for a given object is scaled by multiplying the number of v_i-v_i pairs by $1/(m_u - 1)$ where m_u is the number of values actually occurring for object u (or, in other words, the number of observers who categorized object u). This factor not only scales the contribution of each value, it also adjusts the calculation for missing values so that each value contributes based on its participation in $m_u - 1$ pairs. Note that if an object has only been categorized once (values from the other $m - 1$ observers is missing data) then there are no comparisons to be made for that object (no possible pairs) and the number of coincidences must be zero for that object.

Compute K_α as:

$$\alpha = 1 - \frac{Average_{\ metric}\delta_{ij}^2 \ within\ all\ units}{Average_{\ metric}\delta_{ij}^2 \ within\ all\ data} = 1 - \frac{\sum_i \sum_j o_{ij\ metric}\delta_{ij}^2}{\sum_i \sum_j e_{ij\ metric}\delta_{ij}^2} \qquad (12)$$

Figure 6.6 Calculation of K_α

metric that is most appropriate for a particular kind of data (whether it is nominal, ordinal, interval, ratio, or unitized). For nominal and binary data, the difference function is simple: $\delta_{ij}^2 = 0$ if and only if $i = j$, and $\delta_{ij}^2 = 1$ if and only if $i \neq j$. The numerator expresses the number of values in the coincidence matrix for which a categorization decision differs (the observed values disagree) and the denominator expresses the number of values for which the categorization decision can be expected to be different (the expected values disagree) if the categorization decision is random. For nominal and binary data, Equation (12) can be expressed as:

$$\alpha = 1 - \frac{n - the\ number\ of\ values\ in\ agreement}{n - the\ number\ of\ values\ expected\ to\ be\ in\ agreement} \qquad (12b)$$

where n is the total number of values that could agree (equal to twice the number of pairs).

A more concrete expression of Equation (12) for nominal and binary data is:

$$\alpha = 1 - (n-1)\frac{n - \sum_i o_{ii}}{n^2 - \sum_i n_i^2} \tag{13}$$

where o_{ii} is the number of times that a pair of observers used category i, or in other words, the number in cell c_{ii} of the coincidence matrix. Equation (13) is derived from the preceding expression as shown in Figure 6.7.

The difference function for unitized data is based on comparing both the categorization (coding or labeling of text) and the unitization that results in partitioning the text into segments that belong to a unit and segments that do not belong to the unit (also called gaps). The difference function for unitization is discussed in more detail in Section 6.4.3.

6.3.1. K_α for Nominal Data

In our work, assignment of document class and categorization of documents by assignment of indexing keywords both produce nominal data. The first step in calculating K_α for nominal data is creation of the m by r reliability matrix for m indexers and r documents. The second step, creation of the observed coincidence matrix, is implemented by first creating a coincidence matrix for each document, then

Derivation of equation (13):

The number of values in agreement is simply the sum of the values in the diagonal cells of the coincidence matrix c_{ij} where $i = j$. Thus the numerator, D_o, is $n - \sum_i o_{ii.}$ The number of values expected to agree is calculated assuming that the frequency of a value in the entire data set, n_i for category i, (reflecting the categorization decisions of all the observers) represents the best estimate of the "true" frequency of i and that calculation of expected agreement is based on random selection without replacement. Therefore the number of values for category i expected to be in agreement, e_{ii} = n * (frequency of i to start)(frequency of i after i selected) = n *$(n_i/n)(n_i - 1)/(n - 1) = n_i(n_i - 1)/(n - 1)$. The total number of values expected to be in agreement for v categories is:

$$n_1(n_1 - 1)/(n - 1) + n_2(n_2 - 1)/(n - 1) + ... + n_v(n_v - 1)/(n - 1).$$

The denominator, D_e, is therefore:

$$n - ((n_1^2 - n_1)/(n - 1) + (n_2^2 - n_2)/(n - 1) + ... + (n_v^2 - n_v)/(n - 1))$$

Multiply the first term, n, by $(n - 1)/(n - 1)$ and rearrange to get:

$$n(n - 1)/(n - 1) - ((n_1^2 - n_1 + n_2^2 - n_2 + ... + n_v^2 - n_v) / (n - 1))$$
$$= (n(n - 1) - (n_1^2 - n_1 + n_2^2 - n_2 + ... + n_v^2 - n_v)) / (n - 1)$$
$$= (n^2 - n - (n_1^2 + n_2^2 ... + n_v^2 - n_1 - n_2 .... - n_v)) / (n - 1)$$
$$= (n^2 - n - (n_1^2 + n_2^2 ... + n_v^2 - n)) / (n - 1)$$
$$= (n^2 - (n_1^2 + n_2^2 ... + n_v^2)) / (n - 1)$$

We can therefore express the denominator as:

$$(n^2 - \sum_i n_i^2) / (n - 1).$$

Multiplying the equation $(n - \sum_i o_{ii}) / ((n^2 - \sum_i n_i^2) / (n - 1))$ by $(n - 1)/(n - 1)$ results in Equation (13).

Figure 6.7 Derivation of equation for calculating K_α for nominal and binary data

summing the corresponding cells from the coincidence matrices for individual documents to populate a final coincidence matrix. The third step, calculating K_α, proceeds according to Equation (13), where n is the total number of values being compared, o_{ii} is the number of document classifications in agreement for category i (a document class or a keyword assignment), and n_i is the number of times any observer classified a document as belonging to category i. One can also find n_i by adding all the values in the coincidence matrix for either row i or column i. Note that n is equal to twice the number of pairwise comparisons used to calculate the coincidence matrix because a value for i is counted once in o_{ij} and again in o_{ji}, where o_{ij} is the number of

times observer A assigned a document to *i* and observer B assigned the document to *j* and o_{ji} is the number of times observer A assigned a document to *j* and observer B assigned the document to *i*.

6.3.2. K_α for Binary Data

Binary data is a special form of nominal data that results from assigning objects to exactly one of two classes. In semantic component indexing, each unit in each document undergoes binary classification with respect to each semantic component for that document's class. Each unit of the text is classified as either belonging to the semantic component instance or as not belonging to the semantic component instance. As discussed in Section 6.1.1.2, we use characters as the basic unit of classification for semantic component indexing.

For calculating binary K_α for semantic component indexing instances, we use one coincidence matrix for each of the *c* characters in the document, then sum the *c* coincidence matrices to populate a final coincidence matrix. The final step is to use Equation (13) to calculate K_α as described for nominal data in the preceding section.

6.3.3. K_α for Unitized Data

Computation of K_α for unitized data follows the same outline as K_α for nominal and binary data, calculating both observed disagreement and expected disagreement, but is more complex than for nominal and binary data. The difference function compares both the categorization (coding and labeling of text) and the unitization that

partitions the text into units and gaps, but it does so by comparing the units and gaps for a given category identified by one observer to the units and gaps for the same category identified by another observer. To calculate D_o, we compare each section (a unit or a gap) identified by one observer to each section identified by another observer, and we repeat this over all pairs of observers. D_e is intended to reflect comparisons between all possible unitizations that could be derived from the overall number of sections identified by all observers, the lengths of the sections, and whether each section is a segment or a gap. The distribution of sections among observers and the positions of the observed sections within the document are ignored and treated as characteristics that are subject to randomization. We discuss each part of the computation in more detail below. Several publications discuss calculating K_α for unitized data and provide the equations [52, 145, 146]. The equations and notation used here are the same as in Krippendorff's book about content analysis [52]. The most complete description of how to calculate K_α for unitized data, and the justification for the calculation of expected disagreement, is published in *Sociological Methodology* [146].

Observed disagreement for a particular category *c*, D_{oc}, is calculated as:

$$D_{oc} = \frac{\sum_{i=1}^{m} \sum_{g} \sum_{j \neq i}^{m} \sum_{h} \delta_{cigjh}^2}{m(m-1)L^2} \qquad (14)$$

where *m* is the number of observers, *L* is the length of the document, *i* and *j* are observers, and *g* and *h* are numbers that identify individual sections identified by observers *i* and *j*, respectively. The difference function δ is a measure of the

difference between two sections, so δ_{cigjh}^2 is the squared difference between section *ig* and section *jh* for the unitizations with respect to category *c*. For δ to equal 0, two sections must be identical with respect to starting location, length, and label (a segment or a gap for category *c*). Nonzero values of δ reflect the amount of deviation from a perfect match between two units identified by different observers (in the same document).

Krippendorff defines δ_{cigjh}^2 as shown in Figure 6.8.[23] The first condition for δ_{cigjh}^2 applies if two units (not gaps) overlap but are not identical. When units meet this condition, the value for the difference function is the sum of the squared lengths of the nonoverlapping parts at either end of their intersection. The second condition applies if a unit *ig* is contained in a gap *jh*. If so, then the difference is the squared length of the unit *ig*. The third condition applies if unit *jh* is contained in a gap *ig*, and the

$$\delta_{cigjh}^2 = \begin{cases} (b_{cig} - b_{cjh})^2 + (b_{cig} + l_{cig} - b_{cjh} - l_{cjh})^2 & \text{iff } w_{cig}=w_{cjh}=1 \text{ and } -l_{cig} < b_{cig} - b_{cjh} < l_{cjh}\ ^* \\ l_{cig}^2 & \text{iff } w_{cig}=1,\ w_{cjh}=0 \text{ and } l_{cjh} - l_{cig} \geq b_{cig} - b_{cjh} \geq 0 \\ l_{cjh}^2 & \text{iff } w_{cig}=0,\ w_{cjh}=1 \text{ and } l_{cjh} - l_{cig} \leq b_{cig} - b_{cjh} \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

where

$$w_{cig} = \begin{cases} 0 \text{ iff section } ig \text{ is not a unit (it is a gap with respect to category } c) \\ 1 \text{ iff section } ig \text{ is a unit (with respect to category } c) \end{cases}$$

and where b_{cig} and b_{cjh} refer to the beginning positions of sections *ig* and *jh* and l_{cig} and l_{cjh} refer to the lengths of sections *ig* and *jh*.

Figure 6.8 Definition of the difference function for calculating K_α for unitized data

[23] The first condition, marked with an asterisk, is expressed differently in one source [52], but is likely a typographical error. The equation shown here matches the equation given in another source [145]. Krippendorff has also expressed the condition as $ig \cap jh \neq \emptyset$ [146].

difference is the squared length of *jh*. The fourth condition applies when both sections are gaps, when both sections are units and they overlap perfectly, and when two sections do not intersect at all.

By defining the difference function for partially overlapping units as the sum of the squared lengths of the nonoverlapping portions, agreement about the center of a unit is valued more highly than agreement about the periphery of a unit. In other words, the calculation discounts the disagreement if a nested unit is centered in the containing unit (there is agreement about the center but not about the periphery) compared to the disagreement that is calculated if a nested unit of the same size is positioned at one end of the containing unit. Valuing agreement about the text in the center of a unit might make sense for content analysis if we believe that the center of a text unit reflects the core meaning of the text unit. Assigning different values to agreement depending on its position within a segment does not make sense for semantic component indexing. A search term is either in the semantic component instance or it is not.

It is important to note that if a unit partially overlaps with a gap, it does not satisfy conditions 1, 2, or 3 and so the difference is 0. However, the portion of the unit that does not overlap the gap will overlap another unit and that difference will be part of the overall calculation. Note also that the beginning positions and lengths of sections must be given as integer values from a measurable continuum, such as character offset from the start of a document.

Expected disagreement for category *c*, D_{ec}, is calculated as:

$$D_{ec} = \frac{\frac{2}{L}\sum_{i=1}^{m}\sum_{g} w_{cig}\left[\frac{N_c - 1}{3}(2l_{cig}^3 - 3l_{cig}^2 + l_{cig}) + l_{cig}^2 \sum_{j=1}^{m}\sum_{h}(1 - w_{cjh})(l_{cjh} - l_{cig} + 1) \quad iff\ l_{cjh} \geq l_{cig}\right]}{mL(mL-1) - \sum_{i=1}^{m}\sum_{g} w_{cig} l_{cig}(l_{cig} - 1)} \tag{15}$$

where N is the total number of units identified by all m observers. The equation for D_{ec} assumes that there is a population of sections (units and gaps) of various lengths that is derived from the collection of sections identified by all the observers. It further assumes that those sections can be randomly distributed among the locations in the document and among the m observers. To derive an expected disagreement, the equation iterates through all possible ways in which the document can be unitized with the given number and sizes of the units and gaps and calculates the resulting disagreement.

Instead of repeating the entire justification for the calculation of D_{ec}, we summarize the contribution of each part of the equation. Additional detail is available elsewhere [146]. The numerator sums the differences calculated for each possible unitization. The left hand part of the numerator generates the difference between two overlapping units that meet the first condition in the difference function and results from an algebraic simplification of summing the squares of the non-overlapping portions of the units, summed over all possible ways the units could overlap. The right hand part of the numerator generates the difference between a containing gap (jh) and the nested unit (ig). If the conditions are met (the second or third conditions in the difference function), the difference contributed is the square of the length of the unit, ig. If jh is not a gap, $w_{cjh} = 1$, $1 - w_{cjh} = 0$, and this term does not contribute to the expected difference. The factor ($l_{cjh} - l_{cig} + 1$) is the number of possible positions in

the document in which the two sections could occur. The denominator of D_{ec} adjusts the weighting of D_{ec} so that it corresponds with that of D_{oc}. Each observer contributes some number of sections (units and gaps) whose lengths sum to the length of the document, L. Because the numerator iterates through all sections for each observer, and compares them to the sections for every other observer, we have lengths mL being compared to $(mL - 1)$ lengths, and being considered for up to L possible positions in the document. The term being subtracted adjusts for not comparing units to themselves and for not comparing gaps to gaps.

After implementing K_α for unitized data, we calculated binary and unitized K_α for hypothetical pairs of semantic component instances to compare the behavior of the two metrics when particular data characteristics are manipulated. Unless otherwise specified, the document length was the same for each test and each of the two instances being compared had exactly one unit. The tests addressed the following questions:

1. What is the effect of varying the unit length when the proportion of overlap (length of overlap/length of unit) is fixed?
2. What is the effect of varying the position of a nested unit relative to the containing unit when the position of the containing unit within the document is fixed?
3. What is the effect of increasing the number of nested units when the overall amount of agreement (overlap) is fixed?

4. What is the effect of changing the position within the document of two overlapping units when the relative positions of the units, and the amount of overlap, is fixed?
5. What is the effect of changing the position of two independent units within the document when the length of the two units is fixed?

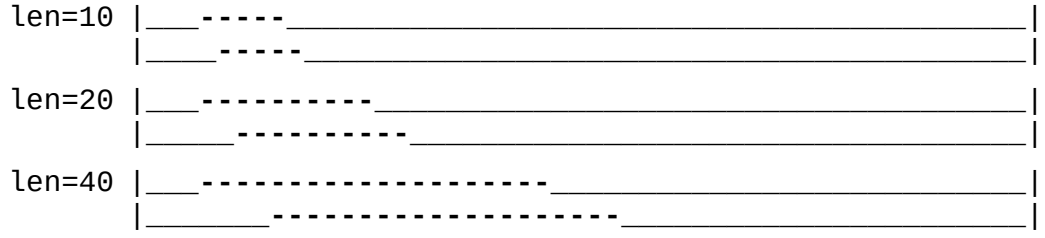
Figure 6.9 illustrates the test data. Each line represents a document of length 100. The dashed (and raised) lines are units, the solid lines are gaps, *len* is the length of the unit, and *st* indicates the starting position of the unit as an offset from the beginning of the document. Our findings are summarized below.

1. As expected, progressively increasing the length of the units, while maintaining an overlap of 80% of unit length, results in a progressive decrease in the value of K_α because the amount of disagreement is increasing. Both versions of K_α behave as expected, although the exact values differ. K_α for three lengths of units, overlapped by 80%:

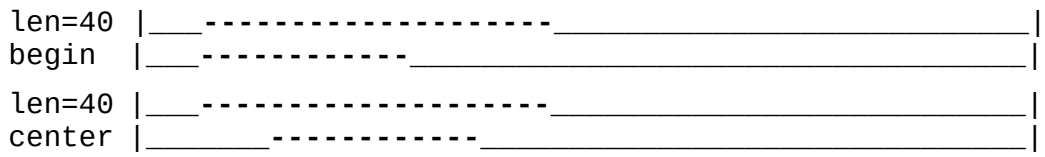
Length of unit	10	20	40
Unitized	0.9484	0.9340	0.8461
Binary	0.7789	0.7513	0.6683

2. For a fixed amount of overlap, binary K_α does not change when the position of the nested unit is changed, as expected. The value of unitized K_α decreases when a unit of length 24 is shifted from overlapping the center of a unit of length 40 to overlapping only the beginning 60% of the other unit. This change occurs largely because the difference function is calculated by

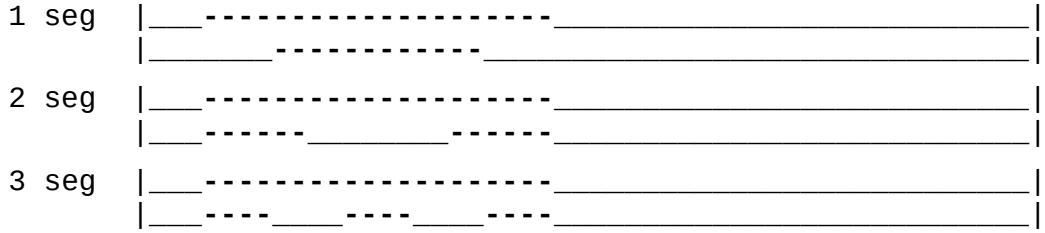
1. Vary unit length; proportion of overlap is 80% of unit length.



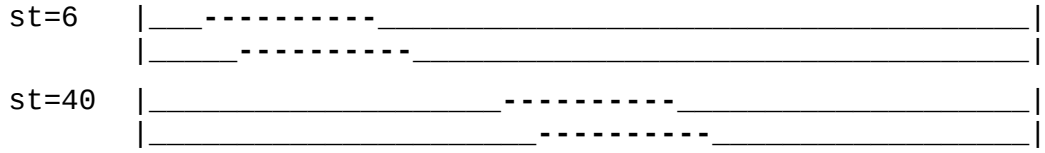
2. Vary position of the nested unit; position of the containing unit is fixed; nested unit is 60% of containing unit.



3. Vary number of nested units; nested units equal (fixed) 60% of containing unit.



4. Vary position of units within document; fixed lengths, overlap (80%), and positions relative to each other.



5. Vary position of units within document; fixed lengths; no overlap.

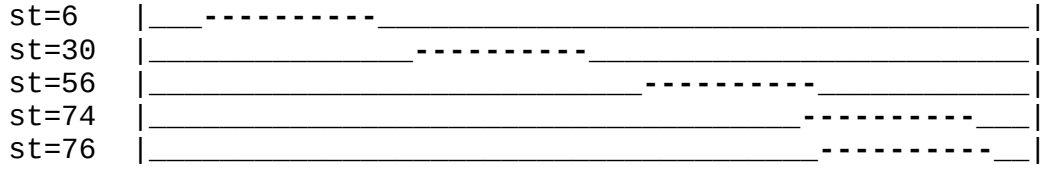


Figure 6.9 Tests to assess the behavior of unitized K_α and binary K_α

summing the squares of the nonoverlapping lengths on either side of the nested unit. For a given total length of nonoverlap between a nested unit and its containing unit, the minimum difference—and therefore the maximum K_α—occurs when the nested unit is centered with respect to the containing unit. However, changes in the sizes of the gaps on either side of the nested unit affect the calculation of expected difference as well. Modifying the difference function, so that the nonoverlapping lengths are first summed and then squared, brings the calculation closer to our desired behavior (that K_α not change when the position of a nested unit is shifted relative to the containing unit but remains nested) but does not eliminate the difference. Some difference persists due to the effect of the length of segments and gaps on D_e, which is discussed under point 4 below.

Position	Center of unit	Start of unit
Unitized	0.8401	0.7133
Binary	0.6342	0.6342

3. For a fixed amount of overlap, binary K_α does not change when the overlap is partitioned into multiple units. The value of unitized K_α decreases considerably when the nested unit is partitioned into multiple units. The change in K_α is mostly due to a substantial increase in the value for the observed difference, but the expected difference also changes somewhat due to the changes in segment and gap lengths. This behavior is undesirable for evaluating semantic component indexing, although it might be desirable for content analysis when the number of units can be important.

Number of units	1 unit	2 units	3 units
Unitized	0.8401	–0.8560	–1.5356
Binary	0.6342	0.6342	0.6342

4. Shifting the units to a different position in the document but maintaining their positions relative to each other changes unitized K_α. The mathematical explanation is that when the expected disagreement is calculated, the lengths of all units and gaps contribute to the collection of sections that are compared to each other. When the starting positions for the two units are 6 and 10, gaps of length 6 and 10 are contributed to the expected disagreement calculation. The units, both of length 20, cannot be nested in the gaps of length 6 or 10. When the starting positions are 40 and 44, the gaps on either side of the units are large enough to contain the units. Different values are calculated for D_e, and therefore for K_α, yet there is no reason to believe that the second pair of units is more likely to occur by chance than the first pair of units. This behavior is undesirable for evaluating semantic component indexing, because the two pairs are logically equivalent, and suggests a flaw in the version of K_α for unitized data.

Starting position	Start = 6	Start = 40
Unitized	0.9340	0.9190
Binary	0.7513	0.7513

5. Varying the positions of independent units within the document does not cause a change in binary K_α. We compared the first instance shown under point 5 in Figure 6.9 (the unit starts at position 6) to each of the other four instances. In each instance there is one unit of length 20. The units in different instances

start at different locations, but no instance has a unit that overlaps with the unit in the first instance. All of the pairs of unit positions have the same binary K_α. Two pairs of unit positions have the same unitized K_α but the other two pairs have different values of unitized K_α. The mathematical explanation is the same as for varying the positions of overlapping units. This behavior of the metric is unsatisfactory.

Starting pos.	Start = 6 & 30	Start = 6 & 56	Start = 6 & 76	Start = 6 & 74
Unitized	–0.7851	–0.7851	–0.5698	–0.5954
Binary	–0.2438	–0.2438	–0.2438	–0.2438

6.4. Evaluation Recommendations

In this section, we discuss the findings from our analyses of indexing tasks and candidate metrics and offer recommendations for measuring accuracy and consistency for semantic component indexing and keyword indexing. In Table 6.1, we summarize our conclusions with respect to the most appropriate metrics for assessing accuracy and consistency of both semantic component indexing and keyword indexing. Two principles guided our analyses of consistency metrics: (1) consistency metrics should account for the probability of agreement by chance, and (2) consistency metrics should reflect agreement, not just correlation.

6.4.1. Evaluation of Semantic Component Indexing

Semantic component indexing consists of two tasks, assigning document class and identifying segments of text that belong to semantic component instances. Document

classification is an example of single-label nominal categorization. Semantic component indexing can be conceptualized either as binary classification of each unit of text (such as a character) with respect to each semantic component associated with the assigned document class or as unitizing the text for each semantic component by identifying the segments within semantic component instances.

We have not found any tasks that are identical to semantic component indexing and that can provide an appropriate measure of agreement. We analyzed a variety of tasks related to identifying the text that belongs to a semantic component instance. In this group we include text segmentation, passage retrieval, question answering, novelty detection, and information extraction. All of these tasks involve selecting and manipulating text at a subdocument level. We considered the evaluation methods that have been used for these tasks and conclude that the unit of evaluation is a critical determinant of evaluation approach. Measuring agreement based on the sizes of text fragments (using units such as the number of characters to measure size) that are assigned to the same semantic component by different indexers is suitable for semantic component indexing. Measurements of agreement that are based on counting the number of boundaries placed at the same location or counting the number of answer elements are not applicable to semantic component indexing.

Identifying semantic component instances is similar to content analysis, but there is a critical difference with respect to the importance of the number of segments. For semantic component indexing, we care only about whether units of text are classified the same, either as belonging or as not belonging to a semantic component instance.

In content analysis, the partitioning of text into some number of units that share a label can be important and therefore adjacent segments must be treated as distinct segments even if they have the same label.

Based on our analysis of the semantic component indexing task, we identified the following criteria for a metric that measures agreement between semantic component indexing instances:

- It allows comparing instances of each semantic component separately.
- For segments of a given length, it results in more agreement when the length of the overlap or the length of the nested segment is larger.
- It treats near misses the same as far misses.
- For segments of a given length, it results in the same measured agreement regardless of the position of the nested segment within the containing segment.
- It allows semantic component instances to be discontiguous. It measures overlap between instances and is agnostic regarding whether such overlaps are contiguous and whether the number of segments is the same.

If we treat semantic component indexing as binary classification of each text unit (such as a character) we can use metrics for comparing instances of binary classification to evaluate semantic component indexing. The binary classification approach allows comparison of each semantic component separately. For measuring accuracy, recall and precision applied to characters both satisfy each of our criteria. For measuring consistency, both K_α for binary data and the kappa statistic satisfy our

criteria. K_α has the advantage of being able to handle missing data (and thus different numbers of indexers for each document).

If we treat semantic component indexing as unitizing the text for each semantic component, K_α for unitized data is the only candidate metric for consistency. K_α for unitized data compares sequences of segments, which correspond to the entities of interest for semantic component indexing. Like K_α for binary data, K_α for unitized data results in more agreement when the length of an overlap or the length of a nested segment is increases. However, it rewards nested segments that are centered more than it rewards uncentered nested segments, it results in different values when the number of nested segments changes, and the measured agreement can vary when segments appear in different locations within a document, even if the total size of overlapping and independent segments does not change. These undesirable differences in unitized K_α, which occur in response to segments appearing in different locations within documents, result from the way that unitized K_α uses data about the lengths of segments and gaps from semantic component instances to calculate how much agreement (or disagreement) can be expected to occur by chance. Although K_α for unitized data initially appeared likely to be the best solution for assessing the consistency of semantic component indexing, we conclude that it is not suitable for measuring consistency of semantic component indexing.

We conclude that recall and precision are useful for assessing the accuracy of document classification for semantic component indexing and that K_α for nominal data is useful for assessing the consistency of document classification. We also

conclude that semantic component indexing can be treated as binary classification of each unit of text (such as a character) for evaluating semantic component instances. Recall and precision are useful for comparing semantic component indexing to a reference standard (measuring accuracy) and K_α for binary data is useful for comparing peer instances of semantic component indexing (measuring consistency). The kappa statistic is also suitable for measuring consistency if there is no missing data.

6.4.2. Evaluation of Keyword Indexing

Keyword indexing is an example of multilabel categorization. Depending on the situation, keyword indexing can also be considered as multiple binary classifications. Recall and precision are appropriate measures for assessing accuracy of keyword indexing as compared to a reference standard. For assessing consistency, agreement metrics for keyword indexing exist, although they are not entirely satisfactory. Existing metrics ignore the possibility of agreement by chance, and are designed for comparing only two indexing instances. However, use of existing metrics may be useful for comparing keyword indexing consistency to results published in other studies. When keyword indexing uses a very small controlled vocabulary, or when a variety of indexing instances are available to indicate a universe of possibly appropriate terms that an indexer can reasonably be assumed to have considered, it can also be reasonable to treat the indexing as multiple binary classifications over the limited indexing term possibilities. This approach provides the advantage of being

able to correct for the possibility of agreement by chance. In such situations, K_α for nominal data is an appropriate measure of agreement and allows calculation of consistency for an arbitrary number of indexers.

Table 6.1 Evaluation methods for assessing indexing accuracy and consistency

Quality **Indexing Type**	**Accuracy**	**Consistency**
Semantic components a) Document Classification	Recall and Precision – of document classification, per document class	Nominal K_α – agreement on document class – per document, over all documents
b) Semantic component identification	Recall and Precision – of characters, in each component	Binary K_α – agreement on inclusion/exclusion in the semantic component – by characters, over all characters in each component
Keywords	Recall and Precision – of keywords – per document, and – per vocabulary	Binary K_α – agreement on keyword inclusion – by keywords, over all keywords either suggested by at least one indexer in the study or appearing in the reference standard Traditional consistency formulas: – *consistency* = $c / (a + b - c)$ – *consistency* = $2c / (a + b)$

6.5. Summary

In this chapter we discussed the motivation for choosing metrics to evaluate agreement between instances of semantic component indexing. Accuracy is a measure of agreement between an indexing instance and a reference standard whereas consistency is a measure of agreement among two or more indexing instances in which no instance is preferred over any other instance. We analyzed the task of semantic component indexing and developed a set of criteria for desirable metrics of

agreement for semantic component indexing. We also discussed keyword indexing and developed criteria for metrics of agreement for keyword indexing.

Next, we compared semantic component indexing and keyword indexing to other tasks and discussed existing metrics for related tasks. From our comparisons we identified three groups of candidate metrics, which we further analyzed: (1) metrics for comparing instances of text categorization, (2) metrics for comparing instances of keyword indexing, and (3) a metric for comparing instances of unitization in content analysis. We described three versions of K_α that we implemented for assessing the consistency of nominal, binary, and unitized data. To assess the behavior of two versions of K_α against our criteria for desirable metrics, we created hypothetical semantic component indexing data and calculated binary K_α and unitized K_α for pairs of data.

We concluded that K_α for binary data satisfies our criteria, but K_α for unitized data does not satisfy our criteria. Finally, we summarized our findings and proposed metrics for calculating the accuracy and consistency of semantic component and keyword indexing. In Chapter 7 we apply these metrics to data from our indexing study.

Chapter 7 Semantic Component Indexing: Feasibility and Quality

In this chapter we report on our experiences with semantic component indexing. In Section 7.1, we describe in detail a comparative study of semantic component indexing and keyword indexing. In Section 7.2 we discuss our experience with having 371 documents indexed with semantic components to support the searching study described in Chapter 8. In Section 7.3 we discuss our findings and offer an outlook for semantic component indexing. We summarize in Section 7.4.

7.1. Comparative Study of Semantic Component Indexing and Keyword Indexing

We performed a comparative indexing study described below in collaboration with sundhed.dk. Sundhed.dk uses a combination of full-text and manual keyword indexing. The keyword indexing is performed by a variety of participants in sundhed.dk who index documents as just one part of their jobs. Some indexers are physicians who author documents for sundhed.dk, and others have backgrounds in nursing or information technology. Few have formal backgrounds in library science or have formal training in the principles and practice of keyword indexing. Indexers have the option of using an automated indexing application that suggests keywords, where they can then either accept or reject individual keywords. The automated indexing application is part of Ultraseek [147], the commercial search engine that sundhed.dk uses to power its search portal. Interviews with indexers for sundhed.dk,

during study design and at the end of study sessions, indicated that some indexers use the application heavily, some indexers do not use it at all, and some indexers look at its output but delete most of the suggestions. The disparate backgrounds and highly distributed nature of the indexing might have a substantial effect on indexing quality in the operational system.

The high-level goal of this study was to assess the feasibility of semantic component indexing. To do so, we studied indexers who were experienced with the documents and keyword indexing procedures of sundhed.dk but who were new to semantic component indexing. To give some context to the semantic component indexing data, we compared it to keyword indexing of the same documents by the same group of indexers. The general questions we sought to answer were: How difficult is semantic component indexing? How much time does it take? Will indexers understand the semantic component indexing task and the semantic component schema well enough to able to index documents in a way that accurately reflects the intent of semantic component indexing?

More specifically, we formulated the following questions:

1. Is semantic component indexing of sundhed.dk documents *more accurate* than keyword indexing compared to a reference standard?
2. Is semantic component indexing of sundhed.dk documents *more consistent* than keyword indexing of the same documents?
3. Is semantic component indexing of sundhed.dk documents *faster* than keyword indexing?

4. Is semantic component indexing of sundhed.dk documents *easier* than keyword indexing, as perceived by the indexers?

In the next section we describe the methods we used to investigate these questions. The study received prior approval from the Portland State University Human Subjects Research Review Committee.

7.1.1. Methods

In this section we first describe the experimental setup for the comparative study. We describe the following elements: the semantic component schema and the keyword indexing vocabularies, the indexer participants, the documents to be indexed, the study design, and the materials we used. Next we describe how we analyzed and evaluated the data regarding the accuracy and consistency of indexing.

7.1.1.1. Experimental Design

In Chapter 4, we described our initial analyses of the sundhed.dk document collection (Section 4.1.1) and subsequent refinements to the semantic component schema (Section 4.3). Here we elaborate on the schema development for the indexing study. In the early stage of designing this study we conducted two interviews. The first interview was with Dr. Peter Vedsted, a physician and researcher at the University of Århus in Denmark. Dr. Vedsted was closely involved in Praxis.dk, a regional predecessor to sundhed.dk, and has been instrumental in the development of sundhed.dk. The second was a group interview with four people who currently index

documents for sundhed.dk. In both interviews we first introduced the semantic components model and its potential uses, then asked for participants' feedback. As a result of their feedback, we decreased the number of semantic components associated with each document class from about ten to about five. We then selected three document classes for study, based on frequency in the collection and importance to the target searching audience, which were family practice physicians. The final document classes and corresponding semantic components that we used are shown in Table 7.1.

With the assistance of sundhed.dk, we recruited 16 volunteer indexers to participate in the study. We based recruitment on willingness to participate, indexer status with sundhed.dk, and availability on one of the two dates of the study.

On the day of the indexing study, we first asked the indexers to complete a brief questionnaire that collected information about their experience with medical concepts and terms and about their experience with indexing. Table 7.2 and Figures 7.1 and 7.2

Table 7.1 Document classes and semantic components used in the indexing study

Document Type	**Short Name**	**Semantic Components**
Documents about a Clinical Problem or Condition	Clinical Problem	*Evaluation:* How to diagnose or evaluate the problem *Management:* How to treat, manage or control the problem *Referral:* How to refer a patient with the problem to a specialist or special service *About:* About the problem
Documents about Diagnostic or Therapeutic Procedures	Procedure	*Preparation:* How to prepare for the procedure *Practical:* Practical details *Description:* Description of the procedure *Risks:* Risks of the procedure *Aftercare:* What to expect after the procedure
Documents about rights and services to patients	Services	*Service or right:* Information about the service or right *Inclusion criteria:* The indication or conditions that the patient should fulfill to get the service *Sequence:* the course of events, the sequence of actions

summarize the characteristics of the participants. Our participants had a wide range of backgrounds, representing the variety of people who index for sundhed.dk, but they did not constitute a random sample of the indexer population. Most of the indexers

Table 7.2 Characteristics of indexers

Characteristic	**Range**	**Mean ± Std. Dev.**
Months of experience indexing documents for sundhed.dk	0 – 60	22.2 ± 16.5
Months of experience dealing with medical information	8 – 420	121.6 ± 98.2
Self reported level of experience with indexing; treated as interval scale (1 = Not at all, 5 = Very experienced)	1 – 5	2.7 ± 1.2
Self reported level of experience with medical information; treated as interval scale (1 = Not at all, 5 = Very experienced)	2 – 5	3.8 ± 1.1
Self reported level of knowledge about medical concepts and vocabulary; treated as interval scale (1 = Not at all, 5 = Very knowledgeable)	1 – 5	3.6 ± 1.2

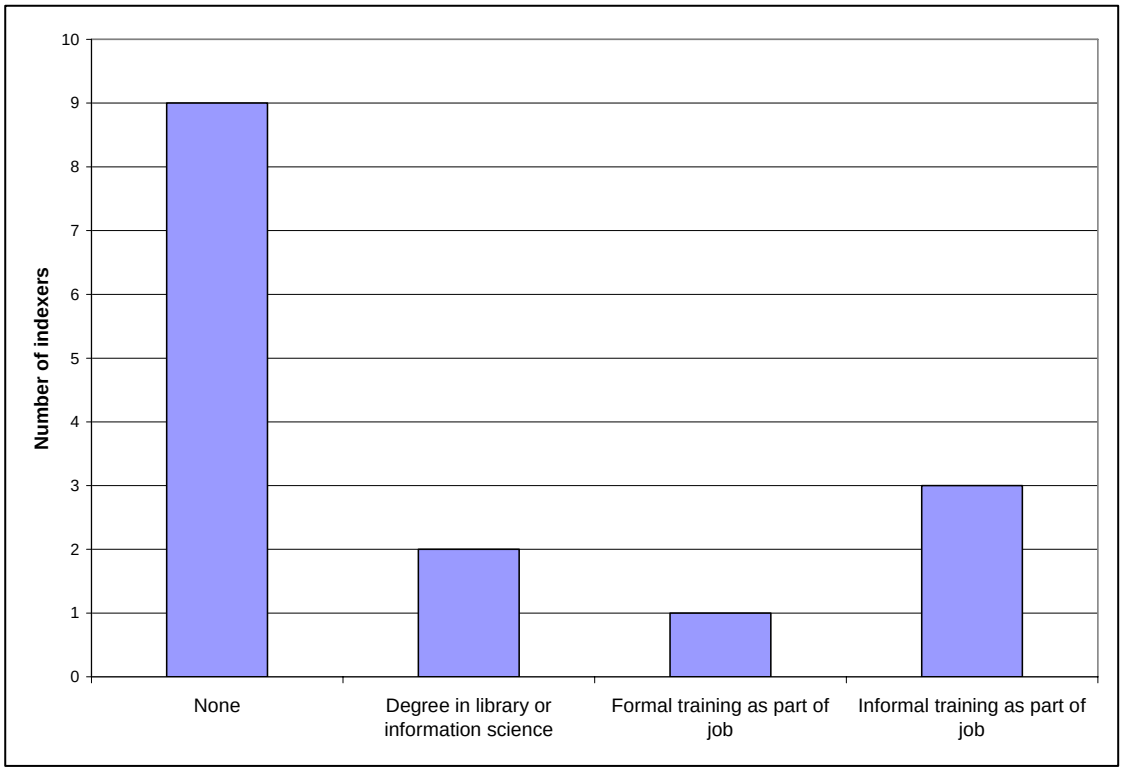


Figure 7.1 Characteristics of indexers: training in indexing

were quite experienced with medical information and were fairly experienced with indexing for sundhed.dk. One indexer was new to indexing. Only a few of the indexers had any formal training in indexing and most of the indexers had received no training at all. Four of the indexers had professional training in medicine or nursing and three of the indexers were medical secretaries. Over half of the indexers had no formal medical training.

We chose twelve documents to be indexed in the study, four documents representing each of the three document classes. We chose documents we believed to be representative of each class and that varied in topic, length and complexity. Some

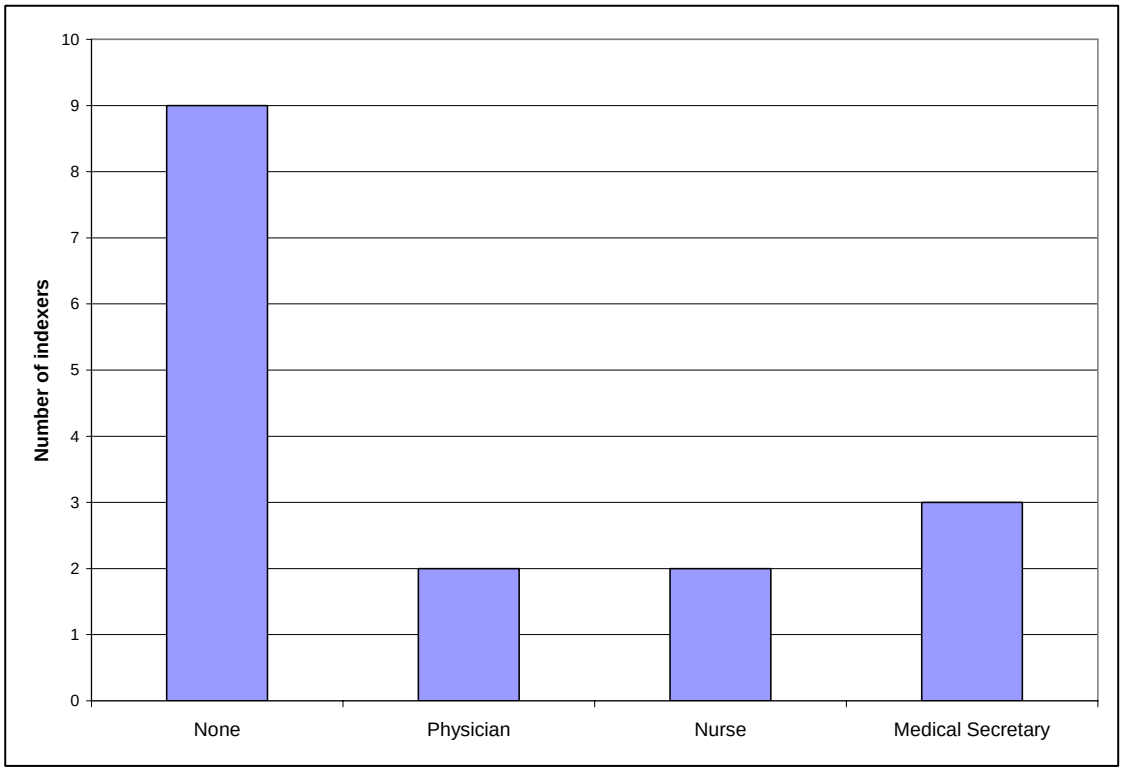


Figure 7.2 Characteristics of indexers: formal medical training

documents appeared to be written for health professionals while others were written for consumers. The documents contain varying amounts of specialized medical vocabulary. Table 7.3 shows the titles (translated from Danish to English) and document class for each of the twelve documents we used in the study. The length shown is the number of paper pages that resulted from printing the web pages.

Table 7.3 Document characteristics

Document Id	Document class	Length (pages)	Title
1	Procedure	1	Hip replacement
2	Clinical Problem	7	Diabetes mellitus
3	Services	1	Psychological help
4	Procedure	3	Radiation therapy
5	Clinical Problem	1	Asthma in children
6	Services	2	Free hospital choice
7	Clinical Problem	4	Dementia
8	Procedure	2	ERCP – Endoscopic Retrograde Cholangio Pancreatography
9	Services	1	Family education (for families of patients with dementia)
10	Clinical Problem	7	Osteoporosis
11	Procedure	1	CT-scanning of the kidney and urinary tract
12	Services	1	Waiting times/free hospital choice

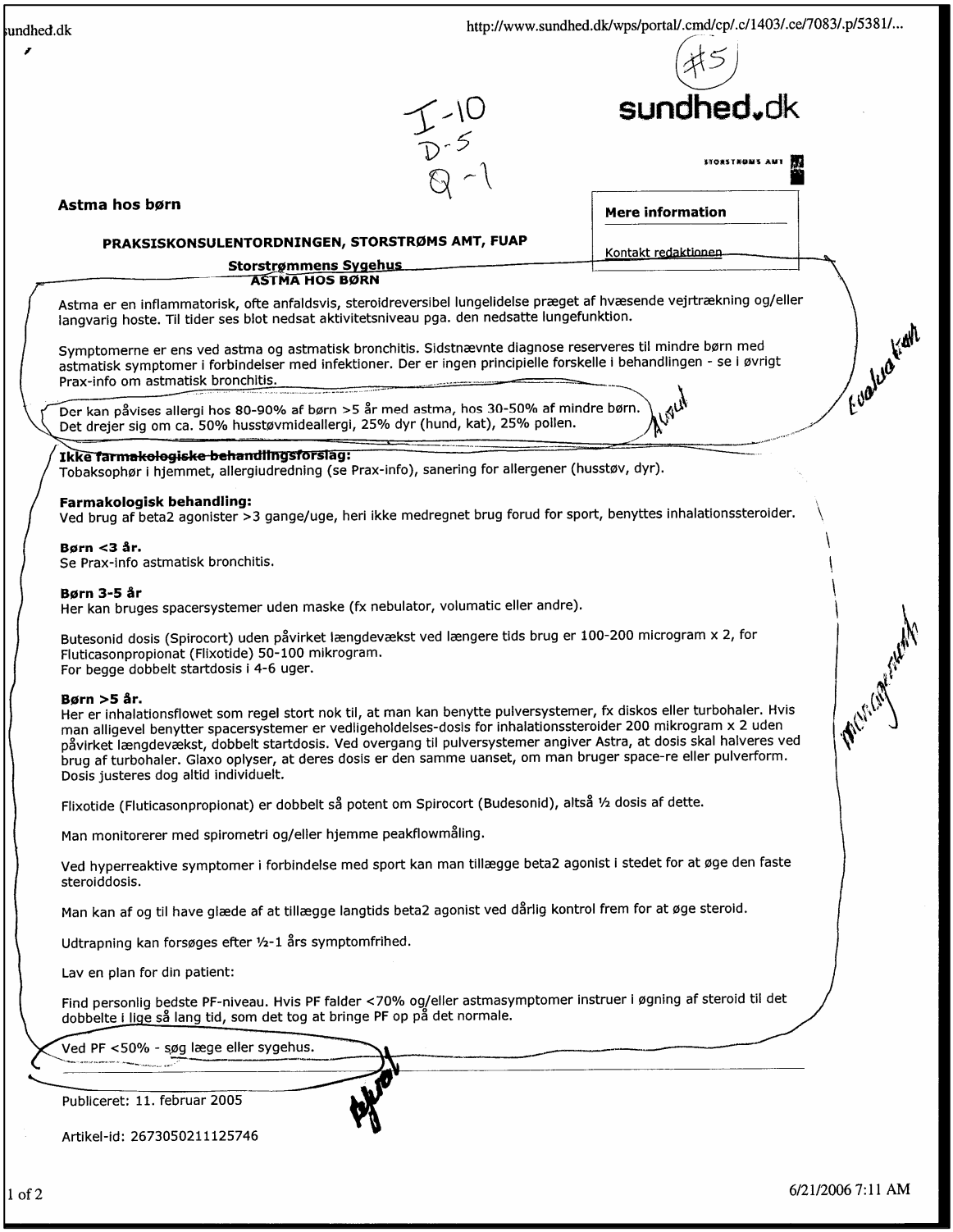
For the keyword indexing portion of the experiment we used the three controlled vocabularies currently in use by sundhed.dk: ICPC, ICD-10, and the Almen Thesaurus. These vocabularies are described in Section 4.1.1. (All the keyword indexing in sundhed.dk and in our study is in Danish). We asked indexers to assign keywords from any or all of the vocabularies as they normally would when indexing documents for the production system. We also allowed indexers to assign "free" keywords that do not appear in any of the vocabularies, the same as they can for the production system.

We collected all indexing information on paper. We did so in order to eliminate any biases that might be introduced by using a software prototype to collect indexing output. User interfaces for the two types of indexing could affect the relative ease or difficulty of making and recording indexing decisions, which could potentially affect time, attitudes, and the quality of indexing decisions. We did not think it would be feasible to create computer interfaces for keyword and semantic component indexing that were equivalent with respect to ease and speed of use. We did provide electronic access to the three controlled vocabularies using the Metadata++ software [148] that allows either browsing (via a collapsible tree structure) or searching of the vocabularies. Each indexer had access to either a desktop computer or a laptop that he could use to view the controlled vocabularies.

We created several paper forms to collect the indexing data. These forms are shown in Appendix A, Figures A.1 – A.3. The first form (A.1) is for recording keyword indexing decisions. The form has slots for the indexer to record indexing concepts as well as for recording the keywords chosen and the source vocabulary for each keyword, if any. (The indexing concepts chosen by the indexers will be the subject of a related study by Dr. Nielsen, one of the collaborators in this project.) There are two forms for semantic component indexing. One form (A.2) contains a description, in English, of each document class. We asked each indexer first to choose one of the three document classes. When he was finished with classifying the document, we gave the indexer a second form that asked him to index the document using a particular document class, even if the class was not the same as the class that

he had chosen. (We did this so that we could compare indexing instances among the indexers. Indexing instances based on different document classes would not be comparable). The semantic components form for each document class, also in English, has a list of the semantic components for the class and a description of each semantic component. The form for Clinical Problem documents appears in the appendix (A.3). We created similar documents for the other two document classes. To index the semantic components, we asked the indexers to draw a line around each segment and label it with the appropriate semantic component label. Figure 7.3 shows a scanned image of a document after it had been indexed by one of the study participants. (The handwritten letters and numbers at the top of the page are codes that were added by the research team at the end of the indexing study to help with document management). We supplied the indexers with colored pens to facilitate distinguishing the different components but we did not require the indexers to use any particular color scheme, or even require them to use the colored pens. We did not want to add to the cognitive task by asking the indexers to associate semantic component labels with particular colors.

We collected information about how long each indexer took to index a document by using a computer program that was integrated into the Metadata++ interface [149] for searching and browsing the indexing vocabularies. The timing interface displayed the titles of the documents in the order they were to be indexed and highlighted the current title. We asked each indexer to click a button when he started indexing a document, and to click a second button when he was finished with that document.



sundhed.dk http://www.sundhed.dk/wps/portal/.cmd/cp/.c/1403/.ce/7083/.p/5381/...

#5

I-10
D-5
Q-1

sundhed.dk

STORSTRØMS AMT

Astma hos børn

PRAKSISKONSULENTORDNINGEN, STORSTRØMS AMT, FUAP

Storstrømmens Sygehus

ASTMA HOS BØRN

Mere information

Kontakt redaktionen

Astma er en inflammatorisk, ofte anfaldsvis, steroidreversibel lungelidelse præget af hvæsende vejrtrækning og/eller langvarig hoste. Til tider ses blot nedsat aktivitetsniveau pga. den nedsatte lungefunktion.

Symptomerne er ens ved astma og astmatisk bronchitis. Sidstnævnte diagnose reserveres til mindre børn med astmatisk symptomer i forbindelser med infektioner. Der er ingen principielle forskelle i behandlingen - se i øvrigt Prax-info om astmatisk bronchitis.

Der kan påvises allergi hos 80-90% af børn >5 år med astma, hos 30-50% af mindre børn. Det drejer sig om ca. 50% husstøvmideallergi, 25% dyr (hund, kat), 25% pollen.

Ikke ~~farmakologiske behandlingsforslag~~:
Tobaksophør i hjemmet, allergiudredning (se Prax-info), sanering for allergener (husstøv, dyr).

Farmakologisk behandling:
Ved brug af beta2 agonister >3 gange/uge, heri ikke medregnet brug forud for sport, benyttes inhalationssteroider.

Børn <3 år.
Se Prax-info astmatisk bronchitis.

Børn 3-5 år
Her kan bruges spacersystemer uden maske (fx nebulator, volumatic eller andre).

Butesonid dosis (Spirocort) uden påvirket længdevækst ved længere tids brug er 100-200 microgram x 2, for Fluticasonpropionat (Flixotide) 50-100 mikrogram.
For begge dobbelt startdosis i 4-6 uger.

Børn >5 år.
Her er inhalationsflowet som regel stort nok til, at man kan benytte pulversystemer, fx diskos eller turbohaler. Hvis man alligevel benytter spacersystemer er vedligeholdelses-dosis for inhalationssteroider 200 mikrogram x 2 uden påvirket længdevækst, dobbelt startdosis. Ved overgang til pulversystemer angiver Astra, at dosis skal halveres ved brug af turbohaler. Glaxo oplyser, at deres dosis er den samme uanset, om man bruger space-re eller pulverform. Dosis justeres dog altid individuelt.

Flixotide (Fluticasonpropionat) er dobbelt så potent om Spirocort (Budesonid), altså ½ dosis af dette.

Man monitorerer med spirometri og/eller hjemme peakflowmåling.

Ved hyperreaktive symptomer i forbindelse med sport kan man tillægge beta2 agonist i stedet for at øge den faste steroiddosis.

Man kan af og til have glæde af at tillægge langtids beta2 agonist ved dårlig kontrol frem for at øge steroid.

Udtrapning kan forsøges efter ½-1 års symptomfrihed.

Lav en plan for din patient:

Find personlig bedste PF-niveau. Hvis PF falder <70% og/eller astmasymptomer instruer i øgning af steroid til det dobbelte i lige så lang tid, som det tog at bringe PF op på det normale.

Ved PF <50% - søg læge eller sygehus.

Publiceret: 11. februar 2005

Artikel-id: 2673050211125746

1 of 2 6/21/2006 7:11 AM

Figure 7.3 Scanned example of an indexing instance

Each indexer participated in a single half-day indexing session. We conducted three indexing sessions, in two different cities on two consecutive days, in order to recruit a variety of participants. The first session was held in Århus, the second two in Odense. Although we had conducted a pilot test before the experiment, we assumed that individual indexers would take variable amounts of time to index the documents. We were not sure that all the indexers would index all the documents in the allotted time, and we wanted to be sure that each indexer used both methods of indexing. We therefore organized the indexing sessions into two blocks, and each block into two sub-blocks. In the first block, each participant indexed three documents using keywords (one sub-block) and three documents using semantic components (another sub-block). In the second block, we gave each participant three additional documents to index using keywords and three additional documents to index using semantic components. Not all participants completed all twelve documents, as is discussed below. We allowed participants to proceed at their own pace and we encouraged them to take a break between the two blocks. At the end of the indexing session, we asked each indexer to complete a final survey that asked questions about the perceived difficulty of each type of indexing and that solicited their opinions about the potential usefulness of semantic component indexing.

We designed a randomization scheme for assigning a sequence of documents and a sequence of indexing techniques to each indexer that would achieve the following effects:

1. We balanced the order of indexing techniques so that, in each indexing session, half the participants started with keyword indexing and half started with semantic component indexing.
2. We systematically varied the order of document presentation.
3. We allocated the sequences of document presentation to indexers in pairs so that, for each document in a sequence, one indexer in the pair indexed the document with keyword indexing and the other indexer in the pair indexed the same document with semantic component indexing.
4. We assigned one document from each class to each sub-block in order to be certain that at least some documents of each of the three document classes would be indexed by all (or nearly all) of the indexers.
5. We assigned every indexer the same six documents (in some order) in the first block to ensure that at least some of the documents would be indexed by as many indexers as possible. We then assigned every indexer the other six documents (in some order) in the second block.

For each of the two blocks in an indexing session, we constructed six sequences of documents, each sequence beginning with a different document in each sub-block. Table 7.3 illustrates the six document sequences. The document id numbers in Table 7.4 correspond to the document id numbers in Table 7.3. All of the *Services* documents were short (1 page or less) and we assigned two *Services* documents to each block (one document in each sub-block). For the *Procedure* and *Clinical Problem* documents, we assigned one fairly short document (1 page and 1-4 pages,

respectively) and one moderately long document (2-3 pages and 7 pages, respectively) of each type to each block. This allocation resulted in six documents for each block (three in each sub-block).

We started with 24 indexer identification numbers, from 1 to 24, inclusive. We then randomly assigned four indexer ids to each of the six document sequences. To balance the order of indexing techniques, we designated two of the four identification numbers that had been assigned to each sequence to start with keyword indexing and two to begin with semantic component indexing. We then had two pairs associated with each sequence; each pair shared the same document sequence but started with a different indexing technique. In other words, we had twelve unique document-indexing technique assignments that were organized in pairs. Because we had recruited more than twelve indexers, we generated enough indexer ids for two instances of every unique sequence (24 ids). We then allocated identification numbers (and thus sequences) to the indexing sessions in pairs, so that two indexers in each session received the same sequence, but one started with keyword indexing and one with semantic component indexing. If a session had an odd number of participants, the unused sequence in a pair was the first to be allocated to the next session. After twelve identification numbers (one instance of each of the twelve unique combinations, which was six paired instances of the six document sequences) had been allocated, we randomly chose two additional pairs to be allocated, for a total of 16 of the possible 24 identification numbers. This organization meant that each document sequence was used at least twice, once beginning with semantic component

indexing and once beginning with keyword indexing. Two sequences were used four times, twice for each order of indexing techniques. Within each session we randomly assigned the indexers to the allocated sequences of documents and indexing techniques.

Table 7.4 Sequences of document presentation

Sequence	**1**	**2**	**3**	**4**	**5**	**6**
Block 1 – Sub-block 1 (keyword or semantic component indexing)						
Document ids	1	2	3	4	5	6
	2	3	4	5	6	1
	3	4	5	6	1	2
Block 1 – Sub-block 2 (semantic component or keyword indexing)						
Document ids	4	5	6	1	2	3
	5	6	1	2	3	4
	6	1	2	3	4	5
Break						
Block 2 – Sub-block 3 (keyword or semantic component indexing)						
Document ids	7	8	9	10	11	12
	8	9	10	11	12	7
	9	10	11	12	7	8
Block 2 – Sub-block 4 (semantic component or keyword indexing)						
Document ids	10	11	12	7	8	9
	11	12	7	8	9	10
	12	7	8	9	10	11

7.1.1.2. Evaluation of Indexing

To evaluate the accuracy and consistency of semantic component indexing, we used the indexing prototype described in Section 3.1 to enter and electronically capture the indexing data from the indexers' original manually marked paper copies of the documents. In the indexing application, we had the option of using either the HTML version of each document or a plain text version derived from the HTML document. Using the HTML version introduces a small amount of artifact because the presence of HTML markup tags affects the character position and segment length of

marked text. Because the indexers marked paper copies (that do not have HTML tags, although the appearance is affected by the HTML markup) we report here the results derived from plain text copies of the documents. However, we entered data using both versions of the documents and found that the results are very similar, generally varying no more than 1 – 2%. Using the HTML version is unlikely to affect conclusions drawn from the indexing instances and is easier for an indexer because the indexing application uses the HTML tags to render the document so that the document appears the same in the application as it would in the web portal.

We entered the keyword indexing data into a spreadsheet, recording the string, as written by the indexer, and the source of the keyword. (Each keyword was either from one of the three controlled vocabularies or was a "free" keyword.) We normalized the keywords chosen from the controlled vocabularies by eliminating differences in case and obvious misspellings. If a keyword did not obviously match a term from the indicated vocabulary, then we used the keyword as it was written by the indexer. For free keywords we converted the words to lower case but we made no other alterations.

In Chapter 6 we discussed the properties of semantic component and keyword indexing and the criteria for suitable evaluation measures of each. We concluded that both kinds of indexing should be evaluated with respect to two qualities: (1) accuracy and (2) consistency. For this study, we evaluate the accuracy of the indexing instances produced by each participant in the indexing study by comparing the instances to a reference standard produced by an indexing expert and a domain expert on the research team. We evaluate consistency by comparing the indexing instances

produced by the participants in the indexing study, treating all indexers as peers. As proposed at the end of Chapter 6, we use the evaluation measures shown in Table 7.5, which is identical to Table 6.1.

Semantic component indexing and keyword indexing are different tasks. The units of evaluation for the two types of indexing (document class assignment and assignment of characters to semantic components versus keyword assignment) are fundamentally different. As a result, the actual values for recall, precision, and K_α are not directly comparable across the two types of indexing. The data presented here allow comparison at a general conceptual level, but do not permit statistical comparison. We present mean values, and standard deviation when appropriate, to summarize the data from various perspectives and to facilitate drawing general conclusions regarding the potential usefulness of semantic component indexing in comparison to an established form of indexing.

7.1.2. Results

First we present the quality data for semantic component indexing, then the quality data for keyword indexing. Next we present the data regarding the time required for indexing. Lastly, we present the data from the questionnaires regarding the indexers' perceptions of the two indexing methods.

Table 7.5 Evaluation methods for assessing indexing accuracy and consistency

Quality Indexing Type	Accuracy	Consistency
Semantic components		
a) Document Classification	Recall and Precision – of document classification, per document class	Nominal K_α – agreement on document class – per document, over all documents
b) Semantic component identification	Recall and Precision – of characters, in each component	Binary K_α – agreement on inclusion/exclusion in the semantic component – by characters, over all characters in each component
Keywords	Recall and Precision – of keywords – per document, and – per vocabulary	Binary K_α – agreement on keyword inclusion – by keywords, over all keywords either suggested by at least one indexer in the study or appearing in the reference standard Traditional consistency formulas: – *consistency* = $c / (a + b - c)$ – *consistency* = $2c / (a + b)$

7.1.2.1. Semantic component indexing quality

We first consider the quality of the document classifications for semantic component indexing and then consider the quality of semantic component identification. Although the indexers were asked to choose the document class that *best* fit the document, in three cases an indexer recorded more than one document class. The three cases involved three different indexers and three different documents. All three documents were about clinical problems that were classified by the indexer as being both C*linical Problem* and *Procedure* documents. In Table 7.6 we show two versions of the results that provide an upper and a lower bound on the accuracy of document classification in this study: (1) we treat the three cases of dual classification as having been classified correctly, ignoring the incorrect class ("best"), and (2) we

treat the three cases of dual classification as having been classified incorrectly, ignoring the correct class ("worst"). The mean recall and precision is the average of the recall and precision, respectively, for the three document classes.[24]

Overall, the accuracy of document classification was fairly good. The primary source of confusion was misclassifying C*linical Problem* documents as *Procedure* documents. Discussion with the indexers revealed some confusion about what kinds of documents, and information, each document class should contain. One possible explanation is related to multiple senses for the word *procedure*. The word procedure can be defined as:

1. "an act or a manner of proceeding in any action or process; conduct."
2. "a particular course or mode of action." [150]

Danish healthcare is heavily subsidized by the government and referral from family practitioners to specialists is carefully managed through a variety of policies and guidelines. The documents in the sundhed.dk health portal often contain information about the procedures (that implement the policies and guidelines) that should be followed by health care professionals and patients, and text about procedures in this sense can appear in many kinds of documents, including all three of the document classes used in this study.

---

[24] We do not report microaveraged values because, as noted in Chapter 6, microaveraged recall and precision for single-label categorization both equal the sum of all true positives (TP) over the total number of categorizations performed.

However, the word *procedure* is also commonly used as jargon in American medical care settings to refer specifically to substantial diagnostic examinations and therapeutic interventions, particularly when such events involve invasive techniques, such as surgical operations, endoscopic examinations, and angiography (examinations of the circulatory system involving injection of dye and radiological imaging). It is the latter sense of the word that we intended in designating the class of documents we called *Procedure*. In our study, there were two ways in which this distinction may have been blurred. First, there is the difference between the American and Danish healthcare systems and word usage that is customary in the two settings. The other is the translation between the English and Danish languages. Although the indexers in our study spoke both English and Danish, English was not their first language and subtle differences in word usage may have caused confusion.

Table 7.7 shows the recall and precision for identifying each semantic component in each of the three document classes. The rows labeled "All" show the results for all of the semantic components in a particular document class combined. The table displays the results using both microaveraging and macroaveraging.

As discussed in Chapter 6, microaveraging averages the results of all decisions over all categories. For semantic component indexing, microaveraging means that we calculated recall and precision by summing all the true positives (TP), all the false positives (FP), and all the false negatives (FN) before calculating recall and precision. Microaveraging gives equal weight to every character, regardless of the size of individual semantic component instances. We do not report a standard deviation for

the microaveraged values because the microaveraged value is actually the result of dividing one sum (the sum over all categories of the TP for each category) by another sum (the sum over all categories of the TP + FN for recall and of the TP + FP for precision) and results in a single value. It is not the mean of several values for which a standard deviation can be calculated. Macroaveraging averages the recall and precision values from each indexing instance over the group of interest, such as all the instances of a particular semantic component, and gives equal weight to each indexing instance.

In general, we see similar trends from both methods of summarizing the data. There are some notable differences however. For example, the precision for the *sequence* semantic component is dramatically different depending upon whether micro- or macro-averaging is applied. This difference occurred because many indexers did not designate any text as belonging to the *sequence* component, which resulted in a precision calculation of 0/0 for some of the individual indexing instances for the *sequence* component. By treating 0/0 as a precision equal to one, the macroaveraged precision is quite high. With microaveraging, because some indexing instances did designate some text as belonging to the *sequence* component, there are no zeros in the calculation. The very small number of TP characters and a moderate number of FP characters result in a low precision with microaveraging.

Recall values are mostly greater than 0.5, with the exception of a few semantic components for which recall is substantially lower. Many of the precision values are even higher, in the range of 0.75 to 0.95, again with a few notable exceptions.

Table 7.6 Accuracy of document classification

Measure	Recall		Precision	
Doc. Class	**Best**	**Worst**	**Best**	**Worst**
Clinical Problem	0.7	0.59	1.0	1.0
Procedure	0.97	0.97	0.74	0.68
Services	0.97	0.97	0.97	0.97
Mean ± SD	0.87 ± 0.16	0.84 ± 0.22	0.90 ± 0.14	0.88 ± 0.17

Table 7.7 Accuracy of semantic component identification by semantic component

Document Class	**Semantic Component**	**Recall** 4 documents microavg.	**Recall±SD** 4 documents macroavg.	**Precision** 4 documents microavg.	**Precision±SD** 4 documents macroavg.
Clinical Problem	Evaluation	0.56	0.60 ± 0.31	0.71	0.66 ± 0.37
	Management	0.53	0.56 ± 0.40	0.86	0.81 ± 0.32
	Referral	0.62	0.62 ± 0.31	0.43	0.59 ± 0.41
	About	0.74	0.76 ± 0.40	0.50	0.75 ± 0.36
	All	0.57	0.63 ± 0.36	0.66	0.70 ± 0.37
Procedure	Aftercare	0.47	0.62 ± 0.36	0.90	0.96 ± 0.15
	Description	0.60	0.66 ± 0.34	0.94	0.93 ± 0.15
	Practical	0.33	0.24 ± 0.30	0.21	0.55 ± 0.48
	Preparation	0.52	0.73 ± 0.41	0.63	0.83 ± 0.33
	Risks	0.45	0.56 ± 0.37	0.93	0.94 ±0.13
	All	0.52	0.58 ± 0.39	0.77	0.84 ± 0.32
Services	Inclusion crit.	0.57	0.51 ± 0.44	0.74	0.84 ± 0.28
	Sequence	0.25	0.25 ± 0.46	0.01	0.70 ± 0.46
	Service	0.77	0.80 ± 0.32	0.83	0.79 ± 0.35
	All	0.69	0.61 ± 0.43	0.72	0.77 ± 0.37

Tables 7.8 and Table 7.9 show the accuracy of semantic component indexing by document and by indexer, respectively. The differences between microaveraging and macroaveraging are less striking, probably because a larger number of components are involved in each calculation, smoothing out the effects of indexing instances and semantic components. Values vary considerably across indexers, and to a somewhat lesser extent across documents.

Tables 7.10 – 7.12 show consistency data for semantic component indexing. Table 7.10 shows the nominal K_α for the agreement among all of the indexers with respect to the document class assigned to the 12 documents in the study. $K_\alpha = 1.0$ would

Table 7.8 Accuracy of semantic component identification by document

Document	**Recall** Microaverage	**Recall ± SD** Macroaverage	**Precision** Microaverage	**Precision ± SD** Macroaverage
1	0.72	0.74 ± 0.37	0.74	0.89 ± 0.26
2	0.54	0.56 ± 0.33	0.61	0.61 ± 0.39
3	0.65	0.59 ± 0.45	0.63	0.72 ± 0.38
4	0.42	0.33 ± 0.29	0.71	0.72 ± 0.41
5	0.85	0.74 ± 0.39	0.83	0.68 ± 0.47
6	0.70	0.59 ± 0.13	0.75	0.81 ± 0.35
7	0.57	0.63 ± 0.39	0.60	0.79 ± 0.31
8	0.68	0.70 ± 0.31	0.88	0.93 ± 0.17
9	0.66	0.66 ± 0.33	0.69	0.76 ± 0.43
10	0.55	0.61 ± 0.35	0.71	0.75 ± 0.26
11	0.67	0.65 ± 0.43	0.83	0.86 ± 0.31
12	0.73	0.63 ± 0.48	0.84	0.83 ± 0.30

Table 7.9 Accuracy of semantic component identification by indexer

Indexer	**Recall** Microaverage	**Recall ± SD** Macroaverage	**Precision** Microaverage	**Precision ± SD** Macroaverage
1	0.66	0.62 ± 0.35	0.71	0.80 ± 0.31
2	0.75	0.60 ± 0.40	0.71	0.89 ± 0.21
3	0.30	0.44 ± 0.43	0.62	0.85 ± 0.28
4	0.45	0.57 ± 0.40	0.70	0.73 ± 0.38
6	0.24	0.43 ± 0.38	0.43	0.71 ± 0.42
7	0.60	0.60 ± 0.38	0.75	0.68 ± 0.45
8	0.55	0.64 ± 0.35	0.80	0.73 ± 0.39
9	0.72	0.69 ± 0.33	0.82	0.80 ± 0.35
10	0.47	0.56 ± 0.40	0.66	0.66 ± 0.46
11	0.70	0.71 ± 0.36	0.76	0.70 ± 0.42
12	0.54	0.55 ± 0.37	0.57	0.75 ± 0.35
14	0.42	0.49 ± 0.45	0.73	0.86 ± 0.28
15	0.73	0.63 ± 0.45	0.88	0.86 ± 0.34
16	0.74	0.50 ± 0.71	0.74	0.75 ± 0.50
17	0.73	0.79 ± 0.33	0.86	0.90 ± 0.24
19	0.81	0.75 ± 0.33	0.61	0.78 ± 0.34

indicate perfect agreement among all the indexers for all documents and $K_\alpha = 0$ would indicate agreement no better than chance. As we did for accuracy, we show two values, a *best* and a *worst* value to set upper and lower bounds on consistency. The two values result from selecting either the "correct" or the "incorrect" class in the

three instances in which an indexer recorded two document classes instead of one class.

Table 7.10 Consistency of document classification

Classification	**Nominal K_α**
Semantic Component Document Classes – Best	0.73
Semantic Component Document Classes – Worst	0.67

Tables 7.11 and 7.12 show consistency data for the assignment of text (characters) to semantic components. In Table 7.11 the data is summarized by semantic component (averaged for the four documents in the document class that contains each component) and in Table 7.12 the data is summarized by document.

The consistency of semantic component indexing is highly variable. For some components the indexers are fairly consistent while for others they are not at all consistent. The consistency is particularly low for the semantic components in *Services* documents. This inconsistency suggests that the indexers did not have a shared understanding of what kind of information belonged in each component. There are at least two possible explanations:

1. Our choice of semantic components for this document class may not have matched the documents' contents well. The inconsistency suggests that the semantic component schema should be reconsidered, to determine whether these documents were unusual representatives of the class or whether a revised schema would be more effective for the documents in this collection.
2. Our descriptions of these semantic components were not sufficiently clear to convey to the indexers the intended information content for each component.

Table 7.11 Consistency of semantic component identification by semantic component

Document Class	**Semantic Component**	**Mean K_α ± SD** (4 documents in each document class)
Clinical Problem	Evaluation	0.42 ± 0.12
	Management	0.35 ± 0.42
	Referral	0.30 ± 0.24
	About	0.48 ± 0.41
	All	0.39 ± 0.30
Procedure	Aftercare	0.65 ± 0.15
	Description	0.39 ± 0.17
	Practical	0.18 ± 0.21
	Preparation	0.61 ± 0.36
	Risks	0.59 ± 0.36
	All	0.48 ± 0.30
Services	Inclusion criteria	0.08 ± 0.23
	Sequence	-0.07 ± 0.04
	Service	0.25 ± 0.15
	All	0.09 ± 0.20

There is also another possible explanation for the markedly low consistency values. The explanation is related to the consistency measurement itself. The ordering of the consistency values for the semantic components in the *Services* documents is the same as the ordering of the recall values. The recall and consistency are highest for the *service* component and lowest for the *sequence* component.

Table 7.12 Consistency of semantic component identification by document

Document	**Mean K_α ± SD** (of all semantic components in the document)
1	0.46 ± 0.35
2	0.21 ± 0.16
3	0.25 ± 0.30
4	0.35 ± 0.23
5	0.50 ± 0.30
6	0.05 ± 0.11
7	0.40 ± 0.48
8	0.66 ± 0.11
9	0.04 ± 0.24
10	0.44 ± 0.16
11	0.48 ± 0.41
12	0.01 ± 0.07

However, the recall values for the *Services* semantic components are in the same range as the recall values for semantic components in other document classes whereas the consistency values for the *Services* semantic components are substantially lower than those for most of the semantic components for the other document classes. These results may be due, at least in part, to the correction by K_α for agreement by chance. K_α (and the other similar agreement metrics discussed in Chapter 6) are affected by the underlying prevalence of the categories (in this case semantic components) [119]. The *Services* documents are all short and tend to be dominated by a single semantic component, either *service* or *inclusion criteria*. The other semantic components in the documents are either very small or absent. The expected agreement is always higher when the distribution among categories is skewed. Therefore, for binary K_α, if a semantic component instance is either very large (nearly all of the characters in a document are in the semantic component instance) or very small (very few of the characters in a document are in the instance), then the expected agreement for that semantic component is quite high. For any given observed agreement, the higher the expected agreement the lower the value of K_α. Therefore it is not too surprising that the agreement for the *sequence* semantic component, which averages only 4.25 characters per document in the reference standard, is near zero, meaning it is not different from agreement by chance. Similarly, the *practical* component of the *Procedure* documents has the poorest consistency and is also relatively small, averaging less than 10% of the document text in the reference standard overall and less than 3% in two of the *Procedure* documents.

However, distribution of text in the semantic components is not the only explanation for the K_α values. We can see from inspecting the raw data that not all the semantic components with low K_α values had skewed distributions and not all semantic components with skewed distributions had low values of K_α. For example, of the two documents with the lowest K_α for the *practical* component, only one has a very small amount of text in the component, as identified in the reference standard. Also, the reference standard for Document 5 includes text in only two of the four semantic components for the *Clinical Problem* class. Despite a skewed distribution, the agreement measured by K_α is 0.5 and none of the individual semantic components has K_α of less than 0.24. Therefore, while a particularly skewed distribution of text for a particular semantic component (that is, either most indexers included a very small, or a very large, proportion of the document in the component) can affect the value of K_α, in general K_α reflects agreement, or lack of agreement, among the indexers.

Because this is the first study of semantic component indexing, we cannot compare our results to any other experimental data. We also cannot say what level of K_α is a good, or acceptable, level of agreement for indexing. Krippendorff [52] has stated that "The choice of reliability standards should always be related to the validity requirements imposed on the research results, specifically to the costs of drawing wrong conclusions." For content analysis, Krippendorff has suggested relying only on variables with $K_\alpha > 0.800$ and to draw only tentative conclusions for variables with K_α between 0.667 and 0.800 [52]. The levels of K_α in our study are mostly below

Krippendorff's suggested threshold for reliability in content analysis. For indexing, we are not drawing conclusions from data, but instead we are establishing a level of agreement between indexers to assess the quality of indexing. We do not know what level of K_α for indexing is required to support enhanced searching. Establishing standards for semantic component indexing consistency will require comparing consistency results with search effectiveness over a range of indexing and searching studies, using a variety of domains and document collections. Our study is a baseline experiment that provides descriptive data that can be compared with data from future studies, which could eventually be used to establish guidelines for indexing quality.

7.1.2.2. Quality of keyword indexing

Tables 7.13 – 7.16 show the accuracy of keyword indexing as calculated by comparing each indexer's choice of keywords to the keywords assigned in the reference standard. Each table shows the recall and precision as determined by both microaveraging (calculating by summing the TP, FP, and FN over all items of interest before calculating recall and precision) and macroaveraging (averaging individual recall and precision values over all items of interest).

Microaveraging gives equal weight to each keyword, regardless of the keyword's source, and is therefore independent of the number of keywords chosen from each vocabulary. The macroaveraged values are obtained by first calculating the recall and precision for each indexer's use of each vocabulary, then averaging these values. This method gives equal weight to each indexing vocabulary-indexing instance

combination. It is not uncommon in our data for the reference standard and the indexing instance being evaluated to contain no keywords from a particular vocabulary. This decision to choose no keywords results in division by zero when calculating recall and precision. As discussed in Chapter 6 (Section 6.3.1.1), occurrences of 0/0 when calculating recall provide no useful information regarding an indexer's recall performance and therefore we did not include such occurrences in the macroaverage calculation for recall. The omission of keywords from a particular vocabulary in both the reference standard and an indexing instance does provide some information about an indexer's precision. An occurrence of 0/0 indicates agreement of the indexer with the reference standard and we therefore treated precision calculations of 0/0 as equal to one for that vocabulary-document combination.

Table 7.13 shows the accuracy by document. The microaveraged recall and precision are quite low. The indexers generally did not choose many of the same keywords as those in the reference standard and also chose keywords that did not appear in the reference standard. As Table 7.13 shows, the different approaches to calculating precision can have a very large effect on the precision value. Instances of 0/0 (no keywords assigned from a vocabulary) were so common that macroaveraged precision was quite high, even though the microaveraged precision was quite low. This high macroaveraged precision means that indexers often concurred with the reference standard by not finding any appropriate keywords from a given vocabulary. When they did choose keywords, their choices often did not appear in the reference standard.

Table 7.13 Accuracy of keyword indexing by document

Document	**Recall** Microaverage	**Recall ± SD** Macroaverage	**Precision** Microaverage	**Precision ± SD** Macroaverage
1	0.13	0.14 ± 0.33	0.21	0.74 ± 0.43
2	0.13	0.35 ± 0.47	0.17	0.74 ± 0.42
3	0.09	0.10 ± 0.23	0.12	0.72 ± 0.42
4	0.10	0.16 ± 0.35	0.14	0.70 ± 0.45
5	0.19	0.38 ± 0.47	0.37	0.85 ± 0.30
6	0.04	0.01 ± 0.04	0.12	0.88 ± 0.31
7	0.17	0.28 ± 0.36	0.24	0.62 ± 0.41
8	0.02	0.01 ± 0.02	0.03	0.61 ± 0.49
9	0.13	0.21 ± 0.39	0.27	0.79 ± 0.39
10	0.11	0.25 ± 0.42	0.33	0.79 ± 0.39
11	0.11	0.12 ± 0.27	0.22	0.80 ± 0.36
12	0.09	0.03 ± 0.08	0.19	0.85 ± 0.34

Table 7.14 Accuracy of keyword indexing by document class

	Recall Microaverage	**Recall ± SD** Macroaverage	**Precision** Microaverage	**Precision ± SD** Macroaverage
Clinical Problem	0.15	0.32 ± 0.42	0.25	0.75 ± 0.39
Procedure	0.09	0.12 ± 0.29	0.15	0.71 ± 0.44
Services	0.09	0.08 ± 0.23	0.16	0.81 ± 0.37

The effect of the sparseness of the keyword data on the macroaveraged precision values is also reflected in the mean values by document class shown in Table 7.14. The macroaveraged recall for the *Clinical Problem* documents was noticeably higher than the microaveraged value because of the occurrence of indexing instances with assignment of a keyword that coincided with the only keyword in the reference standard from that vocabulary. Weighting these instances with a perfect recall for a single keyword as highly as other instances, and eliminating instances where no keywords were assigned in the reference standard, resulted in a recall more than twice as high as the microaverage recall. Fewer keywords were assigned, and fewer

keywords coincided with the reference standard, for the other two document classes. The use of few keywords is probably because the controlled vocabularies are not well-suited to the *Procedure* and *Services* documents. Neither ICPC nor ICD-10 were designed for document indexing. The Almen thesaurus is intended for document indexing, but the keywords in the vocabulary, like those in ICPC and ICD-10, mostly represent concepts related to health and disease. Terms that represent specific medical treatments or examinations, or that represent concepts related to services provided by the healthcare system, are uncommon in all three vocabularies. Note that these three vocabularies are currently in use in the operational sundhed.dk portal and are familiar to the participants in the indexing study. Although these results suggest that the vocabularies are not ideal for some of the documents in the portal, we are not aware of any more suitable vocabularies that are available in Danish.

Tables 7.15 and 7.16 show the accuracy of keyword indexing by vocabulary and by indexer, respectively. Accuracy and precision is the highest for ICPC. ICPC is also the smallest of the vocabularies, which may facilitate finding appropriate keywords and may result in fewer choices between keywords with similar meanings. The Almen thesaurus had the next highest recall and precision, and is also the next smallest vocabulary. ICD-10 is a very large vocabulary that is designed for coding diagnoses and contains multiple codes related to certain diseases. It is not surprising that accuracy was low for ICD-10. The lowest accuracy was for unrestricted keywords, which also is not surprising given an infinite universe of terms and the lack of any normalization for word variations (such as alternate spellings, different verb

tenses, or use of synonyms). The accuracy varied somewhat by indexer. The indexers' precision was consistently better than their recall. The relatively high macroaveraged precision values reflect the frequent occurrence of agreement with the reference standard by not choosing any appropriate keywords from a given vocabulary.

Table 7.15 Accuracy of keyword indexing by vocabulary

Vocabulary	**Recall**	**Recall ± SD**	**Precision**	**Precision ± SD**
	Microaverage	Macroaverage	Microaverage	Macroaverage
Almen Thes.	0.21	0.26 ± 0.37	0.28	0.67 ± 0.42
ICPC	0.31	0.31 ± 0.44	0.46	0.82 ± 0.36
ICD-10	0.10	0.11 ± 0.30	0.23	0.82 ± 0.38
Free keywords	0.04	0.04 ± 0.09	0.08	0.72 ± 0.42

Table 7.16 Accuracy of keyword indexing by indexer

Indexer	**Recall**	**Recall ± SD**	**Precision**	**Precision ± SD**
	Microaverage	Macroaverage	Microaverage	Macroaverage
1	0.14	0.15 ± 0.30	0.32	0.81 ± 0.34
2	0.10	0.13 ± 0.30	0.24	0.80 ± 0.38
3	0.11	0.17 ± 0.38	0.45	0.92 ± 0.26
4	0.12	0.13 ± 0.29	0.19	0.75 ± 0.40
6	0.15	0.21 ± 0.38	0.35	0.78 ± 0.39
7	0.10	0.26 ± 0.41	0.11	0.71 ± 0.42
8	0.08	0.18 ± 0.39	0.15	0.79 ± 0.41
9	0.23	0.32 ± 0.43	0.44	0.73 ± 0.41
10	0.12	0.14 ± 0.29	0.2	0.60 ± 0.47
11	0.06	0.09 ± 0.29	0.14	0.84 ± 0.37
12	0.14	0.17 ± 0.35	0.4	0.80 ± 0.37
14	0.16	0.11 ± 0.24	0.20	0.69 ± 0.42
15	0.18	0.27 ± 0.44	0.64	0.90 ± 0.29
16	0.05	0.06 ± 0.10	0.05	0.21 ± 0.38
17	0.10	0.15 ± 0.29	0.11	0.65 ± 0.44
19	0.15	0.20 ± 0.38	0.29	0.78 ± 0.40

The results for consistency of keyword indexing appear in Tables 7.17 – 7.19. These three tables show the consistency by document, by document class, and by vocabulary, respectively.

Table 7.17 shows the K_α and the values for the two traditional consistency formulas for keyword usage by document. For calculating consistency by document we treated each keyword-source pair as a single keyword, thus combining the keywords from all vocabularies. In other words, the same string, such as "Asthma," from two different vocabularies (or as a free keyword and from a controlled vocabulary) are considered distinct keywords. K_α produces a single value for the consistency between an arbitrary number of indexers, while the traditional formulas only calculate consistency for a pair of indexers. We therefore report a mean consistency value that is the average across all pairs of indexers for the traditional consistency formulas. With a few exceptions, the consistency values are quite low. Most K_α values suggest that agreement is no better than would be expected by chance. Generally, documents with the lowest values of K_α also have low values calculated by the two traditional consistency formulas. An exception is Document 6. Inspection of the indexing data shows that six pairs of indexers had traditional consistency values of 1.0 because four indexers did not assign any keywords at all to Document 6. Document 12 had one such pair. If we omit pairs of indexers who assigned no keywords, Document 6 has consistency values of 0.02 and 0.03 for traditional formulas 1 and 2, respectively. Document 12 has consistency values of 0.04 and 0.05 for traditional formulas 1 and 2, respectively.

Table 7.17 Consistency of keyword indexing by document

Document	**Binary K_α** (all vocabularies)	**Traditional 1 ± SD** *consistency* = $c / (a + b - c)$	**Traditional 2 ± SD** *consistency* = $2c / (a + b)$
1	–0.08	0.05 ± 0.13	0.07 ± 0.17
2	0.001	0.18 ± 0.19	0.27 ± 0.24
3	–0.08	0.05 ± 0.11	0.08 ± 0.16
4	0.02	0.19 ± 0.30	0.24 ± 0.31
5	0.32	0.33 ± 0.23	0.45 ± 0.25
6	–0.07	0.23 ± 0.41	0.24 ± 0.41
7	0.26	0.27 ± 0.18	0.4 ± 0.18
8	–0.08	0.05 ± 0.11	0.08 ± 0.17
9	–0.02	0.09 ± 0.14	0.13 ± 0.21
10	0.27	0.29 ± 0.13	0.43 ± 0.16
11	–0.06	0.04 ± 0.09	0.06 ± 0.14
12	–0.12	0.08 ± 0.24	0.10 ± 0.26

Table 7.18 Consistency of keyword indexing by document class

Document Class	**Mean K_α ± SD** (four documents)	**Traditional 1 ± SD** *consist.* = $c / (a + b - c)$	**Traditional 2 ± SD** *consist.* = $2c / (a + b)$
Clinical Problem	0.21 ± 0.14	0.27 ± 0.20	0.39 ± 0.22
Procedure	–0.05 ± 0.05	0.08 ± 0.20	0.12 ± 0.23
Services	–0.07 ± 0.04	0.12 ± 0.12	0.14 ± 0.14

Table 7.18 shows the same data as Table 7.17 after averaging the K_α values for the four documents in each document class and after averaging all the pairwise consistency values across all documents in a class (instead of averaging the values for the pairs for a single document). Although the raw numbers are different for each method of measuring consistency, the ordering across document class is the same. Clearly, consistency was higher for the *Clinical Problem* documents than for the *Procedure* and *Services* documents, although consistency is generally low. Only the *Services* document class is affected by agreement due to choosing no terms. If pairs of indexers who chose no terms for a document are eliminated from consideration, the consistency for the *Services* document class drops to 0.05 and 0.07 for traditional formulas 1 and 2, respectively.

The consistency values by document class reported in Table 7.18 are based on combining terms from all vocabularies. If, instead, we look at the usage of each of the three controlled vocabularies (eliminating free keywords from consideration) we find substantial differences in the number of indexers who chose no terms from a vocabulary. There were 15 instances of an indexer choosing no terms from a vocabulary while indexing a document for *Clinical Problem* documents, 28 instances for *Procedure* documents, and 68 instances for *Services* documents. This data suggests that while the controlled vocabularies might have been mostly adequate for *Clinical Problem* documents, they were less useful for *Procedure* documents and much less useful for the *Services* documents.

For consistency by vocabulary, we calculated the K_α consistency values for use of each vocabulary in each document separately, then averaged the values from all documents for each vocabulary. We also calculated values for the traditional consistency formulas for each pair of indexers for each document-vocabulary pair separately, then averaged the values across all indexer pairs and all documents for each vocabulary. When neither indexer in a pair of indexers being compared assigned a keyword from a vocabulary (or from any vocabulary when calculating consistency by document), both consistency formulas yield 0/0. In these cases we followed the same reasoning we used in calculating precision. We treated the consistency as 1.0 because not choosing a keyword reflects agreement between the indexers.

Table 7.19 shows the consistency data by vocabulary. By all measures, consistency is best when the indexers used ICPC. Consistency is next best using

keywords from the Almen thesaurus when measured by K_α, but is better with ICD-10 when measured with either of the two traditional consistency formulas. However, the ICD-10 results are heavily influenced by the number of pairs who chose no keywords from ICD-10. Although for each of the vocabularies there are some pairs of indexers who chose no keywords from the vocabulary, the Almen thesaurus has the fewest such pairs (five) while ICD-10 had the most (79 pairs). ICPC and free keywords had intermediate numbers of pairs (53 and 39, respectively). If we eliminate such pairs from consideration, then the consistency for the Almen thesaurus is much closer to that for ICPC. For Traditional Formula 1, the consistency becomes 0.17 and 0.20 for the Almen thesaurus and ICPC, respectively. For Formula 2, consistency becomes 0.38 and 0.35 for the Almen thesaurus and ICPC, respectively. Eliminating pairs that chose no keywords from ICD-10 and chose no free keywords reveals the substantial failure to agree on keywords from those sources. Consistency becomes 0.07 and 0.02 for ICD-10 and free keywords, respectively, for Formula 1, and 0.07 and 0.03, respectively, for Formula 2.

Table 7.19 Consistency of keyword indexing by vocabulary

Vocabulary	**Mean $K_\alpha \pm$ SD**	**Traditional 1** *consistency* $= c / (a + b - c)$	**Traditional 2** *consistency* $= 2c / (a + b)$
Almen Thes.	0.03 ± 0.21	0.18 ± 0.34	0.21 ± 0.36
ICPC	0.09 ± 0.25	0.35 ± 0.45	0.37 ± 0.46
ICD-10	-0.05 ± 0.06	0.33 ± 0.47	0.33 ± 0.47
Free	-0.07 ± 0.05	0.16 ± 0.35	0.16 ± 0.35
All	0.03 ± 0.16	0.16 ± 0.24	0.21 ± 0.28

K_α calculations are based on binary decisions for each keyword (the keyword is either chosen, or not chosen). K_α gracefully handles comparisons over an arbitrary

number of indexers and also comparisons in which keywords are used frequently or infrequently. The traditional formulas suffer from the limitation of comparing one pair of indexers at a time and also from yielding undefined values when no keywords are chosen by either indexer in a pair. However, using two different methods for handling 0/0 values in the two traditional formulas, either omitting them or treating them as representing perfect agreement, helps to highlight the amount of agreement due to not choosing keywords versus the amount of agreement due to keyword choices. Overall, the three approaches to calculating consistency generally provide similar information regarding the relative consistency across vocabularies or across documents (or groups of documents). Using multiple methods for assessing consistency provides greater confidence in the conclusions drawn from our evaluations.

We are not aware of any previous studies on keyword indexing that have used K_α for assessing indexing consistency. Certainly the values in this study are substantially below the threshold that Krippendorff offers for reliability in content analysis studies. And while not directly comparable, the values for keyword indexing are strikingly lower than those we obtained for semantic component indexing.

For comparison, we briefly describe two keyword indexing studies that used Traditional Formula 1 to analyze indexing consistency.

- Funk and Reid [28] analyzed episodes of unintentional duplicate indexing for 760 journal articles in the National Library of Medicine MEDLINE database that were indexed with MeSH. Funk and Reid analyzed the indexing terms in

nine categories, including *Checktags*, which are a limited number of frequently-used descriptors that indexers are expected to consider for every MEDLINE article (such as HUMAN, MALE, PREGNANCY, INFANT), *Main headings*, which comprise the bulk of MeSH terms, *Central concept Main headings*, which are MeSH terms that appear with an asterisk before the term to indicate that the term reflects a central concept of the article, and *Main Heading/subheading* combinations, which are MeSH terms with an attached subheading. Indexers had the highest consistency for *Checktags*, 0.75, and the lowest consistency for *Main heading/subheading* combinations, 0.34. Funk and Reid compared their results to some older, smaller studies of MEDLINE articles that reported indexing consistency from 0.34 for *Checktags*, *Main headings*, and *subheadings* to 0.48 for *Checktags* and *Main headings* only. They also noted a consistency of 0.55 for *Central concepts* only in a study of a computer-assisted indexing method.

- As part of a text categorization study, Uren [151] studied the consistency of four experienced indexers using a thesaurus related to welding technology to index bibliographic records (title and abstract only) for nine documents related to welding. She reported the mean consistency for each of the six possible pairs of indexers and an overall mean. The pairwise consistency means ranged from 0.37 to 0.44 with an overall mean of 0.41.

The indexing consistency in both studies is higher than we observed in our study. There are at least two possible explanations for the lower consistency in our study.

First, both of these studies used a single keyword indexing vocabulary that had been designed specifically for indexing the type of documents in the study. It seems reasonable to assume that appropriate keywords were available for each document and that the indexer had only to determine which keywords were most appropriate. Neither study mentions the issue of zero keywords being assigned and it is reasonable to assume that all documents had at least one keyword assigned by each indexer. If the indexing vocabulary is less well-suited to the documents, as we believe to be the case for at least some of our documents, the indexer's task is more difficult and more likely to result in inconsistency. Second, the study by Funk and Reid and the study by Uren used professional indexers. Although the indexers in our study were experienced in the domain and had experience indexing documents for sundhed.dk, indexing was not their primary job. Indexing was an intermittent task they performed in the course of their other duties, usually without any formal training.

7.1.2.3. Time required for indexing

Using a paper interface for the indexing and a computer interface for the timing was unsatisfactory. Indexers sometimes forgot to click on the interface and the times recorded are not always reliable indicators of the time actually spent indexing documents. Most of the errors can be identified because the indexer clicked the start and stop buttons in rapid succession. There may be additional errors if an indexer started the next document before remembering to click the appropriate buttons, or if the indexer took a break without clicking on the button to indicate completion. We

eliminated all instances for which the elapsed time was less than 15 seconds, assuming that these instances represent errors. All other data is included in the summary shown in Table 7.20. While this data probably contains some errors, we believe that the mean times at least provide a rough estimate of how much time the indexers spent using each indexing method.

If all indexers had completed indexing all their assigned documents, there would be 96 instances of each type of indexing (6 instances for each of 16 indexers). However, not all indexers completed all documents. We have a total of 83 instances of semantic component indexing and 88 instances of keyword indexing, for an average of 5.2 and 5.5 documents completed per indexer. The average indexing times shown in Table 7.20 are based on 78 semantic component indexing instances and 77 keyword indexing instances (after eliminating times less than 15 seconds). Table 7.20 also shows the maximum and minimum indexing times. Although 24 seconds is a fast indexing time and could also represent a timing error, the document indexed in 24 seconds was very short, less than half a page. The next fastest times for semantic component indexing were 56 and 86 seconds, suggesting that at least some documents can be indexed quite quickly. Differences in indexing time are probably related to several factors, including document length and how difficult the document was to comprehend. Some documents are written in nontechnical language and contain no concepts that are likely to be difficult to understand for an average reader. Others contain domain-specific concepts and medical terminology that may be difficult for someone without specialized training. The effect of document length is supported by

a correlation coefficient of 0.57 between document length (in characters) and indexing time.

Figure 7.4 shows the distribution of indexing times for both types of indexing. The data in Figure 7.4 and the mean times in Table 7.20 suggest that keyword indexing was slightly faster than semantic component indexing. The difference is relatively small, however. Figure 7.5 shows the mean indexing time by document class for each type of indexing, demonstrating that the similarity applies to all three classes of documents. Most of the increased time for semantic component indexing is attributable to the *Clinical Problem* document class. These documents were generally longer than documents of the other two classes. One possible explanation for the larger difference in indexing time for *Clinical Problem* documents is that the longer documents may have had more segments per semantic component. If so, this would require more handwritten labels whereas the number of keywords would not necessarily increase for longer documents.

Table 7.20 Time required for indexing documents

Indexing Type	Total Documents Indexed (max = 96)	Mean Num. Docs Indexed Per Indexer (max = 6)	Mean Time (min:sec)	Min Time (min:sec)	Max Time (min:sec)
Semantic Components	83	5.2	07:03	00:24	27:05
Keywords	88	5.5	05:56	01:06	31:26

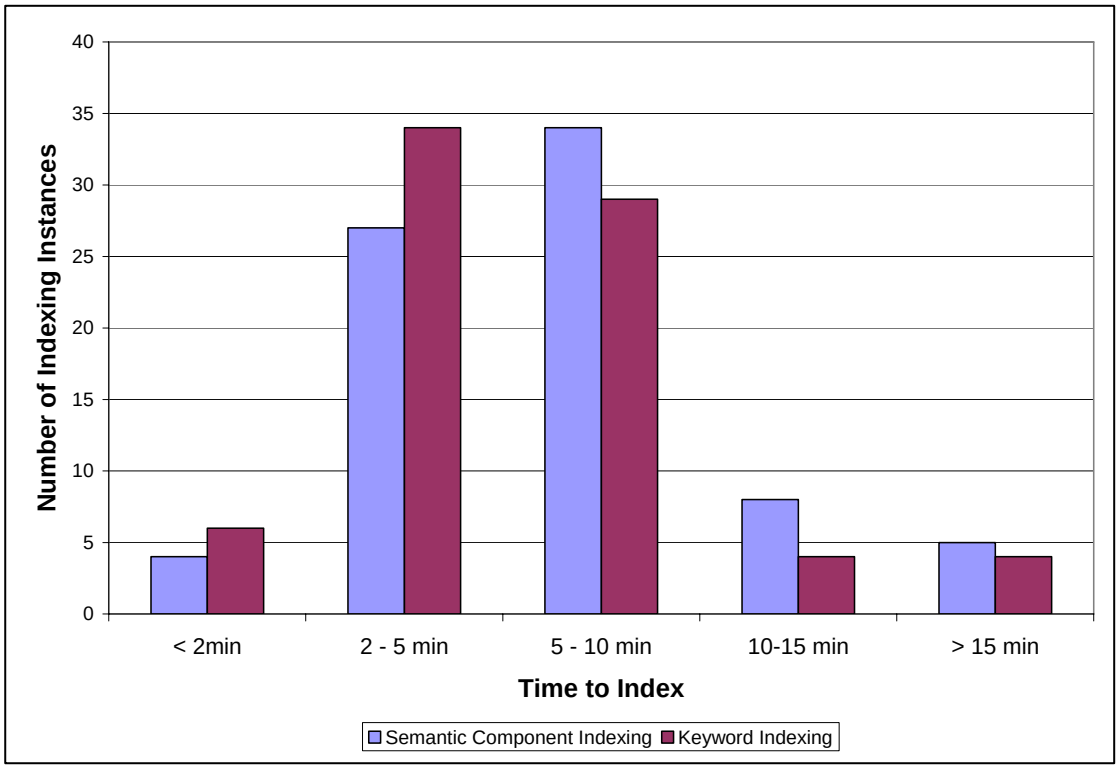


Figure 7.4 Distribution of indexing times

7.1.2.4. Indexers perceptions of the indexing tasks

In this section we summarize the findings from the final survey by grouping questions relating to:

- an indexer's perception of the difficulty of indexing
- an indexer's confidence in the indexing just performed
- an indexer's preference regarding the two types of indexing.

Figure 7.6 displays the data from six survey questions related to indexing difficulty. The bars show the number of responses in each category for each question. Each question allowed five possible responses, with the extremes labeled as "Very Difficult (left-most bars, in red) and "Very Easy" (right-most bars, in green). The

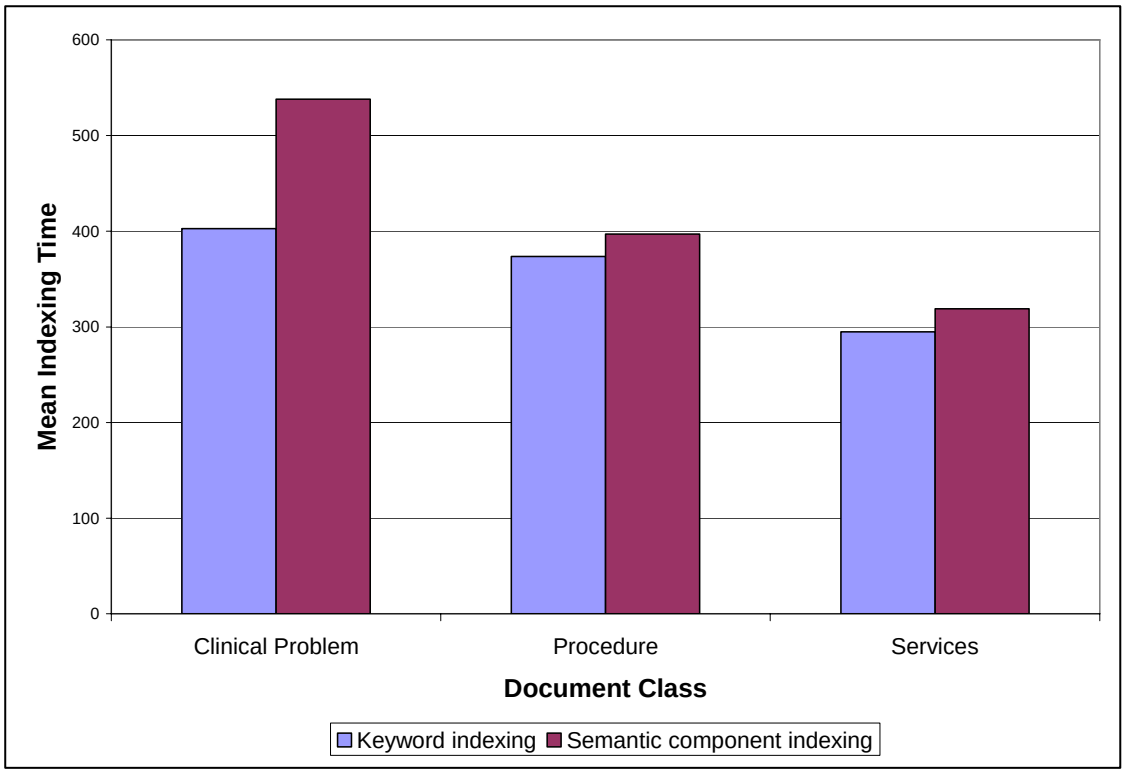


Figure 7.5 Mean indexing times by document class

intermediate bars (pink, yellow, and light green) represent the choices between the extremes. One indexer circled more than one score for some questions and wrote the comment "depend on the document and the information." We distributed the responses for that indexer evenly (half of one point each) between the two scores that were circled on the survey.

Two questions addressed aspects of keyword indexing: (1) how easy (or difficult) it was to choose which concepts to index, and (2) how easy (or difficult) it was to choose keywords to represent the concepts. Responses ranged from *Very Difficult* to *Very Easy*, with slightly more indexers finding the tasks easy (the two green bars) rather than difficult (the pink and red bars).

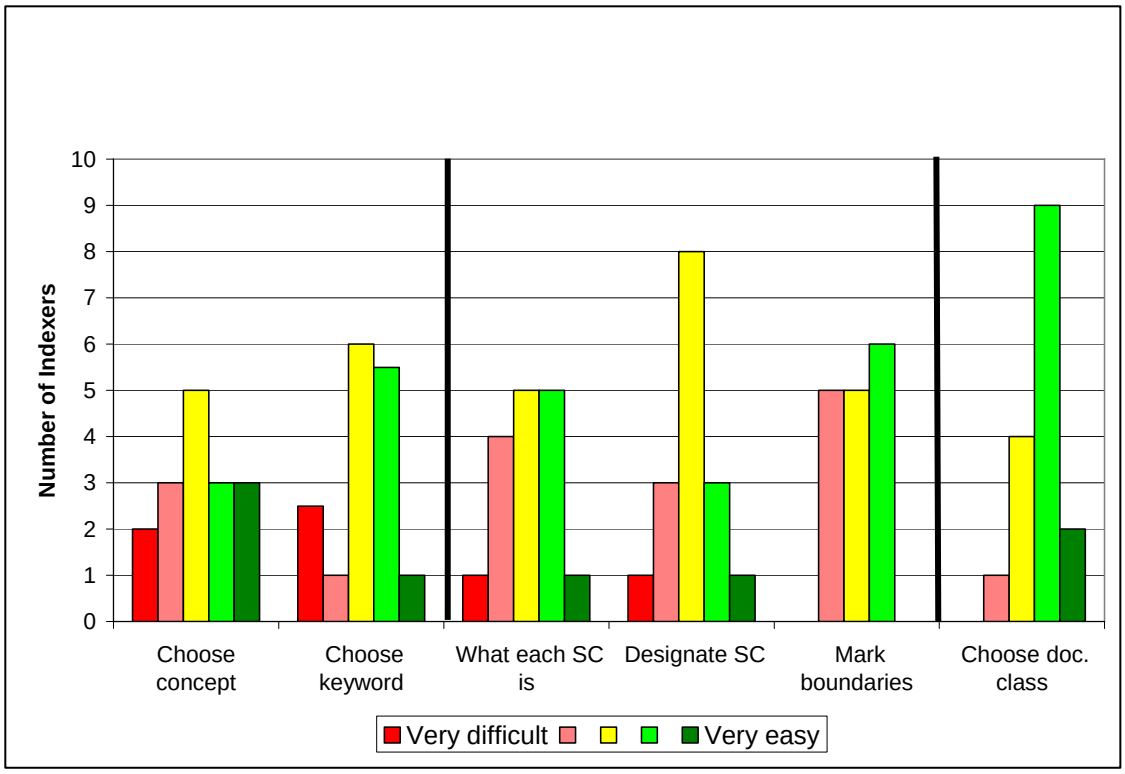


Figure 7.6 Indexing difficulty

Three questions addressed aspects of semantic component indexing: (1) how easy (or difficult) it was to understand what each semantic component was (what kind of information it should contain), (2) how easy (or difficult) it was to designate semantic components for the documents, and (3) how easy (or difficult) it was to decide where the boundaries of the semantic component text should be. Answers again ranged from *Very Difficult* to *Very Easy*, except for the question about boundaries for which all responses were in the middle three categories. As many or more indexers found these tasks to be easy as indexers who found the tasks to be difficult. Compared to keyword indexing, about the same number of indexers found the semantic component tasks at

least somewhat easy, except for designating semantic components. The scores for designating the semantic components were evenly distributed between easy and difficult. Slightly fewer rated the task easy (the two bars furthest to the right) compared to those who rated keyword indexing to be easy.

The final question addressed the difficulty of choosing the document class. None of the indexers found this to be *Very Difficult* and most found the task to be easy.

Figure 7.7 displays data from five questions related to the indexers' confidence regarding the indexing they had just completed. Again, the bars indicate the number of indexers who chose each response. The five possible responses ranged from "Not At All Confident" (left-most bars, in red) to "Very Confident" (right-most bars, in green) with intermediate bars (pink, yellow, and light green) representing the choices between the extremes.

The first two questions addressed the indexer's confidence regarding (1) the keywords chosen from the three controlled vocabularies, and (2) the free keywords assigned. The next two questions addressed the indexer's confidence regarding (1) the semantic component labels, and (2) the boundaries chosen for semantic component instances. The final question addressed the indexer's confidence regarding the document class. For all questions (both types of indexing), the responses tended toward the middle range (neutral), and more indexers were confident (the two response categories represented by the bars to the right of middle) than not confident (the two response categories represented by the bars to the left of middle). More indexers were confident about document classification than any of the other tasks. For

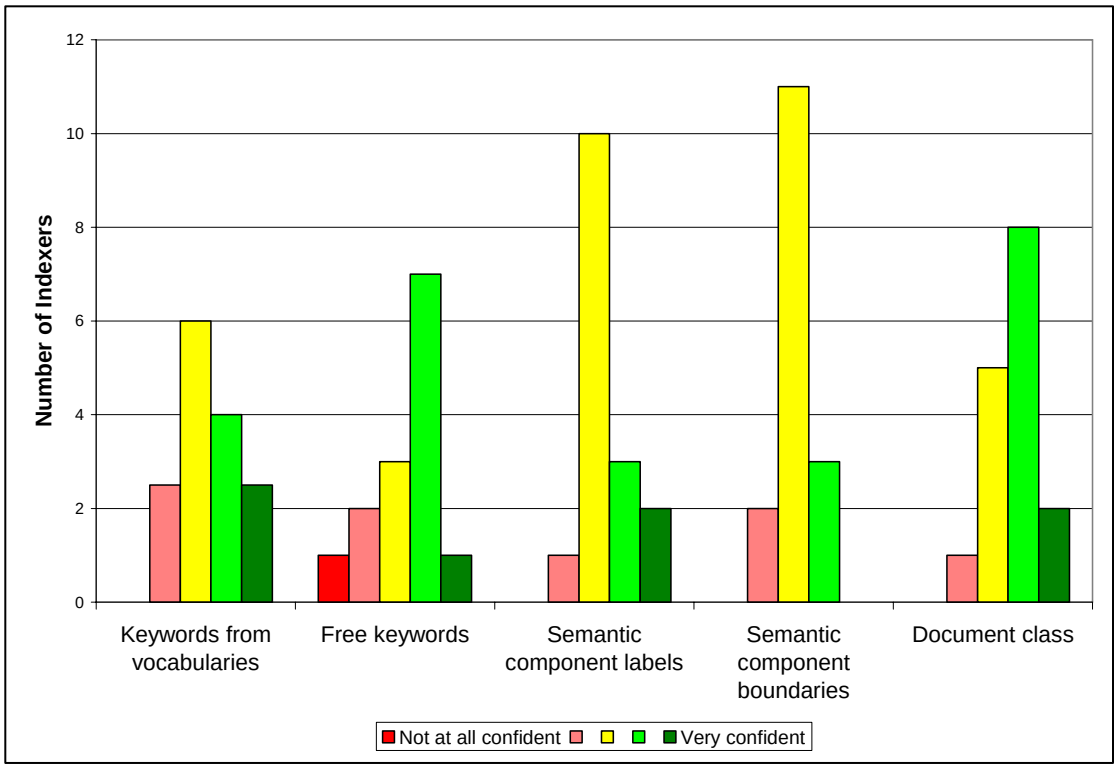


Figure 7.7 Indexer confidence

keyword indexing, more indexers were confident about their free keyword choices than were confident about their choices from the controlled vocabularies. For semantic component indexing, more indexers were confident about the semantic component labels than were confident about the boundaries. Although somewhat more indexers were confident about their keyword indexing than were confident about their semantic component indexing, more indexers expressed a lack of confidence in their keyword indexing than expressed a lack of confidence in their semantic component indexing.

The final group of questions is about indexer preferences. Figure 7.8 shows that most indexers had an equal preference for the two types of indexing. However,

slightly more indexers preferred keyword indexing as a task to perform whereas slightly more indexers thought semantic component indexing would be better for searching.

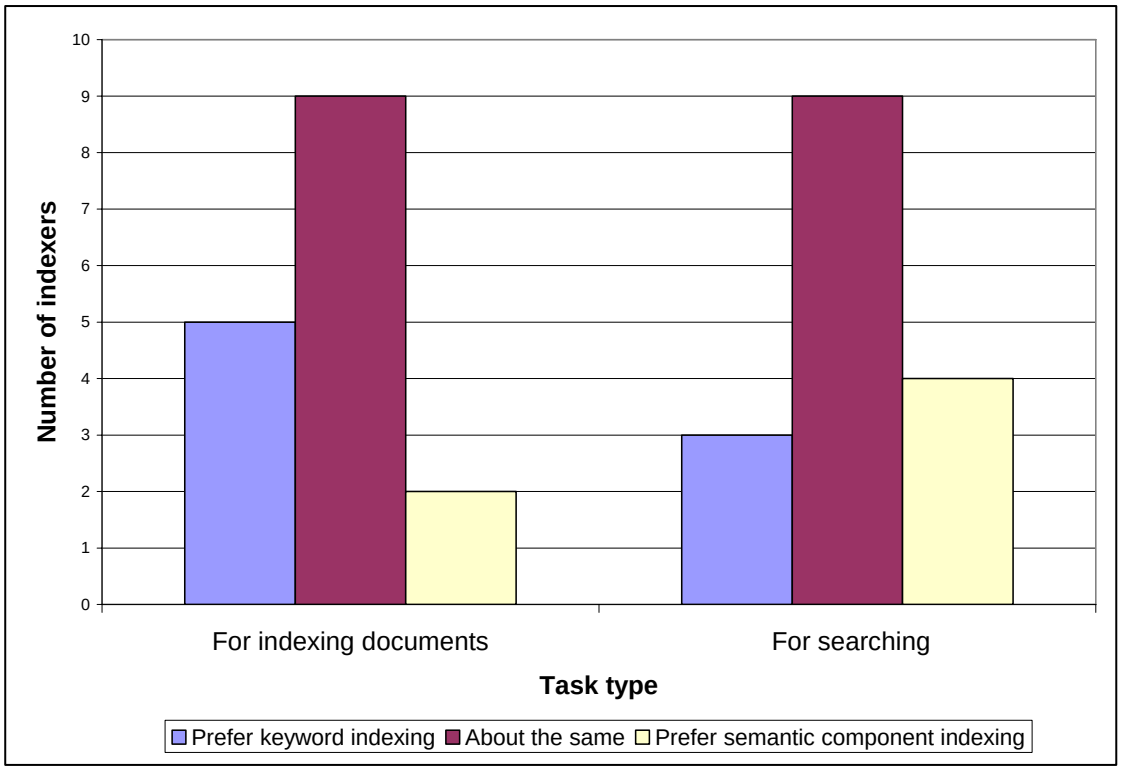


Figure 7.8 Indexer preferences regarding indexing system for performing indexing and searching tasks

Overall, it appears that the semantic component indexing tasks were comparable to keyword indexing. Document classification was somewhat easier than keyword indexing and indexing the semantic components was perhaps slightly more difficult. Although we did not show that semantic component indexing was clearly easier than keyword indexing, the results are encouraging given that semantic component indexing was entirely new to these indexers, whereas keyword indexing was a familiar task. Similarly, it is not surprising that the indexers had somewhat more confidence in

their ability to perform a familiar task, keyword indexing, and a slight preference for performing that task. Despite having less confidence in their ability to perform semantic component indexing well, the indexers were fairly positive about the potential usefulness of semantic component indexing for searching. Only two indexers would prefer to have keyword indexing when they search, whereas four indexers thought semantic components would be better for searching.

7.1.3. Discussion of Indexing Study Results

Although the indexers in our study perceived keyword indexing to be only moderately difficult, their agreement with the indexing standard and with each other was quite low. Semantic component indexing was a new, unfamiliar task and yet they perceived it as similar in difficulty. The accuracy and consistency of the two types of indexing are not directly comparable because the indexing tasks are somewhat different and are assessed using different metrics and different units of measurement. Nevertheless, we note that the data suggests more agreement of the indexers, both with the reference standard and with each other, for semantic component indexing than for keyword indexing.

Lancaster suggests that indexing is more likely to be consistent when terms are displayed "to remind an indexer that they *must* be used whenever applicable" [22] and notes that the Funk and Reid study supports this idea. It is possible that semantic component indexing can promote consistency by using a small schema so that indexers have relatively few semantic components to choose from within a given

document class. The indexing application can further support indexing quality by providing explicit reminders of what semantic components are available in the menu that appears each time a user highlights text and performs a right click.

Although our data is not sufficiently accurate to draw firm conclusions about the time required for indexing, it appears that semantic component indexing took slightly longer than keyword indexing and that the time was in the same general range. We speculate that reading (or at least skimming) and comprehending the document may take a similar amount of time for both types of indexing and might be a relatively large component of the total indexing time regardless of indexing method. A recent evaluation of a machine-aided indexing system [152] suggests that much of the time and effort of indexing is attributable to reading and understanding the document. If so, then reading time may set a lower bound on time required for manual indexing. Whether one type of indexing requires more in-depth reading or understanding of the document than the other type of indexing is unknown.

7.2. Indexing To Support A Searching Study

For the searching study described in Chapter 8 we used random sampling and purposeful browsing to further analyze the sundhed.dk document collection. We created a semantic component schema consisting of six document classes and associated semantic components. Seven experienced indexers collectively indexed 371 documents using this schema. In Chapter 8 we discuss how we selected the documents that were indexed. The indexers consisted of one member of the research

team (Dr. Nielsen) and six indexers from sundhed.dk. Most of the indexers had participated in the indexing study just described and all them had received training about semantic components and training about how to use our semantic component indexing software.

Instead of indexing the documents on paper, the indexers used the indexing prototype that is described in Section 7.2. The indexing application automatically logged timestamps, recording when each document was first displayed and when the indexer submitted the indexing for that document. The mean indexing time was 3 minutes 28 seconds, with a minimum time of 6 seconds and a maximum time of 60 minutes and 3 seconds. The maximum time may be an artifact because it is possible that the indexer left the application in an unfinished state during lunch. The next longest time was 45 minutes and the next shortest time was 9 seconds. Figure 7.9 shows the distribution of the indexing times required for these 371 documents. Most of the documents required less than 5 minutes to index.

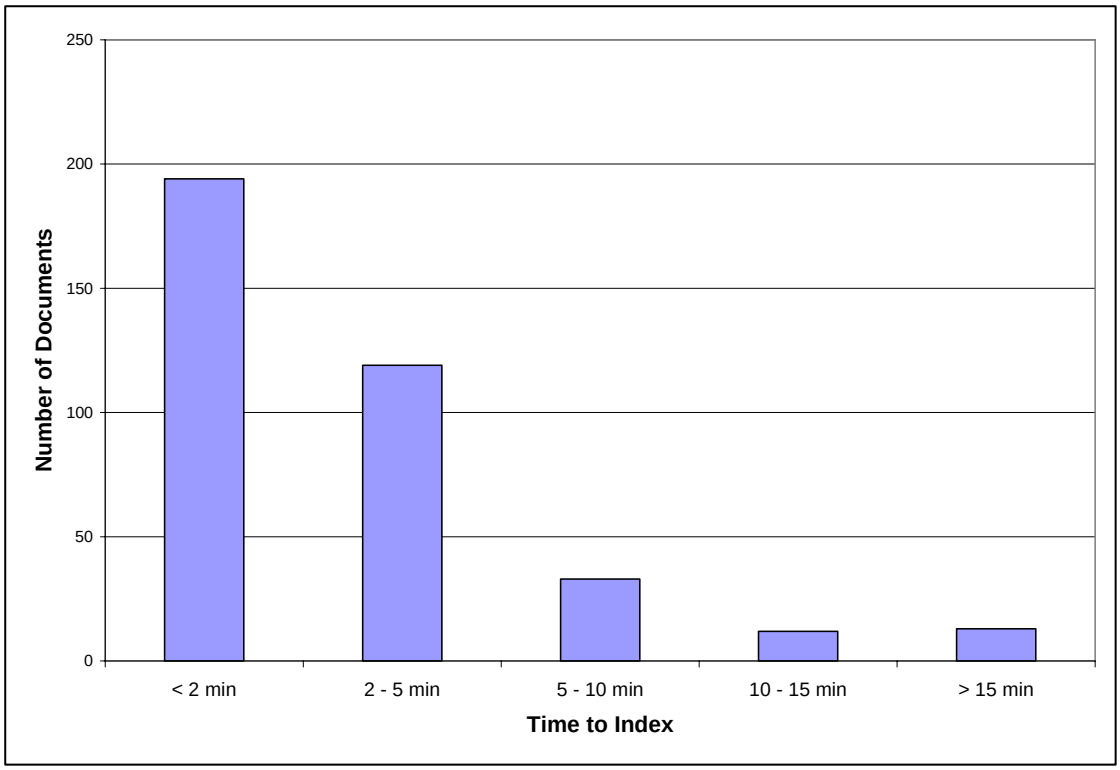


Figure 7.9 Distribution of indexing times: indexing to support the searching study

These times are noticeably shorter than the times recorded during the indexing study. Six factors might have contributed to the shorter indexing times:

1. The indexing application has a menu-driven interface and eliminates the need to manually record the semantic component labels.
2. The indexers indexed more documents for the searching study than for the indexing study. Their indexing speed might have increased as they gained familiarity with both the process of semantic component indexing and with the semantic component schema.
3. The indexers who volunteered to participate were probably those who were most comfortable with semantic component indexing during the indexing study.

4. The mean document length might have been shorter.
5. The schema used in the searching study was a refinement of the schema used in the indexing study and might have been a better reflection of the documents
6. We might have described the semantic components better, leading to a clearer understanding about what kind of information belonged in each component.

7.3. Discussion

The accuracy and consistency data provide an overall assessment of the quality of indexing and also allow the researcher to determine whether particular documents, document classes, semantic components, or keyword vocabularies are more problematic than others. In our indexing study, it appears that the semantic components for the *Services* documents should be reconsidered. Both the accuracy and the consistency are lower for these semantic components than for the semantic components in other document classes. This discrepancy might indicate that the set of semantic components we used do not provide a good description of the contents of this class of documents or that the descriptions are not adequate for consistent use by the indexers.

The data from the keyword indexing portion of the study suggests that indexers had more trouble with both the *Procedure* documents and the *Services* documents than with the *Clinical Problem* documents. We conjecture that the indexing vocabularies are not adequate for describing documents in those two document classes. The difference in keyword indexing quality between the *Procedure* documents and the

Clinical Problem documents was larger than the difference in semantic component indexing quality between the same two document classes. Although the quality measurements for the two types of indexing are not directly comparable, these differences suggest that semantic components might be especially useful when an appropriate indexing vocabulary does not exist. The variations in accuracy and consistency across the semantic components highlight the importance of carefully developing the right semantic component schema and providing good descriptions of the kinds of information each semantic component should contain. It is likely that additional training and practice could increase the accuracy and consistency of semantic component indexing. (Additional training might increase the accuracy and consistency of keyword indexing as well.)

As stated earlier, semantic component indexing and keyword indexing are not directly comparable because, although we used recall and precision to measure accuracy for both kinds of indexing and we used K_α to measure consistency for both kinds of indexing, the units of measurement are different. Evaluating instances of semantic component indexing compares the binary classification of text units (characters) within documents whereas evaluating keyword indexing compares variably sized sets of keywords that are extrinsic to the documents. The substantial differences in the range of numbers produced for the two types of indexing suggests that accuracy and consistency might be higher for semantic component indexing. However, assessing the effects on searching of keyword indexing and semantic

component indexing will provide an important comparison between the two types of indexing.

Despite their unfamiliarity with semantic component indexing, the indexers' perception of task difficulty was quite similar between the two types of indexing. Only slightly more indexers were more confident in their keyword indexing than in their semantic component indexing. The indexers rated semantic component indexing difficulty almost exactly the same as choosing which concepts should be indexed with keywords. This similarity suggests that both types of indexing may share the same underlying intellectual tasks that determine the overall difficulty of indexing, such as comprehending the text and recognizing the important concepts in the document.

Although semantic component indexing took the indexers in the study slightly longer than did keyword indexing, the average time to perform semantic component indexing for the documents used in the searching study was faster than the average time for either type of indexing during the indexing study. In Section 7.2 we discussed several possible explanations for the faster indexing. Overall, the scalability of manual semantic component indexing appears to be in the same general range as for manual keyword indexing when we consider indexing quality, perceived difficulty, and the time required for indexing. If appropriate keyword indexing vocabularies are not already available, semantic component indexing may be preferable because the semantic component schema can be customized to a particular document collection and should take less time to develop than a comprehensive keyword vocabulary.

The research just described has limitations. We studied only sixteen indexers and twelve documents in a single domain. Additional studies in different document collections and domains are needed to confirm the feasibility of semantic component indexing. Even if manual semantic component indexing is as scalable as manual keyword indexing, any type of manual indexing is infeasible for many document collections and settings due to limited resources. On the other hand, automating semantic component indexing could extend its usefulness considerably if the indexing quality is sufficiently high. The research reported in Chapter 6 and in this chapter provides a foundation for pursuing automated semantic component indexing. We have created a framework for evaluating semantic component indexing and have shown that manual semantic component indexing is sufficiently scalable for creating data sets for training and evaluating automated indexing applications. However, semantic component indexing is only worth pursuing if it can enhance searching. In Chapter 8 we describe an interactive searching study that explores whether semantic component indexing can enhance search results.

7.4. Summary

We assessed the feasibility of semantic component indexing by comparing semantic component indexing to keyword indexing in a user study. Sixteen indexers indexed twelve documents, half with semantic component indexing and half with keyword indexing. We reported data for accuracy, consistency, time required for manual indexing, and perceived difficulty of indexing. Both types of indexing had

quality that varied, especially by document type. We cannot directly compare the values, but the data suggests that agreement for semantic component indexing might be better than for keyword indexing. In particular, the quality of semantic component indexing appears to be less sensitive to document type than keyword indexing, which is affected by the suitability of the keyword indexing vocabularies for the documents being indexed. Our results also suggest that semantic component indexing is similar to keyword indexing with respect to indexing time and perceived difficulty.

We also reported the time required to perform semantic component indexing for 371 documents that were used in the searching study. The data from these 371 documents indicate that it may be possible to perform semantic component indexing substantially faster than the times we recorded in the indexing study.

Chapter 8 Searching with Semantic Components

In previous chapters we discussed developing a semantic component schema, using semantic components to express information needs, and identifying semantic components in documents. The real test of whether semantic components are useful, however, is whether semantic components can help retrieve documents. In this chapter we describe the first experiment to investigate the effect of semantic components on searching. We report on the retrieval performance of an implementation of the semantic components model on top of an existing information retrieval system. Our general goal was to answer the question: *Are semantic components useful for retrieving documents*?

More specifically, we asked:

1. Can physicians using a search system with semantic components formulate queries that result in better search performance than when using a basic system without semantic components?
2. Can physicians using a search system with semantic components successfully complete more search scenarios than when using a basic search system without semantic components?
3. Can physicians using a search system with semantic components successfully complete search scenarios more quickly than when using a basic search system without semantic components?

4. Are physicians more satisfied with the searching experience and with search results when using a search system with semantic components that when using a basic system without semantic components?

In the next section we describe the methods we used to investigate these questions.

8.1. Experimental Methods

First we describe the experimental search system, including the documents, the search engine, the interfaces, the indexing, and the results display. We then describe the design of the study, including the subjects, the organization of the study sessions, the scenarios, the relevance judgments, and the evaluation metrics.

8.1.1. Experimental Search System

We created an experimental search system based on the existing sundhed.dk portal that consisted of documents, a search engine, and two different search interfaces. Figure 8.1 shows a schematic of the experimental search system.

8.1.1.1. Documents

With the permission of sundhed.dk, we copied all 24,712 documents owned by sundhed.dk as of July 2006 (including keyword and metadata fields). These documents, formatted as web pages, contain information about health and healthcare and also about the Danish healthcare system. Some information is written for

healthcare providers and some for patients and their families, but all the documents are available to anyone on the public web portal.

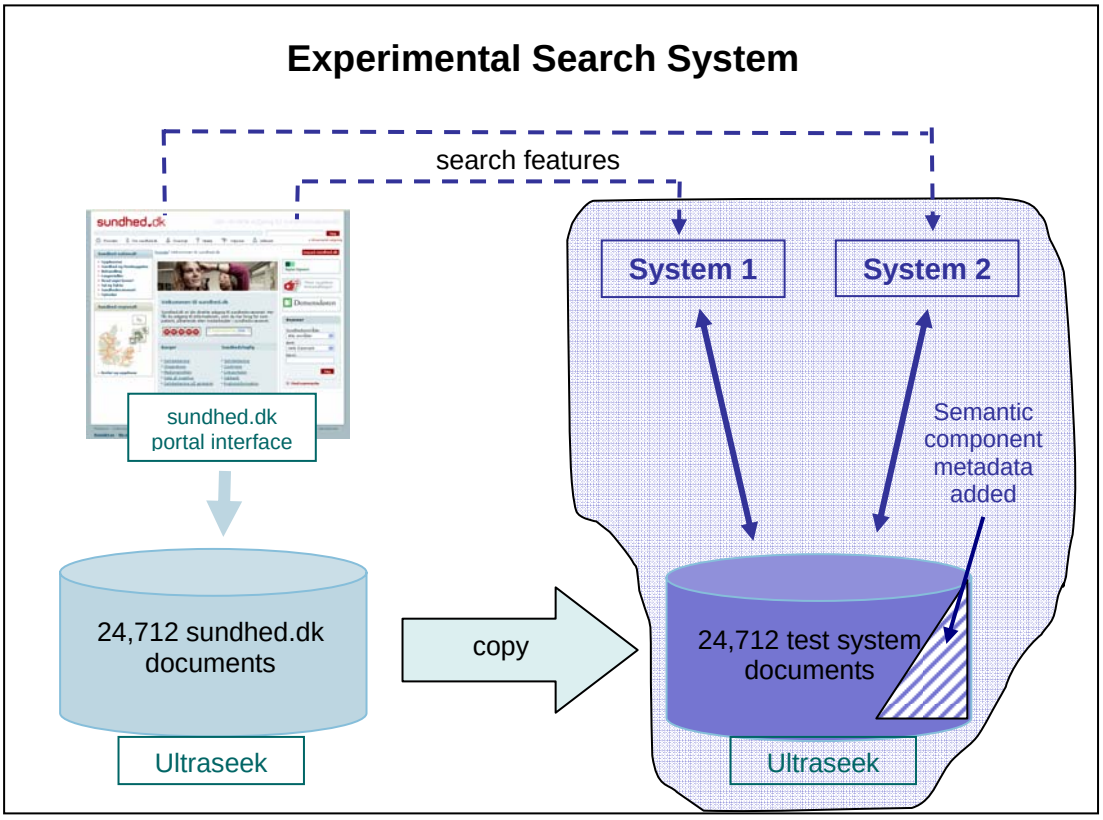


Figure 8.1 Schematic of the experimental search system

8.1.1.2. Search Engine

The operational sundhed.dk web portal uses Ultraseek [147], a commercial search engine developed by Verity Inc., and subsequently acquired by Autonomy Corporation [153]. We were granted a temporary license for the Ultraseek 5.6 software by Ensight (now Metier), the Danish distributor for Verity/Autonomy products. Sundhed.dk gave us copies of its configuration files so that we could mimic the operational system.

Ultraseek provides three main functionalities in the sundhed.dk portal: (1) it indexes all the documents, (2) it generates a search interface, implemented as a web page, and (3) it performs requested searches and generates a web page with a ranked list of links to documents that comprise the search result. Both the indexing and the search interface are customizable by setting parameters through an administrative interface and by editing the code that generates the user interface. Ultraseek performs full text indexing of the body of the document and also indexes metadata fields that are specified in the configuration file. The internal algorithms for searching the indexes and for ranking results are proprietary and cannot be viewed or modified. Documentation on the Ultraseek website describes the scoring algorithm in general terms as taking into account term frequency, term location within the document, rarity of individual terms, occurrence of multiple query terms, and document quality "based on numerous factors" [154].

8.1.1.3. Search Interfaces

The operational sundhed.dk site offers two search interfaces, a simple search (a single search box only) and an advanced search that provides several filters and the ability to designate terms as desired or required. We created two interfaces to our search system that we labeled as System 1 and System 2. The System 1 interface consists of a simple search box plus two filters from the sundhed.dk advanced interface that are controlled by pulldown menus, one to filter documents by the region of Denmark to which the documents are applicable (labeled *Regionalt indhold* in the

interface) and one to filter documents by an existing document classification (labeled *Informationstype* in the interface) used by sundhed.dk. We included these two filters after discussions with physician users and indexers, plus a review of the sundhed.dk search log, indicated these filters to be useful and frequently used. The default behavior for both filters is to include all documents (apply no filter). Queries typed into the search box use the Ultraseek query syntax, which includes wildcard expansion when an asterisk is included in a search term. Figure 8.2 shows a screenshot of the System 1 interface.

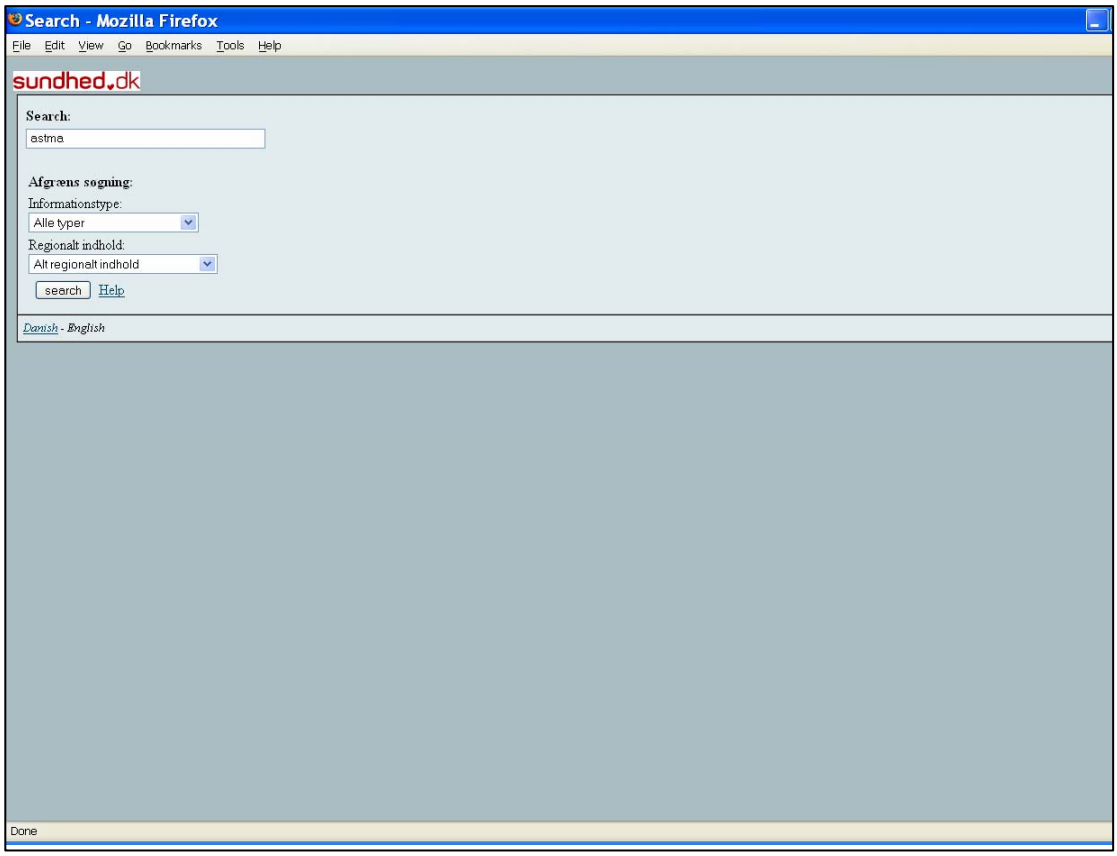


Figure 8.2 Screenshot of System 1 search interface

System 2 has the same features as System 1 plus the ability to further specify the search using semantic components. To search using System 2, the searcher types one or more search terms into the search box labeled *Search* and optionally chooses an item from the pulldown menus for the two filters, as when using System 1. In addition, the searcher can (optionally) enter one or more search terms into one or more of the text boxes for the semantic components. Figure 8.3 shows a screenshot of the System 2 interface. The text boxes are grouped by document class (the name of the class is in bold font) and are labeled with the semantic component. The green circle highlights a search term (the Danish equivalent of pregnan*) in a text box associated with the semantic component for *treatment*. System 1 was produced by standard Ultraseek code, configured to mimic the operational system. System 2 was based on the Ultraseek code for System 1 but required extensive customization of the Python code that produces the web interface.

8.1.1.4. Document Indexing

We indexed the documents using a semantic component schema that is the third refinement of a schema for the sundhed.dk document collection. As discussed in Chapters 4 and 7, we iteratively improved the schema as we gained more experience and knowledge about the documents and the users of the documents. This version of the schema consists of six document classes and associated semantic components. Table 8.1 shows the schema, with English translations of the Danish labels.

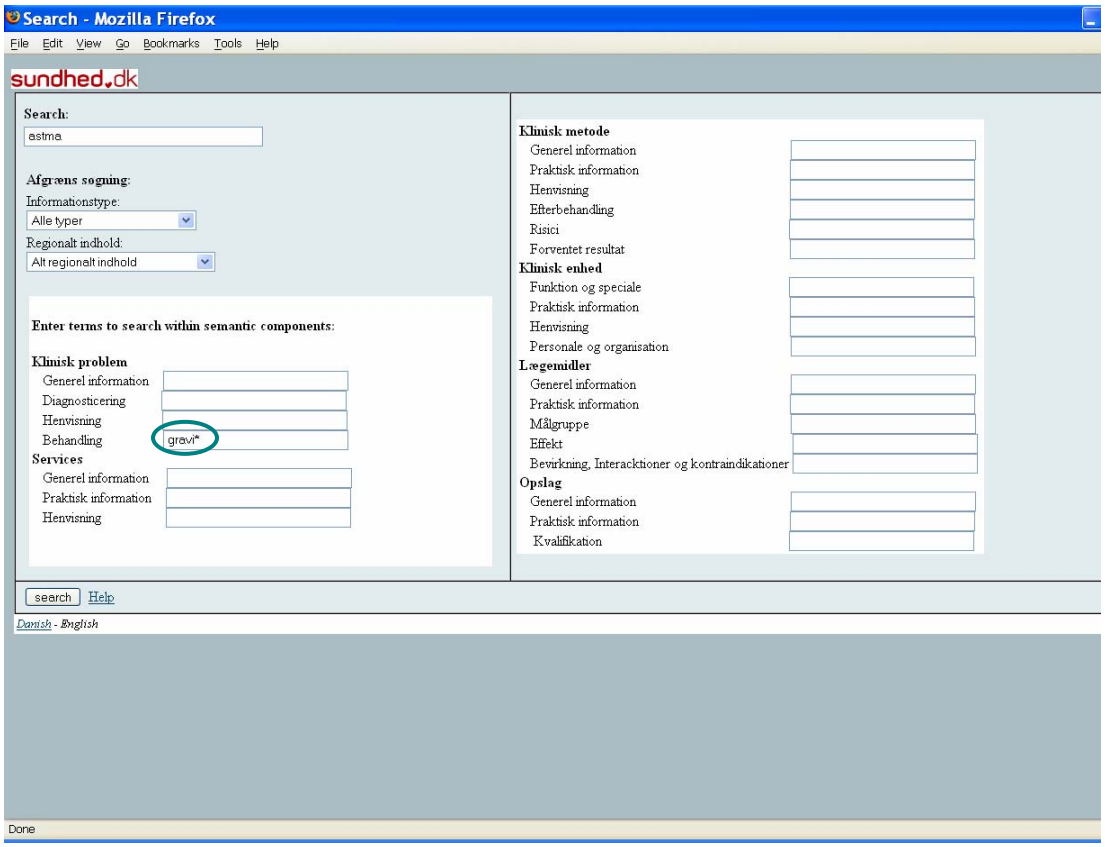


Figure 8.3 Screenshot of System 2 search interface

Because it was not feasible to manually index the semantic components in 24,712 documents, we chose a subset of documents for indexing by executing a variety of searches applicable to each of the four scenarios for the searching study. Our goal was to identify documents most likely to be retrieved at a high rank by the users, plus all documents relevant to the search scenarios. We selected the documents to be indexed using the following method. First, we composed five to seven queries per scenario that we thought searchers were likely to use. We entered each query into the operational sundhed.dk interface and programmatically extracted the ranked results. For each scenario, we merged the results lists from all the queries for that scenario so

that we had one ranked list per scenario. The merging algorithm took into account how many of the queries returned a particular document and the average rank at which the document was returned. We then used a round-robin algorithm to create a single priority list of the documents to be indexed with semantic components. The round-robin algorithm ensured that our indexing resources were allocated approximately equally among the four scenarios.

Table 8.1 Semantic component schema for the searching study

Document Class	**Semantic Components**	**Document Class**	**Semantic Components**
Klinisk problem (Clinical problem)	*Generel information* (general information) *Diagnosticering* (diagnosis, evaluation) *Henvisning* (referral) *Behandling* (treatment)	*Klinisk enhed* (Clinical unit)	*Funktion og speciale* (function and specialty) *Praktisk information* (practical information) *Henvisning* (referral) *Personale og organisation* (staff and organization)
Klinisk Metode (Clinical method)	*Generel information* (general information) *Praktisk information* (practical information) *Henvisning* (referral) *Efterbehandling* (aftercare) *Risici* (risks) *Forventet resultat* (expected results)	*Lægemidler* (Drugs)	*Generel information* (general information) *Praktisk information* (practical information) *Målgruppe* (target group) *Effekt* (effect) *Bivirkning, Interacktioner og kontraindikationer* (side effects, interactions and contraindications)
Services (services)	*Generel information* (general information) *Praktisk information* (practical information) *Henvisning* (referral)	*Opslag* (Notice)	*Generel information* (general information) *Praktisk information* (practical information) *Kvalifikation* (qualification)

Seven experienced indexers, who had received training about semantic components and training about the use of our semantic component indexing software,

indexed 371 documents. The indexing software is described in Section 3.1. We stored the semantic component data in metadata fields that we added to each indexed document. Data included the indexer-assigned document class, a list of the semantic components present in the document, the size of each semantic component instance (the number of characters in the instance), and the text in each semantic component instance. After configuring Ultraseek to index our newly-defined metadata fields, we indexed the full text and metadata fields (including both the metadata fields in the original document and also the semantic component metadata fields, when present) in all 24,712 documents with Ultraseek.

After completing the searching experiment, we retrospectively analyzed the distribution of documents indexed with semantic components. We did not want to bias the results by indexing only the relevant documents that contained words related to the scenarios, so we deliberately indexed documents likely to be returned by searches for the four scenarios. In other words, in addition to relevant documents we indexed the nonrelevant documents most likely to compete with relevant documents for ranking. To assess our results, we calculated the percentage of retrieved documents that had been indexed and the percentage of highly relevant documents that had been indexed. If a difference between systems were due only to System 2 preferentially returning indexed documents, the percentage of highly relevant documents in the result would be directly related to the percentage of indexed documents in the result. We describe the analysis in more detail in Section 8.2.5.

8.1.1.5. Retrieval and Results Display

We configured both System 1 and System 2 to return 100 hits, ordered by similarity score. System 1 returned documents using the Ultraseek similarity algorithm based on full text indexing of the title, body, keywords, and designated metadata fields. If a value was selected for either of the two filters, *Informationstype* or *Regionalt indhold*, documents matching the topical query term(s) were returned only if the document also contained the appropriate value in the metadata field for the selected filter(s). System 1 did not search metadata fields representing semantic components.

System 2 sent the query in the main (simple) search box plus the values for the two filters, if any, to the Ultraseek search engine exactly as in System 1. Unlike System 1, System 2 intercepted the result list and similarity scores, and sent a second query with the terms that were entered into the semantic component fields as a fielded search of the indicated semantic component metadata fields. The similarity scores for the second search were determined solely by the similarity of the semantic component part of the query to the corresponding semantic component instances in the retrieved documents. Documents without an instance of the requested semantic component were not returned from the second search and were assigned a similarity score of zero (for the second search). An asterisk in a query term acted as a wildcard and matched any text in any word. If only an asterisk and no other characters were entered in a search box for a semantic component, the asterisk acted as both a wildcard and a filter. In other words, the asterisk matched any text, but only documents that contained an

instance of the requested semantic component were returned from the second search. Documents were returned to the user only if they appeared in the result of the first "topical" query. Document ranking was determined by a final similarity score that was computed as the average of the similarity scores from the two searches (the search based on terms in the main search box and the search based on semantic components).

In summary, System 1 returned documents (that matched the filters, if any) ordered by their similarity to a simple query as calculated by Ultraseek based on full text indexing and keyword indexing. System 2 returned documents (that matched the filters, if any) ordered by the average of the similarity to a topical query and the similarity of any queries applied to particular semantic components. System 2 returned exactly the same documents that would have been returned by System 1 (re-ranked) unless a query to System 2 included an asterisk-only semantic component query, in which case it returned only documents from the topical query that also contained an instance of the semantic component.

The results displays for both System 1 and System 2 mimicked the operational system. Both systems displayed the title, a snippet of text showing the query term in context, the document ID, the region (if any) for which the document was written, the document type (*Informationskategori*) used by the operational system, and a summary written by the document author. In addition, System 2 also displayed: (1) the document class selected by the indexer from our list of six document classes (*Documenttyper*) and (2) a list of semantic components appearing in the document plus an integer to indicate the size, in number of characters, of the semantic

component instance. Figures 8.4 and 8.5 illustrate the results display from System 1 and System 2, respectively.

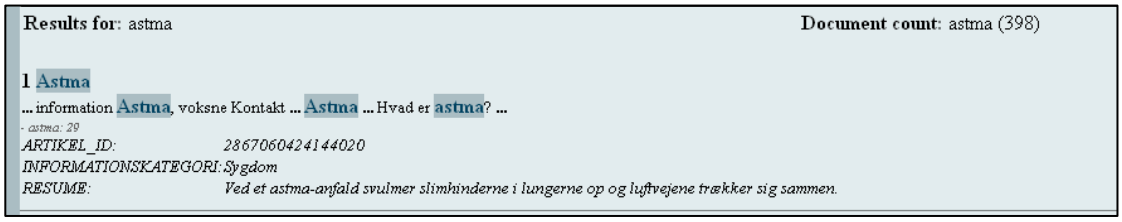


Figure 8.4 Cropped screen shot of System 1 results display

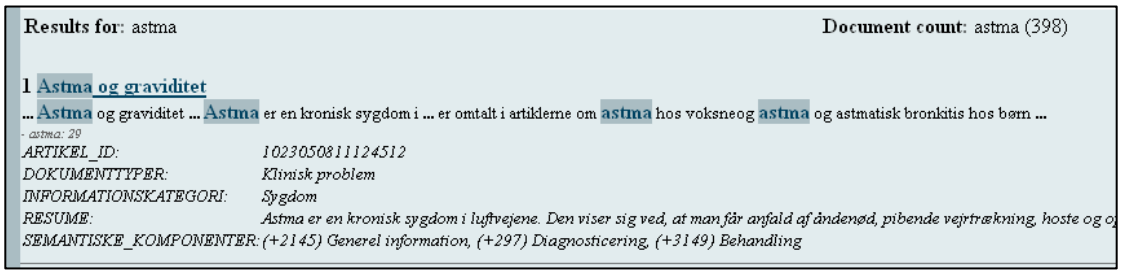


Figure 8.5 Cropped screen shot of System 2 results display

8.1.2. Experimental Design

8.1.2.1. Subjects

A convenience sample (as distinguished from a random sample) of 30 Danish family practice physicians from the Århus Region who were familiar with sundhed.dk participated in the searching study. The physicians were paid an amount equivalent to what they would have earned in their practice during the two hours of the study plus travel expenses. The study received prior approval from the Portland State University

Human Subjects Research Review Committee. Table 8.2 summarizes the self-reported medical and searching experience of the 30 participants.

Table 8.2 Searcher characteristics

Searcher Characteristic	Value ± Std. Dev.
Experience using Internet search engines	7.2 ± 2.8 years
Experience using sundhed.dk to find information about health care or the healthcare system (not patient data)	2.4 ± 1.4 years
Self described level of searching experience on a scale from 1 (not at all experienced) to 5 (very experienced)	2.4 ± 0.9
Experience as a medical professional	21.4 ± 7.6 years

8.1.2.2. Study Organization

We studied each subject separately in a two hour block that consisted of a training session followed by an experimental session. We performed the studies during five consecutive days to maintain consistency. Each study session followed the same sequence. The *training session* consisted of an introduction to semantic components and to the interfaces for Systems 1 and 2 plus a series of guided searches using the two systems. Each training session lasted about 45 minutes. The *experimental session* consisted of four search sessions, one for each scenario. We define a *search session* as the set of queries issued by a single searcher for a single scenario. Each subject used System 1 for two scenarios and System 2 for two scenarios. We randomized the order of scenarios and system use. Fifteen physicians used System 1 for their first two scenarios and System 2 for their second two scenarios. The other fifteen physicians

used the two systems in reverse order. We also varied the order of the scenarios in a random fashion. We randomly selected 15 of the 24 possible sequences of four scenarios; these were randomly assigned to two physicians, one who started with System 1 and one who started with System 2. Figure 8.6 depicts the organization of the study sessions and Table 8.3 summarizes the randomization of participants to sequences of systems and scenarios.

Training Session

Introduction to the study
Preliminary demographic survey
Training and guided practice with System 1 and System 2 interfaces

Experimental session

15 physicians		15 physicians	
System	**Scenarios**	**System**	**Scenarios**
System 1	First scenario Second scenario	System 2	First scenario Second scenario
System 2	Third scenario Fourth scenario	System 1	Third scenario Fourth scenario

Figure 8.6 Study organization

The searcher used one of the two interfaces to enter queries and view the results. The searcher could click on any of the hits in the result list to view the full document. For any documents the searcher considered relevant, we asked him or her to record an explicit relevance judgment. At the end of each scenario the searcher filled out a brief questionnaire. Each participant also completed a final questionnaire and participated in a short interview after having completed all four scenarios.

8.1.2.3. Scenarios

Each scenario represented a typical information need that might be encountered in the context of a patient visit in order to make a decision about patient care. We

developed the scenarios following the methodology of Borlund, who recommends evaluating systems using potential system users as test subjects and using simulated work tasks to motivate the searches [47]. The questions posed in the scenarios address specific aspects of medical care in the context of individual patient circumstances and are in line with prior work on clinical questions [9], adapted to the specifics of the Danish healthcare system and the information available in sundhed.dk. The scenarios each represent needs for information available in the sundhed.dk document collection but are of variable difficulty. We asked the searchers to search as they would in real life, letting the constraints of the clinical setting determine how long they would search and when they would either be satisfied or abandon the search. Table 8.4 provides a condensed summary of each scenario.

Table 8.3 Randomization of exposure of searchers to systems and scenarios

Day	Searcher ID	System 1		System 2		Day	Searcher ID	System 1		System 2	
	1	B	A	C	D		2	B	A	C	D
1	3	A	B	D	C	1	4	A	B	D	C
	5	C	A	D	B		6	C	A	D	B
	7	D	B	C	A		8	D	B	C	A
2	9	A	D	C	B	2	10	A	D	C	B
	11	C	D	B	A		12	C	D	B	A
	13	D	C	B	A		14	D	C	B	A
3	15	B	C	A	D	3	16	B	C	A	D
	17	D	C	A	B		18	D	C	A	B
	19	B	C	A	D		20	B	C	A	D
4	21	A	B	C	D	4	22	A	B	C	D
	23	B	C	D	A		24	B	C	D	A
	25	D	B	A	C		26	D	B	A	C
5	27	A	C	B	D	5	28	A	C	B	D
	29	C	D	A	B		30	C	D	A	B

Table 8.4 Scenarios

Scenario A	Ex-smoker; cough, fatigue, shortness of breath How should he be evaluated for emphysema?
Scenario B	Woman, 23 weeks pregnant, with vaginal bleeding Should she be referred for immediate examination?
Scenario C	Childless woman who has had two miscarriages and wants to become pregnant Should she take folate and at what dose?
Scenario D	Man who has been attacked with a knife, now nervous and afraid to leave his apartment alone Can he be referred for free psychological help (covered by the public insurance)?

8.1.2.4. Relevance Judgments

We used two sets of relevance judgments for this study, individual user judgments and a reference standard. We asked searchers to record a graded relevance judgment of 0 to 3 for documents that they viewed, or in a few cases, for relevant documents they were already familiar with and did not need to open to know the contents. We used the four point scale of Sormunen that classifies documents as *irrelevant*, *marginally relevant*, *fairly relevant*, and *highly relevant* [45]. Our reference standard consisted of graded relevance judgments made independently by a domain expert using the same scale. The standard included documents that were identified as relevant during scenario creation plus all documents identified as relevant (rating 1–3) by at least one searcher. Thus the reference standard incorporated searcher input but was developed independently of individual searchers' judgments. Whereas individual searchers typically identified a single highly relevant document for each session, the reference standard had multiple relevant documents per scenario, as shown in Table

8.5. Using the reference standard allowed us to assess the quality of complete ranked lists returned by each system.

Table 8.5 Number of highly relevant documents per scenario

Scenario A	Scenario B	Scenario C	Scenario D
3	1	9	3

8.1.2.5. Evaluation Metrics

We evaluated search system performance from multiple perspectives that can be grouped as two pairs of perspectives:

- The system perspective and the user perspective
- The single query perspective and the session-based perspective.

Table 8.6, at the end of this section, summarizes the evaluation strategy for the system perspective and the user perspective and shows how we considered both the single query and session-based perspectives for both the system and user perspectives.

Because we were simulating a search setting where information needs are very specific, we assigned gain values of 0, 1, 10, and 100 to documents with relevance ratings of 0, 1, 2, and 3, respectively, for all metrics based on discounted cumulative gain (for calculating $G[j]$). We used a factor of 10 to separate the values because marginally relevant and partially relevant documents are generally not very useful in the setting we simulated. Also, because the scenarios simulate a setting where time available for searching is limited, we used a discounting parameter of base 2 for DCG to simulate a "busy" user [43]. A larger parameter, such as base 10, results in a smaller discounting of relevant documents that appear later in a search and simulates a

more patient user. All results for metrics based on DCG use the newly modified version of DCG [44], shown in Equation (2) below.

Because this searching study was interactive, searchers often issued multiple queries for a single scenario. Little has been written about system evaluation in the presence of multiple queries. From a system point of view, the goal is to produce the best results for a given query. From the searcher point of view, the goal is to: (1) find the desired information, and (2) find it as quickly and efficiently as possible. The second goal can be satisfied by having the desired information appear early in a result set, by finding the desired information with as few queries as possible, or with some combination of these two outcomes.

We used two approaches to compare System 1 and System 2, each applied to both the system perspective and the user perspective. First, we defined a *best query* for each search session (where a session is all the queries posed by one searcher for one scenario). The best query in a search session is the query that had the best performance, as determined by a metric appropriate to the perspective being considered. We describe the metrics we used below. The best query approach allowed us to compare the results returned by the two systems given the user's best effort at using the query language provided by each system. Second, we looked at the gain provided by search results in the context of a sequence of queries in each session, using the new session-based discounting approach described below.

Session-based Discounting

Existing IR metrics evaluate the results of a single query per information need. Yet interactive searching often results in multiple queries for a single information need when a user reformulates his query in response to unsatisfactory search results. The session-based discounting methods presented below were developed in response to the need for a session-oriented methodology for comparing the performance of IR systems in our interactive searching study. The metric, motivations, proposed uses, examples of use, and the challenges of evaluating new metrics are discussed in detail elsewhere [44]. Session-based discounting is a method for evaluating search results in multiple-query sessions. Session-based discounting assigns value to each returned document not only according to its rank in a result list, as is done by the established discounted cumulative gain (DCG) metric [43], but also progressively discounts the results of each query after the first query in a sequence of queries. In an interactive search, results are penalized if the user must issue additional queries to find the desired result.

A session is a sequence of one or more queries that each yields a ranked list of documents. The session-based DCG (sDCG) metric produces a value associated with each ranked result for each query by discounting the DCG of each result according to the query iteration that produced the result.

$$sDCG[i] = (1 + \log_s q)^{-1} * DCG[i] \quad (1)$$

where i is the i[th] ranked result in query iteration q , bq[25] is the log base chosen for session-based discounting, and *DCG(i)* is calculated using a new version of the original DCG metric.

The original and modified versions of DCG are shown in Section 2.1.4. The original DCG only discounted documents that appeared at ranks greater than the logarithm base used to discount documents. The new version discounts all documents after the first document, regardless of the logarithm base used for discounting.[26] The modified DCG, used in calculating sDCG is:

$$DCG[i] = \sum_{j=1}^{i} \frac{G[j]}{(1 + \log_b i)} \qquad (2)$$

where j represents each document up to (preceding) and including document i in a ranked list and b is parameter chosen to govern the steepness with which document values are discounted as they appear further down a ranked list. $G[j]$ is the gain value assigned to the relevance score given to document j. Applying sDCG to each result in a ranked list for a query produces a vector of values for the query results. Like CG and DCG, sDCG can be normalized by dividing each value in the vector by the corresponding value in a vector that represents ideal search results (where the highly relevant documents appear first in a ranked list, followed by each less relevant document in the order determined by its relevance score). Normalization facilitates

---

[25] We follow the notation used in the published version of the metric, which uses bq to represent the log base. The log base, bq , is a single parameter, not the product of two variables.
[26] The modification was suggested by the author.

comparing results from search sessions for scenarios with different numbers of relevant documents.

Session-based discounting can be used in several ways. For example:

- The effectiveness of IR systems can be compared with respect to the cumulated gain (discounted for query iteration) of the best, or last, query issued in each session. The last query is of interest because it is usually the one that satisfied the user.
- The effectiveness of IR systems can be compared with respect to how early they return a particular relevant document, or the first relevant document in an interactive session.
- The effectiveness of IR systems can be compared with respect to the gain provided by concatenating the top *n* discounted results for the *q* queries in a session. This method assumes that a user views, on average, *n* results before reformulating the query.

We illustrate the use of sDCG in results presented below (Section 8.2.3). We used a base of 2 for the *bq* parameter in session-based discounting.

The System Perspective

For the system perspective evaluation, we used the reference standard to calculate results using the best query approach and using sDCG. We determined the best query for each session using two metrics, average precision (AP) and DCG. If multiple queries in the same session had identical AP or DCG, we designated the first such query as the best query. We used the graded relevance judgments in the reference

standard to calculate AP and DCG. Because AP is calculated using binary relevance judgments, we considered only highly relevant documents (relevance rating of 3) as relevant for calculating AP. This threshold ensured that only documents that satisfied the targeted information need in the scenarios were treated as relevant, a much stricter standard than is used in many retrieval studies, such as those using TREC data sets. Graded relevance judgments are inherent in the DCG metric, hence all relevance data are incorporated in DCG.

We chose AP and DCG after considering a variety of metrics popular in the IR literature. P@5, P@10, or R-Precision lacked enough power to discriminate among the queries in many of our sessions because of the sparseness of highly relevant documents. These metrics generated too many ties, and often equaled zero when AP or DCG was positive. If ties are ignored, none of those three metrics change which queries were the best queries. We also considered bpref, but found that it, too, was unsuitable. Bpref is generally robust to incomplete relevance judgments but it relies on comparing the ranks of pairs of relevant and judged nonrelevant documents (by this we mean documents that were explicitly judged as nonrelevant, not documents treated as nonrelevant because they were never judged). It requires that nonrelevant documents have as much chance of being judged as relevant documents. It also lacks discriminating power if the number of comparisons is too small [42]. The pool of documents judged by our domain expert were documents deemed relevant by human searchers, not a ranking algorithm. Therefore the pool was biased (relevant

documents were more likely to be explicitly judged than nonrelevant documents) and very small; we had few documents that were explicitly judged as nonrelevant.

AP is appealing because it reflects the quality of document ranking and has been termed stable and discriminating [41]. It only accepts binary relevance judgments, but in the setting that we simulated the searchers are usually interested only in highly relevant documents. Treating all other documents as nonrelevant is a reasonable choice. However, this choice resulted in a very small number of relevant documents, a situation in which most metrics are less stable. DCG may be more stable with few relevant documents than other metrics because, while it assigns more value to highly relevant documents, it incorporates ranking information about all relevant documents [155].

For session-based discounting we calculated sDCG for all documents returned by the best query in each session (i.e., at rank 100) and compared the mean sDCG for Systems 1 and 2. We also plotted the mean sDCG at each document rank for the two systems and plotted the concatenated sDCG for the ten top-ranked documents returned by each query in a session.

The User Perspective

We define a *successful* search session as a search session for which the searcher found at least one document to which the searcher explicitly assigned a relevance rating of either 2 or 3 (fairly relevant or highly relevant) on a scale of 0 to 3. Because most searchers stopped searching after finding a single useful document, we report

two metrics based on the rank of the *best user-relevant document*. We defined the best user-relevant document as the first explicitly identified relevant document found (during the earliest query, or if a query returned multiple relevant documents then the highest-ranked relevant document for that query) at the highest relevance level during a session. If the session resulted in finding at least one document to which the user explicitly assigned a relevance of 3, then the best user-relevant document was the first such document found. We defined this document as the best even if the user had already found a document with a relevance level of 2 earlier in the session. If the session resulted in no documents with a user-relevance of 3, but at least one document to which the user explicitly assigned a relevance level of 2, then the best user-relevant document was the first document found with an explicit user-relevance of 2. If no documents with a user-relevance of 2 or 3 were identified in a session, we considered it to be a *failed* search session. We define the *best user-relevant query* in each successful search session as the query in which the user identified the best user-relevant document. Our user-perspective search performance metrics evaluate the rank at which the system returned the best user-relevant document in two ways: (1) based solely on document rank within the results of the best user-relevant query, independent of how quickly the searcher formulated this particular query (i.e., the best query approach); and (2) based on how many queries preceded the best user-relevant query as well as the ranking within the query (i.e., the session-based discounting approach).

We used two metrics to evaluate performance based solely on document rank. The first metric is the reciprocal rank (RR) of the best user-relevant document, or $1/Rank_b$ where $Rank_b$ is the rank at which the best user-relevant document appeared in the list of search results. The reciprocal rank is independent of the number of search iterations that preceded the successful query and only reflects the performance of an individual query. The second metric is the discounted gain of the best user-relevant document (DG_{best}) in which a gain value is assigned to the relevant document, depending on relevance score, and is then discounted to reflect the rank at which the document appeared on the results list. This metric is based on DCG. For DCG, each relevant document is assigned a gain value based on it relevance score. Because most users quit after finding one highly relevant (or sometimes fairly relevant) document, we calculate the discounted gain of the best (first highly relevant) document instead of cumulating gain. We calculated $DG_{best} = gain * (1 + \log_b r)^{-1}$ to calculate the metric, where *gain* is the gain value assigned based on the relevance score of the document (either 10 or 100), r is the rank of the best document and b is the logarithm base for discounting by document rank. We chose 2 for the value of b for discounting to simulate an impatient user.

For session-based discounting we calculated sDG_{best}, which additionally discounts the value of DG_{best} to reflect the number of search iterations required to find the document. We calculated $sDG_{best} = DG_{best} * (1 + \log_{bq} q)^{-1}$, where bq is the logarithm base for session discounting and q is the position of the query in the sequence of queries that occurred in a session. sDG_{best} is an adaptation of the ideas underlying the

sDCG in which the gain contributed by a document is discounted to reflect both how far down a result list the user must look to find a given relevant document and the number of search iterations required to return the document. We chose 2 for the value of *bq* for discounting to simulate an impatient user. A larger value of *bq* would result in less severe discounting of the results of each subsequent query and thus simulate a more patient user, willing to issue more queries.

Table 8.6 Evaluation strategy

	System Perspective	**User Perspective**
Relevance judgments	Reference standard	Each user
Single query perspective	Average precision Best query defined by AP MAP for comparisons	Reciprocal rank of best document
	Discounted cumulative gain Best query defined by DCG nDCG for comparisons	DG of best document
Session-based perspective	sDCG of best query Concatenated sDCG of top ten results for each query	Number of successful search sessions Iteration number of best query Time to complete scenario Number of queries per session sDG of best user-relevant documents
User satisfaction		Ease of expressing search Satisfaction with results

8.1.2.6. Statistical Analysis

Search performance is influenced by both the search system and the scenario, and we hypothesized that system might interact with scenario, such that System 2 might have better performance on some scenarios and worse performance on others. As a result, we compared the systems using a mixed effect two-way factorial analysis of variance model [156]. The search system is a fixed effect since we are interested only

in comparing System 1 and System 2. Search scenario is a random factor; we studied only a small subset of all possible search scenarios but we are interested in being able to generalize to other scenarios. For all comparisons, we first determined the presence or absence of an interaction between search system and search scenario before determining whether results from System 1 and System 2 were statistically different.

8.2. Experimental Results

8.2.1. Search Performance Evaluated from a System Perspective for Single Queries

System 2 achieved a higher mean performance for each scenario, and for all scenarios combined, using either MAP or nDCG. Over all scenarios, the improvement was 35.5% as measured by MAP and 28.6% as measured by nDCG. Analysis of variance found no interaction between system and scenario for either MAP or nDCG. The difference between System 1 and System 2 was statistically significant for both MAP ($p < 0.02$) and nDCG ($p < 0.01$). As expected, given the varying difficulty of the scenarios, the difference among scenarios was highly significant using either performance metric. Tables 8.7 and 8.8 show the mean performance and standard error (SE) by system and scenario of the best (system-oriented) query for each session using AP (Table 8.7) and nDCG (Table 8.8). Figure 8.7 shows the shape of the average nDCG curve for each system over all scenarios combined. This plot indicates that System 2 returns relevant documents at rank 1 more often than System 1, and this early retrieval is responsible for its better performance.

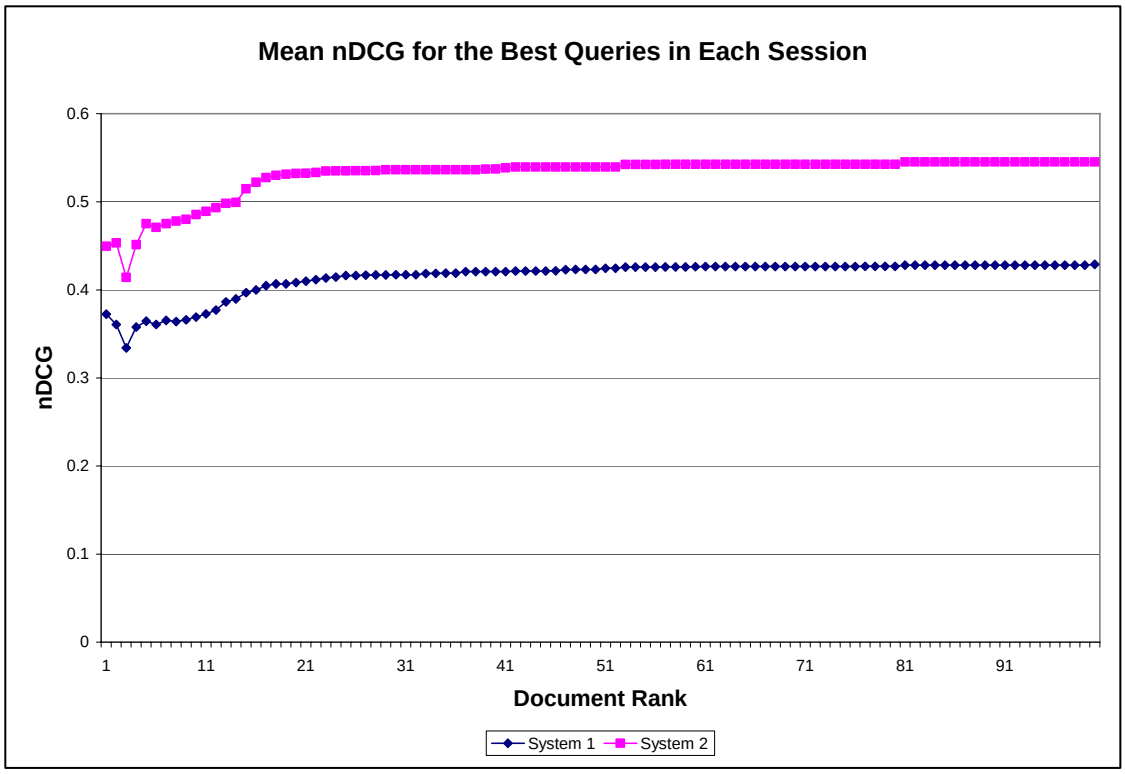


Figure 8.7 Mean nDCG of the best queries for all scenarios

Table 8.7 Average precision of the best query per session (mean ± SE)

System	Scenario A	Scenario B	Scenario C	Scenario D	All scenarios
1	0.28 ± 0.03	0.53 ± 0.11	0.21 ± 0.05	0.19 ± 0.03	0.31 ± 0.03
2	0.56 ± 0.06	0.58 ± 0.11	0.26 ± 0.03	0.27 ± 0.06	0.42 ± 0.04

Table 8.8 nDCG of the best query per session (mean ± SE)

System	Scenario A	Scenario B	Scenario C	Scenario D	All scenarios
1	0.44 ± 0.03	0.52 ± 0.09	0.38 ± 0.07	0.36 ± 0.03	0.42 ± 0.03
2	0.71 ± 0.05	0.55 ± 0.09	0.48 ± 0.04	0.41 ± 0.06	0.54 ± 0.03

We did not require searchers to use semantic components when using System 2. We compared performance of System 1 to System 2 without regard to whether a searcher used semantic components in a query submitted to System 2. We did this for three reasons: (1) the System 2 results display included information about semantic components regardless of whether the query used semantic components, (2) we wanted to assess the overall effect of making semantic components available to searchers and choices about whether to use a feature is part of such an assessment, and (3) we wanted to maintain the randomization applied at the beginning of the study. Using this approach ensured that the results are less likely to over-predict the effects of usage in an operational setting.

Twenty-nine of the 30 searchers used semantic components in at least some of their queries. Fifty-six (93%) of the 60 search sessions with System 2 contained at least one semantic component query. The best query included at least one semantic component in 46 (82%) of those 56 sessions, whether determined by AP or by DCG. For all but two System 2 sessions, the best query was the same when determined by either AP or DCG. In both cases, documents with relevance scores of 1 or 2 increased DCG but not AP. Queries to System 2 resulted in 506 instances of retrieving a document with a relevance score greater than zero. In 92 (18%) instances, the rank of a relevant document was changed by a semantic component part of the query. In 46 instances, reranking resulted in a relevant document being rated higher (appearing at a lower rank, i.e., higher on the results list). In the other 46 instances, reranking resulted in a relevant document being rated lower (appearing lower on the results list) but the

changes were of a smaller mean magnitude than for the rerankings that result in relevant documents being rated higher. Changes ranged from a rank improvement of 94 (from 96th to 2nd) to a 17-place worsening (from 24 to 41 and from 25 to 42). The mean change was an improvement in rank by 8.1 places.

The addition of semantic components consistently improved search performance as measured by MAP and nDCG for the best query in each session, suggesting that semantic components can be a valuable supplement to existing indexing techniques. The results reflect both the use of semantic component information in the query to return relevant documents and the ability of searchers to use the model to express information needs.

8.2.2. Search Performance Evaluated from a User Perspective for Single Queries

8.2.2.1. Successful Search Sessions

Thirty physicians each completed four search scenarios, resulting in 120 search sessions. Of the 107 successful search sessions, the searcher found at least one highly relevant document (relevance rating of 3) in 93 sessions (87% of successful sessions, 78% of all sessions). Only 14 sessions (12%) were terminated by the searcher after finding only a fairly relevant document. Eleven sessions (9%) were terminated by the searcher after finding no relevant documents, and two sessions (2%) were terminated after finding only a marginally relevant document.

Table 8.9 shows the number of successful search sessions using each of the experimental systems for each of the four scenarios. Each scenario was searched by 15 searchers with System 1 and 15 searchers with System 2. Across the four scenarios, searchers completed three more scenarios using System 2 than when using System 1. If the definition of success is restricted to only the highest relevance rating, the difference remains the same; System 2 resulted in 48 successful searches compared to 45 with System 1.

Table 8.9 Number of successful search sessions

System	Scenario A	Scenario B	Scenario C	Scenario D	All scenarios
1	14	9	15	14	52
2	15	10	15	15	55

Overall, searchers were quite successful at finding at least one relevant document. Use of System 2 resulted in more successful search sessions than System 1. However, a detailed examination revealed that for 12 of the 13 unsuccessful search sessions, failure to find user relevant documents could be attributed either to disagreements about whether a document was actually relevant to the scenario or to failure of the user to recognize a relevant document within a list of returned hits. Scenarios A and D each had one unsuccessful session. In both cases, at least one of the queries returned a highly relevant document (according to the reference standard) at a very low rank (1 and 4 respectively) that was never examined by the user. Scenario B had 11 failed sessions. Scenario B was controversial in that only one document was judged highly relevant in the reference standard. Five of the users who examined this document also judged it highly relevant, and three judged it fairly relevant, but 6 judged it irrelevant

and one judged it only marginally relevant. In 6 of the failed sessions, the user examined this document, did not find it relevant, and did not find another document he scored as relevant. In another five sessions, the highly relevant document was returned within the top 15 hits for at least one query (ranks 1, 1, 4, 4, and 15) but the searcher did not click on the document. Only one session, a search on Scenario B using System 2, was a failed search from the system perspective, in that the queries issued failed to return any documents with relevance ratings of 2 or 3.

For the sessions in which the user did not view documents considered highly relevant in the reference standard, we do not know whether the searchers would have explicitly regarded the document as irrelevant if they had viewed the document. We also do not know whether the title, summary, and snippet displayed in the results were not informative enough to indicate the document's relevance to the scenario. It is possible that System 2 was more helpful for recognizing relevant documents than System 1 because it provided additional information about each returned document.

8.2.2.2. Time to Complete Search Scenarios

We define the time to complete a search scenario as the time elapsed from when the searcher was given the scenario to read until the searcher declared that he or she was finished searching. The post-session interview and questionnaire were not considered as part of the time spent in a search session. Table 8.10 shows the mean time in seconds to complete each search scenario using either System 1 or System 2. Two-way analysis of variance revealed no interaction between system and scenario.

Scenario had a highly significant effect on time to complete a session ($p < .000005$). On average, search sessions took 1 minute and 22 seconds longer using System 2 than System 1, a difference that was statistically significant ($p < 0.02$).

Table 8.10 Time (in seconds) to complete search scenarios (mean ± SE)

System	Scenario A	Scenario B	Scenario C	Scenario D	All scenarios
1	265.7 ± 30.2	529.0 ± 49.8	293.5 ± 29.9	302.9 ± 38.0	347.8 ± 23.0
2	364.7 ± 55.8	554.7 ± 66.1	454.7 ± 46.4	346.9 ± 32.3	430.2 ± 27.4

Two factors may have contributed to sessions lasting longer when using System 2:

- Searchers issued more queries with System 2 (see below).
- Searchers were unfamiliar with semantic components and with the System 2 interface. Searchers might have spent extra time looking at the descriptions for the document classes and semantic components and deciding how to formulate queries.

8.2.2.3. Number of Queries Per Search Session

We define the number of search iterations per session as the number of queries issued during a session. Search sessions in our study ranged from 1 to 11 queries with a mean of 2.85 and a median of 2 queries per session. Table 8.11 shows the average total number of search iterations by scenario and by system. Searchers consistently entered more queries into System 2 than into System 1. We also report the iteration number at which the best user-perspective query (Table 8.12) and the best system-perspective query (Table 8.13) occurred in each session.

Table 8.11 Number of search iterations (queries) per session (mean ± SE)

System	Scenario A	Scenario B	Scenario C	Scenario D	All scenarios
1	1.67 ± 0.3	3.93 ± 0.6	1.87 ± 0.4	2.67 ± 0.5	2.53 ± 0.2
2	1.60 ± 0.3	4.33 ± 0.7	3.47 ± 0.7	3.33 ± 0.6	3.18 ± 0.3

Table 8.12 Iteration number of best user-perspective query in each session (mean ± SE)

System	Scenario A	Scenario B	Scenario C	Scenario D	All scenarios
1	1.67 ± 0.3	3.87 ± 0.6	1.87 ± 0.4	2.53 ± 0.4	2.48 ± 0.2
2	1.53 ± 0.3	3.73 ± 0.7	3.13 ± 0.6	3.33 ± 0.6	2.93 ± 0.3

There was no interaction between system and scenario for the number of search iterations per session, the iteration number of the best user-perspective query in each session, the iteration number of the best system-perspective query as determined by AP, or the iteration number of the best system-perspective query as determined by nDCG. The difference between search systems was not statistically significant with respect to the number of search iterations per session. Although the mean iteration number of the best user-perspective query and the best system-perspective query as determined by either AP or nDCG was greater for System 2 than for System 1, the difference was statistically significant only for the best system-perspective query as determined by AP ($p < 0.05$). The difference was not significant for either the best queries as determined by the reciprocal rank of the best user-relevant document or for the best queries as determined by nDCG. The difference between scenarios was highly significant for the number of search iterations per session ($p < 0.0001$) and the iteration number of the best user-perspective query ($p < 0.0005$). The difference between scenarios was less dramatic but still statistically significant for the iteration number of the best system-perspective query as determined by AP ($p < 0.04$) and by nDCG ($p < 0.005$).

Table 8.13 Iteration number of best system-perspective query by AP and by nDCG (mean ± SE)

System	Scenario A	Scenario B	Scenario C	Scenario D	All scenarios
AP					
1	1.47 ± 0.2	2.27 ± 0.4	1.6 ± 0.3	2.13 ± 0.3	1.87 ± 0.2
2	1.40 ± 0.2	2.67 ± 0.5	2.87 ± 0.6	2.87 ± 0.5	2.45 ± 0.2
nDCG					
1	1.47 ± 0.2	3.13 ± 0.6	1.60 ± 0.3	2.13 ± 0.4	2.08 ± 0.2
2	1.40 ± 0.2	2.73 ± 0.5	2.87 ± 0.6	3.33 ± 0.6	2.58 ± 0.3

Although System 2 ranked documents in a better order, as determined by the reference standard, the users did not find documents to satisfy the information needs in the scenarios more quickly with System 2.

8.2.2.4. Search Performance Based on Explicit User Relevance

Table 8.14 shows the mean search performance by system and scenario as evaluated using reciprocal rank (RR) and Table 8.15 shows the mean search performance as evaluated by the DG of the best user relevant document. Analysis of variance revealed no interaction between system and scenario with respect to reciprocal rank or discounted gain. The mean reciprocal rank was higher for System 2 than for System 1 for three of four scenarios, and the mean discounted gain of the best user-relevant document was higher for System 2 than for System 1 for all four scenarios. Both metrics were higher for System 2 for all scenarios combined, however, the difference was not statistically significant. The difference between scenarios was highly significant with respect to both the reciprocal rank of the best user-relevant document ($p < 0.0001$) and the mean discounted gain of the best user-relevant document ($p < 0.00000001$).

This data indicates that users found relevant documents at somewhat better ranks (higher on the results list) with their best queries when using System 2 compared to System 1, but the difference was not enough to be statistically significant. Although System 2 returned documents in a better rank order than System 1 (according to the reference standard), the individual searchers exhibited considerable variation with respect to their implicit and explicit relevance judgments and did not always agree with the reference standard. They often skipped over documents that were highly relevant in the reference standard, implicitly evaluating the documents as nonrelevant.

Table 8.14 Reciprocal rank of the best user-relevant document (mean ± SE)

System	Scenario A	Scenario B	Scenario C	Scenario D	All scenarios
1	0.75 ± 0.10	0.30 ± 0.10	0.45 ± 0.07	0.53 ± 0.11	0.51 ± 0.05
2	0.83 ± 0.08	0.47 ± 0.12	0.38 ± 0.08	0.58 ± 0.09	0.57 ± 0.05

Table 8.15 Gain, discounted by rank, of the best user-relevant document (mean ± SE).

System	Scenario A	Scenario B	Scenario C	Scenario D	All scenarios
1	70.58 ± 10.1	23.80 ± 9.0	38.44 ± 6.3	57.57 ± 9.8	47.60 ± 4.9
2	84.23 ± 7.1	26.31 ± 10.1	39.21 ± 7.5	60. 91 ± 7.7	52.66 ± 4.9

8.2.2.5. User Satisfaction

For both systems, we asked the searchers at the end of each session (1) *how easy was it to express what you wanted to find*, and (2) *how satisfied were you with the results of your search*. For both questions, subjects were asked to circle an answer on a 5 point scale in which 1 represented either *very easy* or *very satisfied* and 5 represented *very difficult* or *very unsatisfied*. Because only the extremes were labeled, we treated the answer scales as interval scales. Tables 8.16 and 8.17 show the results, by scenario and by system, of the ease of expression and satisfaction with search

results, respectively. Two way analysis of variance did not indicate a significant interaction between scenario and system with respect to ease of expressing the search. There was no significant difference between the two systems with regard to expressing the search, but there was a significant difference between scenarios ($p < 0.001$). With respect to search results, two way analysis of variance did indicate a significant interaction between scenario and satisfaction. The effect of scenario on satisfaction is not surprising given the evidence that it was much more difficult to find relevant documents for some scenarios than for others. Users were least satisfied with both systems after searching for Scenario B, the scenario that had controversial relevance judgments and for which users spent the most time and had the worst results (according to user relevance assessments). We also show the number of searchers who chose each score of the ease of expression and satisfaction with search results in Figures 8.8 and 8.9, respectively.

Table 8.16 Ease of expressing search (mean ± SE) 1 = very easy; 5 = very difficult

System	Scenario A	Scenario B	Scenario C	Scenario D	All scenarios
1	1.4 ± 0.3	2.7 ± 0.3	1.7 ± 0.3	2.3 ± 0.3	2.0 ± 0.2
2	1.3 ± 0.2	2.5 ± 0.4	2.1 ± 0.4	2.3 ± 0.3	2.1 ± 0.2

Table 8.17 Satisfaction with results (mean ± SE) 1 = very satisfied; 5 = very dissatisfied

System	Scenario A	Scenario B	Scenario C	Scenario D	All scenarios
1	1.3 ± 0.3	3.2 ± 0.4	1.5 ± 0.2	2.3 ± 0.4	2.1 ± 0.2
2	1.1 ± 0.1	3.5 ± 0.4	3.0 ± 0.4	1.7 ± 0.3	2.3 ± 0.2

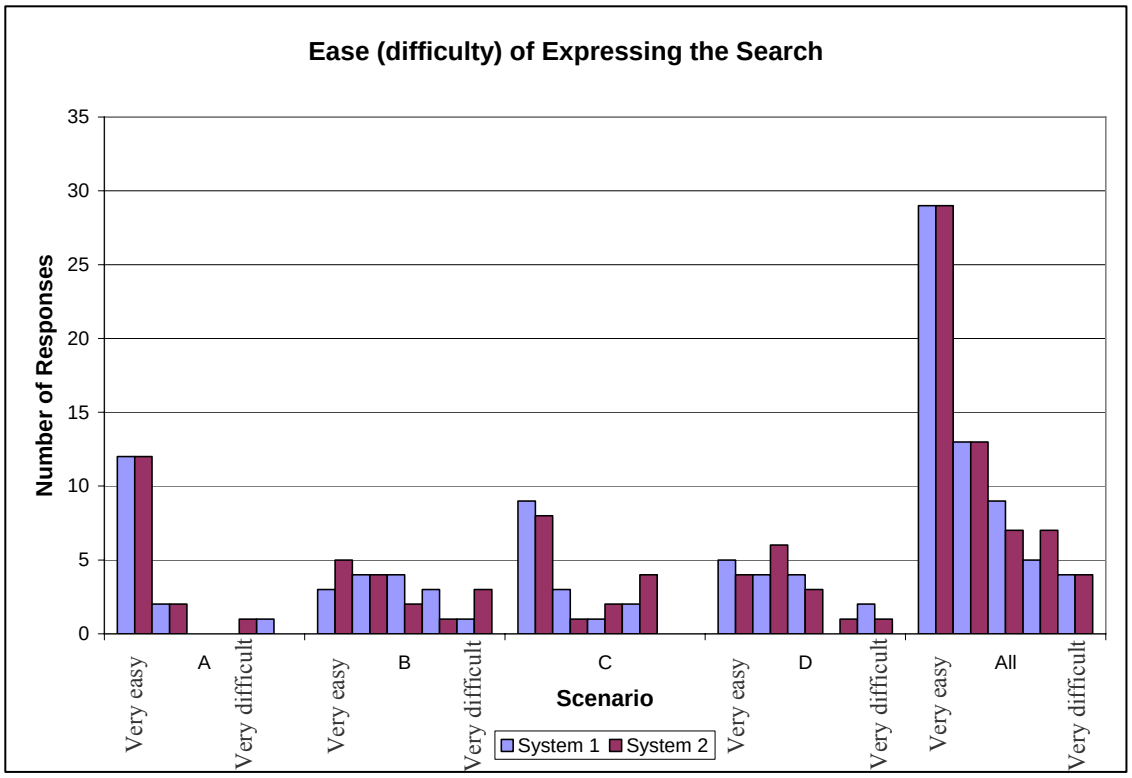


Figure 8.8 Survey responses regarding ease of expressing the search with each system

8.2.3. Search Performance Evaluated Using Session-Based Discounting

We calculated session-based discounting metrics for both the system perspective and the user perspective. Figure 8.10 shows a plot of mean sDCG at each document rank for the best queries in each session for the two systems. Table 8.18 shows the mean sDCG (at document rank 100) for the best system-perspective query in each session. Although the overall mean for System 2 was somewhat higher than for System 1, the difference between search systems was not statistically significant. When using System 2, the best queries tended to occur after more query iterations. Discounting for query iteration diminishes the apparent benefit of better document ranking by System 2 for the best queries.

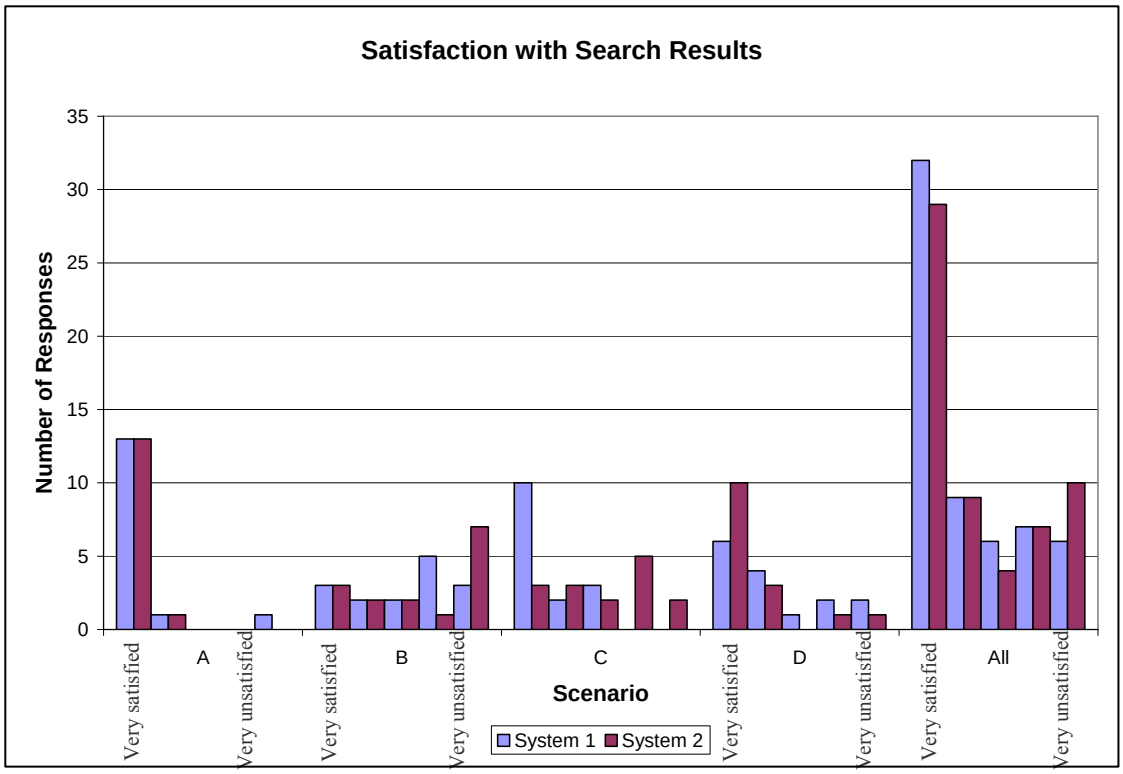


Figure 8.9 Survey responses regarding satisfaction with search results from each system

Table 8.18 sDCG of the best query per session (mean ± SE)
logarithm base = 2 for discounting rank and for discounting query iteration

System	Scenario A	Scenario B	Scenario C	Scenario D	All scenarios
1	74.7 ± 8.6	41.0 ± 10.0	110.2 ± 23.8	54.1 ± 8.8	70.0 ± 7.8
2	121.6 ± 12.5	36.3 ± 7.5	114.8 ± 14.1	49.1 ± 12.7	80.4 ± 7.6

Table 8.19 shows the mean sDCG for the best user-relevant document. Results varied by scenario but the overall difference between System 1 and System 2 was small and not statistically significant.

Table 8.19 sDG of the best user relevant document per session (mean ± SE)

System	Scenario A	Scenario B	Scenario C	Scenario D	All scenarios
1	64.10 ± 10.9	8.13 ± 3.0	29.85 ± 6.6	42.61 ± 10.0	36.17 ± 4.8
2	70.32 ± 8.7	18.29 ± 8.8	23.90 ± 6.8	33.98 ± 7.6	36.62 ± 4.7

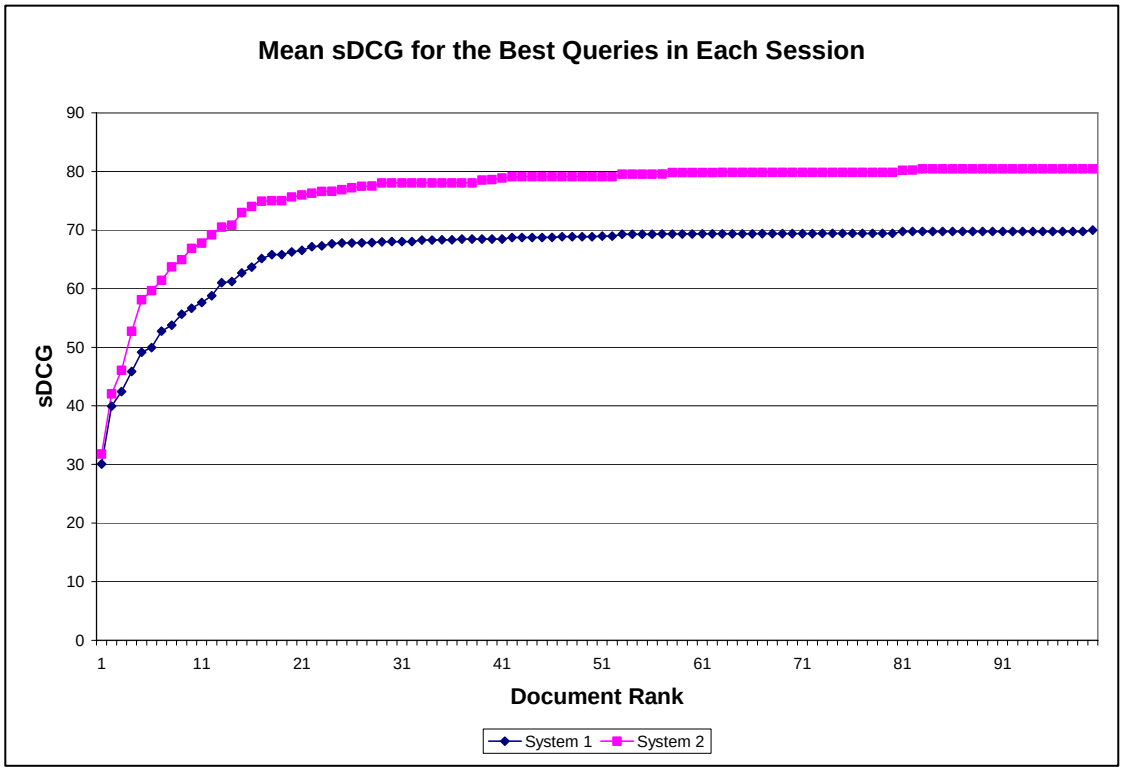


Figure 8.10 Mean sDCG of best queries for Systems 1 and 2

We also performed an analysis based on entire sessions. Although results lists displayed up to 100 hits, users rarely looked beyond the more highly-ranked hits. We did not try to capture data about how far down the list the users scanned, but we know that the user clicked on a document appearing after rank 14 in only 4 of 120 sessions (3.3%). The user clicked on a document ranked between 11 and 14 in only six additional sessions (5%). We therefore concatenated the top 10 documents from each query in a session, calculating sDCG on the single concatenated list of results for each session. If a query returned fewer than ten results, we treated the empty slots in the list as if they contained irrelevant documents since there is a cost to the user for formulating each query and looking at the list, even if the list is not full. Sessions with

only one or a few queries were treated as having no additional gain after the last query. We then averaged the resulting sDCG vectors for each system and plotted the results in Figure 8.11. We carried the results out to 110 places because one session had 11 queries. The concatenated results show that System 1 and System 2 are fairly similar for the first 20 ranks, corresponding to the first two queries, then System 2 has a consistently higher sDCG. The rise in sDCG is steeper for System 2 than for System 1 as results from each new query are added (at ranks 1, 11, 21 and so forth), reflecting the appearance of relevant documents in top-ranked positions.

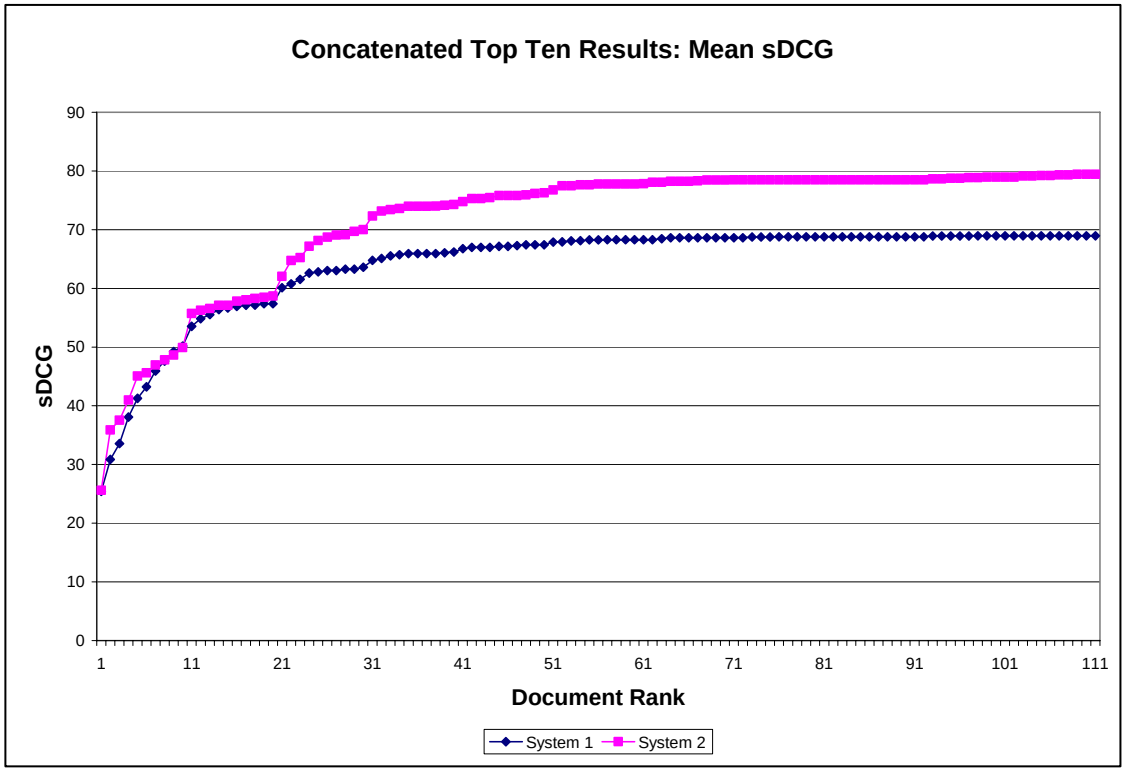


Figure 8.11 Mean sDCG for the concatenated top ten results of each query in a session

8.2.4. Effect of Relevance Assessments in the Reference Standard

We examined the possible effect on our results of disagreements with respect to relevance judgments. Sixteen of the 41 documents in the relevance standard were scored as highly relevant. For 14 of those 16, there was substantial agreement: all the users who clicked on those documents rated them as either highly relevant or fairly relevant. Only two documents were controversial. The single relevant document for Scenario B was judged highly or fairly relevant by eight searchers but was judged irrelevant by six and marginally relevant by one. The disagreement concerned (1) whether the document only applied to the first trimester of pregnancy, and (2) whether any search, instead of a phone call, was an appropriate action. One of the nine highly relevant documents for Scenario C was rated as either highly or fairly relevant by 12 searchers but two searchers rated it irrelevant.

Because the only highly relevant document for Scenario B was controversial, we also calculated nDCG when Scenario B was excluded (the system-oriented evaluation) and repeated the analysis of variance. The improved performance of System 2 was even more highly significant ($p < 0.002$). Exclusion of Scenario B from the performance evaluations based on explicit user relevance does not change the results of that analysis.

8.2.5. Effect of Document Selection for Indexing

We strategically chose documents to be indexed before the searching study and then retrospectively analyzed the effect of our choices because our resources for

manual semantic component indexing were limited. The searchers identified 37 documents as being at least marginally relevant (score ≥ 1) to one of the scenarios. The reference standard included those 37 documents plus an additional 4 documents that we identified before the study, which were not viewed by any of the searchers, resulting in a total of 41 documents. We indexed 30 of the 37 user-relevant documents and all 4 additional ones. Of the seven documents that were not indexed, only three received at least one relevance rating of 3 and none were scored as highly relevant in the reference standard.

Because we deliberately tried to index all relevant documents, we were concerned that the presence of semantic component indexing alone might bias the performance results. Of the 14993 hits returned by all 343 queries in the study, 5459 hits (36%) had been indexed with semantic components. (The same document could be returned by multiple queries). Of the 5459 indexed hits, 508 (9%) were highly relevant and 4398 hits (81%) were irrelevant. The remaining hits were marginally or partially relevant.

Table 8.20 shows the rate at which each system returned indexed documents (*Fi*) and the rate each returned highly relevant documents (*Fr*) in ranks 1 to 30 (in increments of 10), and at all ranks. We defined these rates as:

$$Fi = Ti/Tr \qquad \text{and} \qquad Fr = Ri/Tr$$

where *Ti* is the number of documents indexed with semantic components that were retrieved by a system over all queries, *Tr* is the total number of documents retrieved by a system over all queries (indexed or not), and *Ri* is the number of highly relevant

documents retrieved by a system over all queries (indexed or not). For asterisk-only semantic component queries, System 2 only returned documents with that semantic component and therefore *Fi* = 1.0 for those queries. This effect explains the overall higher *Fi* for System 2. If we consider only System 2 queries that did not use an asterisk filter (S2 no*), the overall *Fi* is nearly identical to System 1. This result is not surprising because System 2 returns the same documents as would be returned by System 1 for the same topical query.

Table 8.20 Fi and Fr for System 1 (S1) and System 2 (S2)

Document Ranks	**Fraction Indexed (Fi)**			**Fraction Relevant (Fr)**		
	S1	S2	S2 no *	S1	S2	S2 no *
1 – 10	0.58	0.74	0.61	0.089	0.130	0.104
11 – 20	0.48	0.58	0.48	0.045	0.065	0.067
21 – 30	0.41	0.47	0.40	0.010	0.025	0.024
1–100 (All)	0.32	0.41	0.34	0.022	0.039	0.029

Figure 8.12 shows the corresponding rates for all ten groups of ranks (up to document rank 100) expressed as ratios. The blue columns (on the left) show *Fi* for System 2 divided by *Fi* for System 1. The red columns (right) show *Fr* of System 2 divided by *Fr* of System 1. Although System 2 returned more indexed documents than System 1, the rate at which System 2 returned highly relevant documents exceeds what could be expected based solely on the higher rate of returning indexed documents. Plotting the same ratios of System 2 to System 1, but excluding queries with an asterisk in a semantic component box, results in a graph with a similar profile, as shown in Figure 8.13, but the S2/S1 ratios (all ranks) are 1.06 for *Fi* and 1.35 for *Fr* instead of 1.28 for *Fi* and 1.80 for *Fr*. We focus on the data for the first 30 ranks

because those ranks are of most interest to searchers and because so few highly relevant documents were returned by either system at the higher ranks. System 1 and System 2 returned 105 and 163 highly relevant documents at ranks 1-10, respectively, but only 3 and 10 highly relevant documents at ranks 51-60. Small changes in the number of relevant documents caused the seemingly erratic behavior of the ratio for *Fr* at higher ranks.

8.3. Discussion

8.3.1. Evaluation Perspectives: User versus System

We evaluated our experimental search systems from both a system-oriented perspective and a user-oriented perspective. Both perspectives are important for gauging the potential usefulness of a new approach to indexing and searching such as ours. The system perspective evaluation of a ranking algorithm is independent of whether a user recognized that a document might be useful and opened it. The system perspective is also independent of variations in how strictly users judged relevance (assigning graded relevance scores) or different opinions about what information satisfies a given scenario. Evaluations of ranking algorithms are important because adequate document ranking is necessary for a search system to provide value to the user. But a good algorithm is not sufficient if the user cannot extract value due to a poor interface or a difficult query language. The user perspective reflects the actual experience of a real user, which ultimately will determine the success of a system. Yet

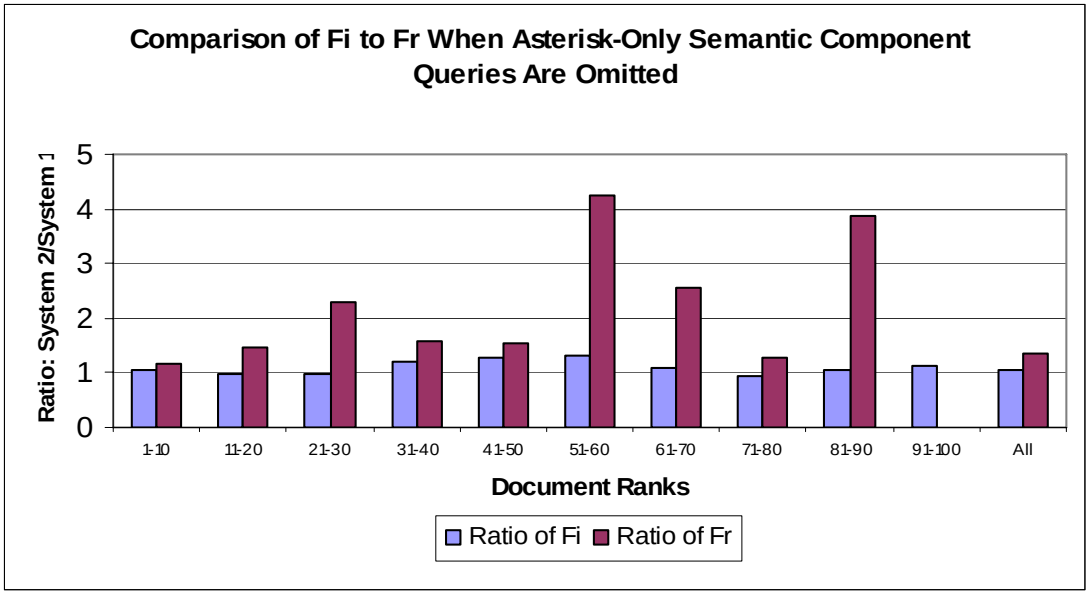


Figure 8.12 Fi and Fr expressed as a ratio System2/System1

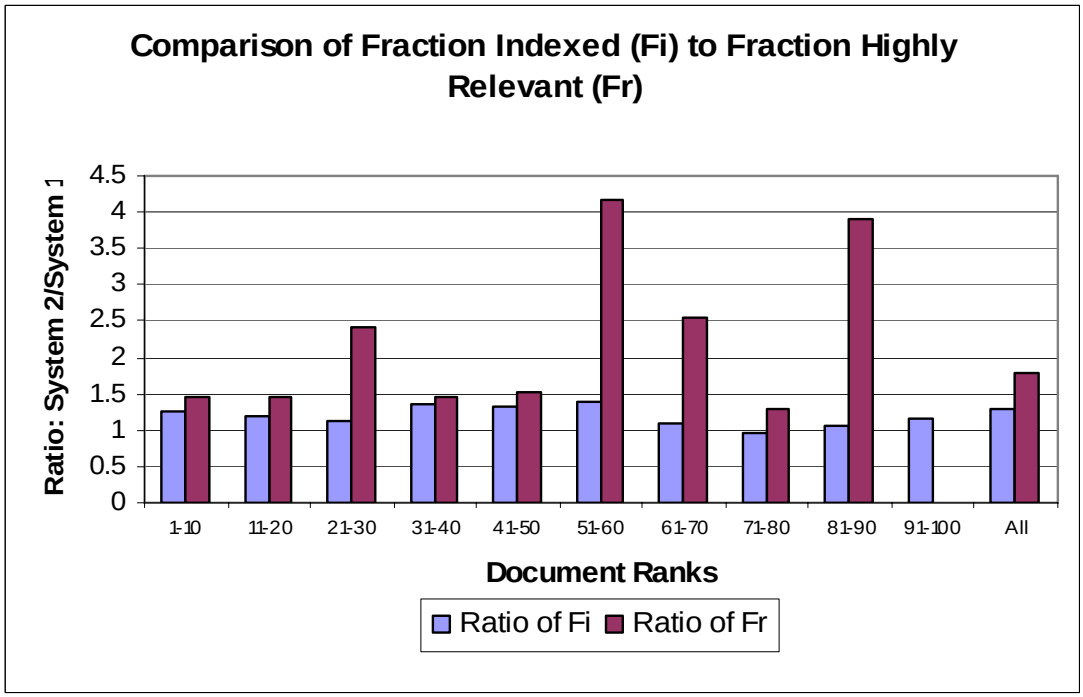


Figure 8.13 Fi and Fr expressed as a ratio of System 2/System 1. Semantic component queries with only an asterisk have been omitted

the user experience is affected by multiple factors, including the user interface, amount of training and experience with a new search system, accuracy of implicit relevance assessments based on the results list returned from a search, interpretation of the scenarios, and ease of understanding the documents.

Our evaluation from a system perspective was different from the usual test-collection approach. We used queries submitted by real users in an interactive setting, not the text describing the information needs, as is done in many test collection-based studies. It was our use of independent expert relevance judgments that makes this a system-oriented evaluation. We used graded relevance judgments in a common reference standard to calculate commonly-used IR evaluation metrics instead of individual user assessments for each query. Based on the system-oriented evaluation, we conclude that semantic components can be a valuable supplement to existing indexing and searching techniques. The addition of semantic components, as reflected in the use of System 2, consistently improved search performance as measured by MAP and nDCG for the best query in each session.

Our evaluation from the user perspective was based on the users' queries and individual user relevance judgments for those queries. Because the scenarios indicated a need for specific information, most users quit searching once they found a single highly relevant document. This gave us much less information to use in the evaluation. We used the rank of the single best user-relevant document per session because we did not want to penalize the searches by users who quit after finding one relevant document. We calculated reciprocal rank and discounted gain because other

metrics are less appropriate in the face of a single relevant document per search. From the user perspective, we did not find a significant difference between the two systems, although the mean scores for System 2 were somewhat better than for System 1. The lack of a significant effect on results achieved by the user is also reflected in the user questionnaires. User satisfaction is a global score that reflects all aspects of the user experience, including the user interface and possibly even the document collection, not just a ranking algorithm. The query interface for System 2 was more complex to use and required more thought than System1. If the user did not achieve better results, even if the underlying document ranking was better, then it is not surprising that user satisfaction was not higher for System 2. Our results highlight the importance of evaluating new systems from multiple perspectives.

We examined the reasons for the disparity between the user perspective evaluation and the system perspective evaluation in considerable detail. Of the 107 successful search sessions (sessions for which the user found at least one highly relevant or fairly relevant document) there were 16 instances, distributed over all four scenarios, in which the best query from a system perspective was different from the best query from the user perspective. We classified these instances as follows:

- In 12 cases, the system best query and user best query were distinct, and the system best query preceded the user best query. In each case, the system returned highly relevant documents early in the results list. The user either failed to notice the document, or determined that the document was not relevant based on the title and summary. We use the term *missed* document for

highly relevant documents from the reference standard that were ignored by the user. The first missed document appeared at ranks: 1 (4 times), 2 (3 times), and 3, 4, 5, 15, 47 (1 time each).

- In 2 cases, the system best query and the user best query were distinct, and the user best query preceded the system best query. Both cases involved differences in relevance judgments and one also involved missed documents. In one case, for Scenario B, the user identified a document as highly relevant that the reference standard labeled as marginally relevant. This was the user best query for the session. Despite having labeled the document as highly relevant, the user issued three more queries, suggesting he was not really satisfied with the document that he found. One of those queries returned the one highly relevant document in the reference standard at rank 15 (the system best query). In the second case, the user identified a fairly relevant document at rank 2 (the user best query), then issued another query and again found only a fairly relevant document, this time at rank 5. According to the reference standard, both queries returned highly relevant documents at rank 2 but the later query also returned additional highly relevant documents and therefore had a higher nDCG and was the system best query.
- In 2 cases, multiple queries tied for system best query (i.e., they had the same nDCG) and one of those queries, but not the first, was the user-best query. In both cases, multiple queries returned the same hit lists. Documents that were

missed in the first query were found in a later query (in which they appeared at the same rank as in the earlier query).

The phenomenon of missed documents was a striking finding in our experiment. Missed document were also responsible for seven unsuccessful search sessions. Why did the users not click on highly relevant documents that appeared high on a result list? We suggest three possibilities:

- *A known- item search that distracted the user from other documents*. Interviews with the participants revealed that, because of their familiarity with the documents, the searchers sometimes searched for a particular familiar document. This focus on a known document may have caused the user to ignore other documents, at least initially.
- *A legitimate relevance disagreement*. Our searchers were familiar with many of the documents in the collection. In some cases they may have had a strong and accurate sense of what was in a document and made a conscious, if implicit, relevance assessment when they ignored the document in a result list.
- *An inadequate or misleading indication of document contents*. The interface displayed the title, a human-authored summary, and an automatically generated snippet showing the query term in context for each document in the result list. The information displayed was the same as that in the operational system. In addition, System 2 also displayed semantic component information for the documents that had been indexed. Nevertheless, it appears that the information

in the results display might not have conveyed document contents adequately for accurate initial user relevance assessment.

We did not design this experiment to study relevance assessment so we do not have enough information to indicate how often each of these factors may have contributed to documents being missed. The specific nature of the information needs in the scenarios and the searchers' familiarity with the domain makes it unlikely that their understanding of the problem changed during the course of a search session. Certainly the differences in relevance judgments for Scenario B highlight the effect of differing relevance judgments on assessing system performance. A user interface that provides a better preview of returned documents could help prevent searchers overlooking relevant documents due to insufficient or misleading information about document contents.

The phenomenon of missed documents (and the factors that cause it) may help explain the failure of system performance improvements to have a significant effect on user performance, as has been documented in other studies [97-99, 157]. For example, Turpin and Hersh [98] analyzed user performance in interactive instance recall and question answering tasks. They noted that 30 – 50% of relevant documents returned in the top ten positions were not read by the searchers. Their research, like ours, did not address why searchers did not read the documents. Understanding why searchers fail to read relevant documents, and developing methods to make the potential relevance of documents more apparent in the search results, is an important area for additional research.

8.3.2. Evaluation Perspectives: Query versus Session

We also evaluated our system from another pair of perspectives: the single-query perspective and the search-session perspective. Test-collection–based studies typically formulate one query for each topic and compare systems based on mean performance over a set of topics. We were interested in knowing how well each system would perform given a "good" query. We therefore identified the query with the best performance in each search session in order to compare the potential performance of the two systems. However, because we are interested in supporting domain experts whose time for searching may be limited, the number of queries to complete a search is also important. We framed the users' task as finding the necessary information to make a clinical decision, so searchers entered queries and examined results until either finding the desired information or declaring the search a failure.

Overall, users spent more time using System 2 and entered more queries into System 2 than System 1. We saw some obvious reasons for query failure that occurred in both systems, including typos, mistakes in using the query syntax, and searches that were either too broad (thousands of hits) or too narrow (zero or few hits). Two possible explanations for the greater number of queries entered into System 2 are:

- A larger number of queries that returned no hits, 34 in System 2 as compared to 20 in System 1. Twenty-four of the 34 queries (70.6%) with no hits in System 2 involved use of the asterisk operator in semantic component queries, which acted as a filter. Although the asterisk would match any text in the

designated semantic component, only documents that matched the topical query and that contained an instance of the requested semantic component were returned. In retrospect, using the asterisk to boost document ranking, instead of as a filter, might have improved session-based search performance with System 2.

- There was likely a need to learn how to use a new system. Although the training session allowed participants to use both systems, they had little time to experiment and discover how to use semantic components to best advantage. In addition to the expected learning curve, because semantic components were new to the searchers, we believe the searchers also engaged in some experimentation with the search interfaces during the experiment, especially with System 2.

Semantic components per se seem unlikely to have caused a substantial increase in number of queries. The topical queries entered into both systems for a given scenario were quite similar. Addition of semantic component query terms, other than the asterisk, resulted in re-ranking of the result set but not omission of documents. Although re-ranking resulted in both increasing and decreasing the rank of relevant documents, the mean change was an improvement in the rank of relevant documents by 8.1 places.

Session-based discounting allowed us to explore the effect of query sequencing in more detail. Despite the larger number of queries issued with System 2, the session-based gain was not less, from either the user or the system perspective. Improved

document ranking in the best queries, once issued, largely countered the effect of query sequence discounting although the difference between the two systems was not statistically significant when session-based discounting was applied. We used a logarithm base of 2, which simulates an impatient user by aggressively discounting gain for both document rank and query iteration. We did not test whether a different combination of discounting parameters changes the results.

When we concatenated the top ten results from each query in the session, simulating what a user would see if he looked only at the first ten documents returned by each query, the resulting mean curve for System 2 appears substantially better than that for System 1 only after about the third query. The data from both the best query and concatenated session analyses suggest that most of the difference between systems was attributable to a markedly better performance by System 2 than by System 1 for Scenario A.

Concatenating the top ten documents places the emphasis on ranking quality at the top of a result list, but is only a rough approximation of the user experience. Individual users may look at more, or fewer, documents and may vary how far down a list they look based on the documents that appear earlier in the list. The relative cost to a user for a query that returns fewer than ten documents (or whatever threshold is chosen) is unknown. We believe that a session based approach to interactive user studies is an important area for future IR research.

8.3.3. Effect of Partial Indexing on Study Results

We successfully predicted and indexed all the highly relevant documents in the reference standard, although there may be highly relevant documents in the collection that we did not discover. We also predicted and indexed a reasonable selection of nonrelevant documents likely to be returned by the queries in the study. Over half of the highest-ranked documents returned by either system had been indexed with semantic components. Given that System 1 ignores semantic component indexing and that there were few relevant documents, the high frequency of semantic component indexing among retrieved documents means that we indexed a high proportion of the nonrelevant documents likely to compete with relevant documents for retrieval. Indexing competing, but nonrelevant, documents minimized any bias towards retrieval of relevant documents due only to the presence of semantic component indexing. Comparing the ratio of the two systems for *Fi* and *Fr* shows that System 2 returned highly relevant documents at a rate higher than would be expected based on the indexing rate alone. Our analysis indicates that the improved performance of System 2 cannot be attributed to our selective indexing.

8.3.4. Limitations of the Searching Study

We recognize, of course, that this study had limitations. The experiment was limited to a single user group searching over a single collection of documents in a single domain. The number of search scenarios was quite small, especially compared to the number of topics typical of TREC, but unlike laboratory-style evaluations we

had 30 domain experts as end-users who formulated queries and interacted with the system, resulting in 120 search sessions. As intended, the scenarios varied in difficulty. Overall, scenario was a stronger determinant of search performance than search system, but two-way analysis of variance indicated that search system and scenario did not interact and allowed us to examine their effects separately. It is encouraging that System 2 generally performed well for all 4 scenarios. This study is the first empirical study to evaluate semantic components; establishing generalizability will require more research, but we believe our current results warrant further investigation into the potential usefulness of this model.

8.4. Summary

In this chapter we presented an interactive searching study in which thirty domain experts searched for documents to satisfy four realistic searching scenarios. Each searcher used a baseline searching system with full-text and keyword indexing for two scenarios and an experimental searching system with semantic components in addition to full text and keyword indexing for the other two scenarios.

We analyzed the searching results from both a system-oriented perspective and a user-oriented perspective. From the system-oriented perspective, when they used the experimental system with semantic components the searchers attained results with better document ranking, as determined by a reference standard of relevance judgments. From the user-perspective, the searchers entered more queries and spent more time searching when they used semantic components. Although the searchers

found relevant documents at somewhat better rankings when using the system with semantic components, based on their own relevance judgments, the difference between systems was not statistically significant. We discussed several reasons why the search results were significantly better from the system perspective but not from the user perspective. We also used a session-based metric, sDCG, for evaluating search results in our multiple-query search sessions. By discounting the gain value of relevant documents returned by later queries, sDCG facilitates comparing search systems in interactive settings where users can refine their queries in addition to scanning further down a results list to find relevant documents. When we applied session-based discounting to our results, the system with semantic components performed somewhat better than the baseline system, but the results were not statistically significant.

Chapter 9 Conclusions and Future Work

In this dissertation, we introduce the *semantic components* model for supplementing traditional document indexing techniques (such as full text and keyword indexing) used in information retrieval systems. The semantic components model uses domain-specific and collection-specific concepts and relationships to introduce additional information about the semantic content of documents into the process of matching documents to information needs. A semantic component schema consists of a set of document classes that describe logical groupings of documents within a document collection and a set of semantic components for each class. Document classes can be based on the types of domain-specific topics addressed by the documents or on the purpose and logical structure of the documents. Semantic components represent the types of information that are common in the document, and that contain content likely to be searched by users. We have so far considered three ways that semantic components can be useful:

- searching for query terms within specific semantic components
- indicating a preference for documents containing particular semantic components
- displaying a list of the semantic components present in the document, and their sizes, for each document in the search results

We also imagine that an IR system using semantic components could employ user interface enhancements to help searchers as well. For example, when a searcher clicks

on a link in a search result, the system could display the corresponding document scrolled to the beginning of a semantic component instance targeted by the search, with the semantic component instance highlighted, in addition to highlighting matching terms in the query. Or, a user interface could display the semantic component instance that matches the query as a series of excerpts. We have not yet implemented or studied such user interface enhancements because we first wanted to investigate the feasibility of semantic component indexing and the potential benefit of using semantic components for searching.

9.1. Findings and Contributions

The main findings of the research relate to four original questions posed in the first chapter.

1. *Can useful document classes and semantic components be identified for particular domain-specific document collections?*

We showed that we were able to identify document classes and semantic components in three document collections from two different domains, medicine and public land management. We described our use of two different methods for developing semantic component schemas. The first method uses a bottom-up approach, analyzing documents sampled from the document collection to determine the types of documents and types of information present in the collection. The second method uses a top-down, domain-centered approach, identifying existing document types, templates, or common document structures. We also discussed using

knowledge about the domain, the users, and common work tasks to refine the schemas. We discovered that the names used for document classes and semantic components can have a substantial effect on how users interpret the schema. Factors that can facilitate schema development include homogeneity of a document collection and pre-existing structures, such as document types, templates, or instructions regarding document preparation. Knowledge about the user community, such as common information needs and types of work tasks that motivate their searches, can also contribute to schema development.

2. *Can searchers express information needs using document classes and semantic components?*

We addressed this question by considering whether information needs in a particular domain can be expressed using the elements of semantic component schemas. We developed semantic component schemas for two medical document collections by analyzing the types of information present in documents sampled from the collections. We used an existing taxonomy of generic questions, derived from real questions posed by family practice physicians in the course of caring for patients, as a source of information needs. We then mapped generic questions from the taxonomy categories to the two semantic component schemas. We were able to map 68% of the question categories to the semantic component schema for one document collection and 72% of the categories to the schema for the other collection. For both collections, our mappings represented over 92% of the original 1396 questions, as indicated by the frequency data for each category. We discussed the types of

questions we did not map to the collections using the schemas. Only two generic questions, representing 0.4 % of the questions, were general "aboutness" questions ("What is condition x?" and "What is test x?") that did not suggest a semantic component to which we could map. Finally, we noted that the schemas themselves could be useful to searchers by profiling the types of information available from a given document collection. Although a schema cannot indicate whether a particular question can be answered, it can indicate the kinds of questions that the collection is likely to be able to answer, or not be able to answer.

Our searching study also demonstrated that searchers could use semantic components when searching for answers to questions posed by realistic scenarios. The participants' searching behavior, their questions and their practice searches in a training session, and their responses to survey questions indicated that they understood how and why to use semantic components for searching.

3. How easily can semantic components be identified in documents?

We compared manual semantic component indexing to manual keyword indexing in a study with sixteen participants who index documents for a Danish health portal, sundhed.dk. Each participant was assigned twelve documents to index, six using keywords and six using semantic components. Although not directly comparable because the units of measurement are different, we found that semantic component indexing had moderate accuracy (compared to a reference standard) and consistency (comparing the indexers among themselves) whereas keyword indexing had low accuracy and consistency. When using keywords, indexers in our study were more

likely to agree, with each other and with the reference standard, on the exclusion of all terms from one or more of the controlled vocabularies than to agree on the inclusion of a particular term. Semantic component indexing might be especially useful for documents about topics that are not well covered by existing controlled vocabularies.

We also compared the time required to index documents using the two indexing techniques. Semantic component indexing took slightly longer than keyword indexing, approximately 7 and 6 minutes per document, respectively. Both types of indexing were recorded on paper. We also recorded the time to index 371 documents with semantic components for the searching study using a prototype indexing application. The mean time was only 3.5 minutes per document.

We used questionnaires to study indexer attitudes about semantic component indexing. The responses regarding perceived difficulty and confidence in their choices were similar for the two types of indexing. Slightly more indexers indicated a preference for performing keyword indexing but slightly more indexers thought semantic component indexing would be more useful for searching.

4. Are semantic components useful for retrieving documents?

We found that semantic components can enhance information retrieval by producing better document rankings. Thirty physicians searched interactively for documents containing information to satisfy four realistic scenarios that required information to support decisions about patient care. Each physician searched two scenarios with a baseline search system and searched two scenarios with an experimental search system that incorporated semantic components but was otherwise

identical to the baseline system. The participants rated the relevance of the documents they found on a four point scale. Because it was an interactive study, the searchers could issue multiple queries until they were satisfied with the information found or until they reached a point when they would quit searching in a real life setting. We evaluated semantic component searching from two different perspectives, a system-oriented perspective and a user-oriented perspective.

For the system perspective, we evaluated query results using a reference standard of relevance judgments made by a physician researcher. In one evaluation, we chose the query with the best results in each search session, where a session was the sequence of queries posed by one searcher for one scenario. In most cases, the best query was the last query because searchers stopped searching after finding useful information. When we compared search performance for the best queries from each session, the experimental system with semantic components performed significantly better than the baseline system. We also compared the session-based performance of the two systems, evaluating query results using a new metric that discounts the value of relevant documents that are returned by later queries compared to earlier queries. Using the session-based discounting metric, the system with semantic components performed somewhat better than the baseline system but the difference was not statistically significant.

For the user perspective, we evaluated query results using the individual searcher's own relevance judgments. In most sessions, a searcher found only one or two documents he judged highly relevant. Sometimes the searchers found no highly

relevant or fairly relevant documents. Using the searchers' relevance judgments, the performance of the system with semantic components was slightly better than the baseline system, but the results were not statistically significant. More search sessions were successful (the searcher found at least one document that he judged to be either highly relevant or fairly relevant) when using the experimental system than when using the baseline system. However, search sessions lasted longer and were comprised of more queries with the experimental system than with the baseline system. Dissatisfaction with search results, measured with a questionnaire following each search session, was slightly higher for the system with semantic components, but the difference was not statistically significant.

In addition to the findings summarized above, we made the following contributions:

- We provided a formal description of the semantic components model.
- We described a prototype implementation of semantic component indexing software.
- We analyzed semantic component and keyword indexing, evaluated candidate metrics, and proposed methods for evaluating each type of indexing.
- We discovered a weakness in a metric used to assess consistency of unitizing (deciding the extent of text that should be annotated with a given category name) in content analysis.
- We implemented semantic components in a prototype search system built on top of an existing search engine.

Our research also led to recognizing two important issues. First is the need for a method of comparing search results from interactive search sessions consisting of multiple queries for a single information need. We addressed this need by collaborating with Dr. Kalervo Jarvelin to develop a session-based metric for evaluating ranked results in multiple query sessions (sDCG) [44]. We used the new metric as one of our evaluation techniques in Chapter 8. The second issue is evaluating new indexing techniques in large document collections. If a document collection is too small, retrieval results may not be generalizable to larger collections. But manual indexing is too expensive to implement in large collections without knowing whether it will be useful. We relied on predicting a large proportion of the documents that would compete with relevant documents for retrieval and then retrospectively analyzing our results to show that selective indexing did not bias our study results. This method was effective because our search system reranked documents that would have been returned by the baseline system and because there were only a few relevant documents for each scenario. However, a more robust and generalizable method for evaluating new types of indexing would be useful to support any future research on new indexing techniques.

9.2. Implications and Limitations of the Research

We have provided evidence of the feasibility and potential usefulness of semantic components for searching domain-specific libraries. Our findings suggest that exploring the introduction of semantic components into operational search systems is

warranted. However, this research has several limitations that affect how such explorations should be pursued.

First, our work was limited to two domains, and most of the work was performed in a single domain, medicine. Evidence of the usefulness of semantic components in other domains is necessary before committing human and economic resources to large-scale implementations of semantic components.

Second, we have provided only preliminary evidence for the potential usefulness of semantic components. Although our searching study demonstrated that semantic components can enhance search results, we conducted a single study, in a single document collection, with a single group of users, for only four scenarios. Similarly, we studied indexing in a small sample of documents, from a single document collection, with a single group of indexers. These limitations can be addressed with additional work, experimenting with semantic component searching and indexing with different user groups and different document collections.

Third, there are variations on the semantic components model that we have not yet explored. In our experiments, we allowed each document to belong to only one document class although the model does not preclude documents belonging to multiple classes. In Chapter 4 we offered an example of a document indexed for the searching study that could usefully be indexed as belonging to two of the classes in our schema. In addition, a given document might be useful for multiple tasks, by multiple user groups. Different target audiences might have different perspectives and find different semantic components to be useful. We conjecture that the semantic

component model might be more effective if schemas are tailored to the needs of different user groups (such as physicians versus patients using medical document collections) or to different types of tasks (such as tasks related to research versus clinical care). Document collections that support multiple user groups might benefit from several semantic component schemas to reflect the interests, needs, and vocabularies of different user groups. We have also considered, but have not implemented, variations on the semantic component model. Three examples are: (1) a flat schema that has only semantic components and no document classes, (2) a mixed schema that has some document classes with associated semantic components and some classes for which class membership is the only additional information in the indexing, and (3) a schema that allows a semantic component to occur in multiple document classes and to be searched across all classes that contain the semantic component.

Fourth, we have only begun to address the issue of scalability. Even if additional research supports our conclusion that manual semantic component indexing is as feasible as manual keyword indexing, the resources required for manual indexing will prevent adoption of semantic component indexing in many settings. Finding ways either to automate semantic component indexing or to improve its scalability in other ways may be a prerequisite to widespread use.

Finally, our work was motivated by domain experts when they are searching domain-specific libraries for targeted information needs. We have not investigated whether semantic components might also be applicable in other settings where some

of the same limitations to the effectiveness of current search algorithms apply. For example, enterprise information systems might contain predictable document types and information types that are enterprise-specific instead of domain-specific. Semantic components that reflect those information types might be useful for searching. Personal document collections might also benefit from semantic components using a user-specific schema.

9.3. Future Work

Each of the limitations mentioned in the preceding section invites future research to address the limitations of the current work. In this section we briefly elaborate on three areas of future work that we find particularly compelling: (1) enhancing the scalability of semantic component indexing by allowing incremental user indexing, (2) extending the semantic components approach using variations on the current model, and (3) automating semantic component indexing to improve scalability.

9.3.1. Incremental End-User Indexing

We propose to explore end-user indexing for two reasons. First, if much of the time and effort of indexing is attributable to reading and understanding the document, then selecting and labeling semantic component instances will take relatively little additional effort, assuming that the indexing tools are easy and relatively seamless to use. Presumably the user of a document has already committed time to reading and understanding the document. User indexing would facilitate re-use of documents by

the indexer and leverage effort already expended for the benefit of other members of a user community. Second, the phenomenon of collaborative tagging suggests that (some) users will volunteer time and effort to categorize resources. Already, document creators and users can assign descriptors, commonly referred to as tags, to describe a variety of electronic resources such as web pages,[27] bibliographic references,[28] and photo collections.[29] Implementing semantic component indexing by end-users extends the notion of collaborative tagging to associating tags with subdocuments and not just whole documents.

At first glance, delegating indexing to users may seem risky because user indexing is unpredictable and uncontrolled. Yet, while one instance of indexing may be unreliable, the accumulation of multiple indexing instances is likely to converge toward a meaningful result and may, on average, be better than indexing produced by a single individual. Studies of del.icio.us provide evidence that tagging by a critical mass of users results in convergence to stable tag usage patterns [158, 159]. Furthermore, semantic components supplement traditional indexing and search, allowing more precise search specification. Poor user indexing will inhibit the ability of semantic components to improve search precision, but is unlikely to degrade retrieval quality compared to traditional whole document search alone.

---

[27] http://del.icio.us
[28] http://www.citeulike.org/
[29] http://flickr.com

Incremental end-user indexing implies that a given document can have zero, one, or many instances of semantic component indexing. An important piece of this work will be investigating how to implement a system that allows such flexibility. Two important issues are (1) how to combine the data from multiple indexing instances in a ranking algorithm, when the instances can provide conflicting evidence about the relevance of a given document to a given query; and (2) how to implement a scalable system that can capture, store, and search multiple indexing instances per document.

9.3.2. Variations on the Semantic Components Model

We have already mentioned the possibility of allowing a document to be indexed as a member of multiple document classes and of allowing multiple indexing instances per document. Here we reflect on varying the model in three ways: (1) loosening the hierarchy so that not every document class must have semantic components, (2) eliminating the hierarchical structure of document classes and semantic components, and (3) eliminating the pre-defined schema.

In some document classes we identified semantic components whose potential value seemed obvious, such as the *treatment* and *referral*. In other cases, we conjectured that most, if not all, the benefit of semantic components could be gained by directing the search at documents belonging to a particular class without specifying a particular semantic component. For example, specifying that a search was for information about *services* or about a *clinical unit* might be sufficient additional information in a query to improve a search.

We also want to investigate a simpler version of the semantic components model, the flat schema. In this version, all documents belong to the same universal class, so only semantic components are indexed. Document classes provide some additional structure for representing semantic information in documents but they also add to complexity. Feedback from both indexers and searchers suggests that they prefer the schema to be simple. A flat schema might decrease the cognitive load of formulating queries and speed query entry into search forms. Also, some information types might imply a particular document class, and thus eliminate the need for a separate document classification. Other information types might appear in multiple document classes, such as *referral* information in sundhed.dk documents. A flat schema will be easier to scale for large numbers of indexing instances. We want to investigate whether a flat schema is more or less effective than a two level hierarchical schema, especially when trying to represent documents from multiple user perspectives.

Developing a schema requires time and intellectual effort that must be repeated for each document collection. Furthermore, successful use of semantic component indexing requires indexers and searchers to have a shared understanding of document class names and semantic component names. We would like to test an open schema (a flat schema with no predefined semantic components) that would allow an indexer to associate a segment of text with any name deemed appropriate. The open-schema approach resembles collaborative tagging except that a tag is bound to a whole document whereas a semantic component name is bound to a selected subdocument. The open schema retains the essence of the semantic component approach, which

extends whole document search by also searching subdocuments, where subdocuments are defined on a semantic, not structural, basis. Such an approach would:

- eliminate the "schema first" requirement
- decrease the need for a pre-existing shared understanding of labels
- provide increased flexibility for indexing more heterogeneous collections
- allow document representations to evolve as a collection changes or as the state of knowledge in a domain changes

However, an open schema also raises additional questions about how to combine information from multiple indexing instances in the presence of synonyms, word variants, or differing levels of specificity.

9.3.3. Automated Semantic Component Indexing

In Chapters 2 and 6 we discussed a variety of text analysis tasks that identify and manipulate subdocuments. We anticipate that successful efforts to automate semantic component indexing will build on existing techniques developed to perform similar types of text analysis. One promising approach is to draw on manual indexing examples to identify features of text in particular semantic components instances and then use existing technologies to identify the features in text. Such technologies might range from using regular expressions, to identify phrases that signal the presence of particular types of information, to using state-of-the-art natural language processing (NLP) tools to identify syntactic and semantic clues. NLP tools that are tuned to

particular domains, including medicine, already exist and could be useful. Machine learning techniques could then exploit the identified features, learning to combine evidence from multiple features in optimal ways.

It will not be easy to produce automated semantic component indexing that is equivalent to manual semantic component indexing. On the other hand, replicating manual indexing may not be necessary to achieve substantial benefit. We do not know what level of quality and indexing granularity is necessary to enhance conventional full text or keyword indexing. Additional research that addresses the tradeoffs between automation and quality might reveal particular characteristics of semantic component indexing that provide benefit and yet can be automated more easily than trying to imitate manual indexing.

References

1. The American Heritage Dictionary of the English Language, Fourth Edition. URL: http: //dictionary.reference.com/browse/domain, Accessed: January 15, 2008.

2. Marcos André Gonçalves, Edward A. Fox, Layne T. Watson, and Neill A. Kipp. Streams, structures, spaces, scenarios, societies (5S): A formal model for digital libraries. *ACM Transactions on Information Systems*, 22(2), pp. 270-312, 2004.

3. Ronald Fagin, Ravi Kumar, Kevin S. McCurley, Jasmine Novak, D. Sivakumar, John A. Tomlin, and David P. Williamson. Searching the workplace web. In *Proceedings of the 12th International Conference on World Wide Web (WWW2003)*, pp. 366-375, Budapest, Hungary, May 20-24, 2003.

4. Jon Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5), pp. 604-632, 1999.

5. Luanne Freund, Elaine G. Toms, and Julie Waterhouse. Modeling the information behaviour of software engineers using a work-task framework. In *ASIS&T '05 Proceedings of the 68th Annual Meeting*, Charlotte, NC, October 28-Nov2, 2005.

6. Paul N. Gorman. Information needs of physicians. *Journal of the American Society for Information Science*, 46, pp. 729-736, 1995.

7. Paul N. Gorman and Mark Helfand. Information seeking in primary care: How physicians choose which clinical questions to pursue and which to leave unanswered. *Medical Decision Making*, 15, pp. 113-119, 1995.

8. John W. Ely, Jerome A. Osheroff, Mark H. Ebell, George R. Bergus, Barcey T. Levy, M. Lee Chambliss, and Eric R. Evans. Analysis of questions asked by family doctors regarding patient care. *BMJ*, 319, pp. 358-361, 1999.

9. John W. Ely, Jerome A. Osheroff, Paul N. Gorman, Mark H. Ebell, M. Lee Chambliss, Eric A. Pifer, and P. Zoe Stavri. A taxonomy of generic clinical questions: classification study. *BMJ*, 321(7258), pp. 429-432, 2000.

10. Linda Schamber, Michael B. Eisenberg, and Michael S. Nilan. A re-examination of relevance: Toward a dynamic, situational definition. *Information Processing & Management*, 26(6), pp. 755-776, 1990.

11. D. G. Covell, G. C. Uman, and P. R. Manning. Information needs in office practice: Are they being met? *Ann Intern Med*, 103, pp. 596-599, 1985.

12. Shawn P. Curley, Donald P. Connelly, and Eugene C. Rich. Physicians' use of medical knowledge resources: Preliminary theoretical framework and findings. *Medical Decision Making*, 10, pp. 231-241, 1990.

13. N. J. Belkin, R. N. Oddy, and H. M. Brooks. ASK for information retrieval: Part 1. Background and theory. In *Journal of Documentation*, 38, pp. 61-71. 1982. Reprinted in Karen Sparck Jones and Peter Willett, Editors. *Readings in Information Retrieval*. Morgan Kaufmann Publishers, Inc., San Francisco, CA, pp. 299-304, 1997.

14. A. Dillon. Reader's models of text structures: the case of academic articles. *International Journal of Man-Machine Studies*, 35(913-925) 1991.

15. A. P. Bishop. Document structure and digital libraries: How researchers mobilize information in journal articles. *Information Processing & Management*, 35, pp. 225-279, 1999.

16. M. L. Nielsen, L. Delcambre, T. Tolle, and M. Weaver. Indexing and retrieval challenges in digital government systems – summary of an empirical research project. In *2nd Scandinavian Workshop on eGov*, Copenhagen, Denmark, 2005.

17. Paul Gorman, Mary Lavelle, Lois Delcambre, and David Maier. Following experts at work in their own information spaces: Using observational methods to develop tools for the digital library. *Journal of the American Society for Information Science and Technology*, 53(14), pp. 1245-1250, 2002.

18. sundhed.dk. URL: http: //www. sundhed.dk, Accessed: December 6, 2007.

19. Shawn Bowers, Lois Delcambre, and David Maier. Superimposed schematics: Introducing E-R structure for in-situ information selections. S. Spaccapietra, S. T. March, and Y. Kambayashi, Editors, *ER 2002*, Lecture Notes in Computer Science 2503, Berlin: Springer-Verlag, pp. 90-104, 2002.

20. Mathew Weaver, Lois M. L. Delcambre, Marianne Lykke Nielsen, Susan L. Price, David Maier, and Timothy Tolle. Supporting domain-specific digital libraries in government: Two case studies. In Hsinchun Chen, et al., Editors, *Digital Government: Advanced Research and Case Studies.* Springer, New York, 2008, forthcoming.

21. J. Rowley. The controlled versus natural indexing languages debate revisited: a perspective on information retrieval practice and research. *Journal of Information Science*, 20(2), pp. 108-119, 1994.

22. F. W. Lancaster. *Indexing and Abstracting in Theory and Practice*. Third ed. University of Illinois Graduate School of Library and Information Science: Champaign, IL, 2003.

23. Jens-Erik Mai. Analysis in indexing: document and domain centered approaches. *Information Processing & Management*, 41, pp. 599-611, 2005.

24. Marcia J. Bates. Indexing and access for digital libraries and the Internet: Human, database, and domain factors. *Journal of the American Society for Information Science*, 49(13), pp. 1185-1205, 1998.

25. J. D. Anderson and J. Pérez-Carballo. The nature of indexing: How humans and machines analyze messages and texts for retrieval. Part I: Research, and the nature of human indexing. *Information Processing & Management*, 37, pp. 231-254, 2001.

26. J. D. Anderson and J. Pérez-Carballo. The nature of indexing: How humans and machines analyze messages and texts for retrieval. Part II. Machine indexing, and the allocation of human versus machine effort. *Information Processing & Management*, 37, pp. 255-277, 2001.

27. J. Savoy. Bibliographic database access using free-text and controlled vocabulary: an evaluation. *Information Processing & Management*, 41, pp. 873-890, 2004.

28. Mark E. Funk and Carolyn Anne Reid. Indexing consistency in MEDLINE. *Bulletin of the Medical Library Association*, 71(2), pp. 176-183, 1983.

29. M. F. Porter. An algorithm for suffix stripping. *Program,* 14, pp. 130-137, 1980. Reprinted in Karen Sparck Jones and Peter Willett, Editors, *Readings in Information Retrieval*. Morgan Kaufmann Publishers, Inc., San Francisco, CA, pp. 313-316, 1997.

30. Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management,* 24(5), pp. 513-523, 1988. Reprinted in Karen Sparck Jones and Peter Willett, Editors, *Readings in Information Retrieval*. Morgan Kaufmann Publishers, Inc., San Francisco, CA, pp. 323-327, 1997.

31. S. E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3), pp. 129-146, 1976.

32. K. Sparck Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and status. *University of Cambridge Computer Laboratory Technical Report 446*, 1998.

33. Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of SIGIR 1998*, pp. 275-281, Melbourne, Australia.

34. Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International Conference on World Wide Web 7*, pp. 107-117, Brisbane, Australia, 1998.

35. David Hawking and Justin Zobel. Does topic metadata help with web search? *Journal of the American Society for Information Science and Technology*, 58(5), pp. 613-628, 2007.

36. Cyril Cleverdon. The Cranfield tests on index language devices. *Aslib Proceedings,* 19, pp. 173-192, 1967. Reprinted in Karen Sparck Jones and Peter Willett, Editors, *Readings in Information Retrieval*. Morgan Kaufmann Publishers, Inc., San Francisco, CA, pp. 47-59, 1997.

37. Ellen M. Voorhees. The philosophy of information retrieval evaluation. C. A. Peters, et al., Editors, *CLEF 2001*, Lecture Notes in Computer Science 2406, Berlin: Springer-Verlag, pp. 355-370, 2002.

38. National Institute of Standards and Technology (NIST). Text REtrieval Conference (TREC) Home Page. URL: http: //trec.nist.gov/, Accessed: December 31, 2007.

39. Ellen M. Voorhees. Overview of TREC 2006. *NIST Special Publication: SP 500-272, The Fifteenth Text REtrieval Conference (TREC 2006) Proceedings*, 2006.

40. Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. ACM Press: New York, NY, 1999.

41. Chris Buckley and Ellen M. Voorhees. Evaluating evaluation measure stability. In *Proceedings of SIGIR 2000*, pp. 33-40, Athens, Greece.

42. Chris Buckley and Ellen M. Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of SIGIR 2004*, Sheffield, South Yorkshire, UK, July 25-29, 2004.

43. Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4), pp. 422-446, 2002.

44. Kalervo Järvelin, Susan Price, Lois Delcambre, and Marianne Lykke Nielsen. Discounted cumulative gain based evaluation of multiple-query IR sessions. In *Proceedings of the 30th European Conference on Information Retrieval (ECIR 08)*, Glasgow, Scotland, March 2008.

45. Eero Sormunen. Liberal relevance criteria of TREC - Counting on negligible documents? In *Proceedings of SIGIR 2002*, Tampere, Finland, August 11-15, 2002.

46. Tefko Saracevic. Evaluation of evaluation in information retrieval. In *Proceedings of SIGIR 1995*, pp. 138-146, Seattle, WA, 1995.

47. Pia Borlund. The IIR evaluation model: A framework for evaluation of interactive information retrieval systems. *Information Research*, 8(3) 2003.

48. Kalervo Järvelin. An analysis of two approaches in information retrieval: From frameworks to study designs. *Journal of the American Society for Information Science and Technology*, 58(7), pp. 971-986, 2007.

49. Peter Ingwersen and Kalervo Järvelin. *The Turn: Integration of Information Seeking and Retrieval in Context*. Springer: Dordrecht, The Netherlands, 2005.

50. Jane Reid, Mounia Lalmas, Karen Finesilver, and Morten Hertzum. Best entry points for structured document retrieval–Part I: Characteristics. *Information Processing & Management*, 42, pp. 74-88, 2006.

51. Jane Reid, Mounia Lalmas, Karen Finesilver, and Morten Hertzum. Best entry points for structured document retrieval–Part II: Types, usage and effectiveness. *Information Processing & Management*, 42, pp. 89-105, 2006.

52. Klaus Krippendorff. *Content analysis: An introduction to its methodology*. Second ed. Sage Publications: Thousand Oaks, CA, 2004.

53. Jay M. Ponte and W. Bruce Croft. Text segmentation by topic. *Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries*, Lecture Notes in Computer Science 1324, pp. 113-125, 1997.

54. James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. Topic Detection and Tracking Pilot Study Final Report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pp. 194-218, Lansdowne, VA, February 8-11, 1998.

55. M. A. Hearst and C. Plaunt. Subtopic structuring for full length document access. In *Proceedings of SIGIR 1993*, pp. 59-68, Pittsburgh, PA, 1993.

56. Marti A. Hearst. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1), pp. 33-64, 1997.

57. Min-Yen Kan, Judith L. Klavans, and Kathleen R. McKeown. Linear segmentation and segment significance. In *Proceedings of the Sixth Workshop on Very Large Corpora*, pp. 197-205, Montreal, Canada.

58. Xianoyong Liu and W. Bruce Croft. Passage retrieval based on language models. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management (CIKM '02)*, pp. 375-382, McLean, VA, November 4-9, 2002.

59. Ian Soboroff. Overview of the TREC 2004 novelty track. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC 2004)*.

60. Claire Cardie. Empirical methods in information extraction. *AI Magazine*, 18(4), pp. 65-79, 1997.

61. Kevin Crowston and Barbara H. Kwasnik. Can document-genre metadata improve information access to large digital collections? *Library Trends*, 52(2), 2003.

62. Andreas Rauber and Alexander Müller-Kögler. Integrating automatic genre analysis into digital libraries. In *Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '01)*, Roanoke, VA, 2001.

63. Luanne Freund, Elaine G. Toms, and Charles L. A. Clarke. Modeling task-genre relationships for IR in the workplace. In *Proceedings of SIGIR 2005*, Salvador, Brazil, August 2005.

64. Wanda J. Orlikowski and JoAnn Yates. Genre repertoire: The structuring of communicative practices in organizations. *Administrative Science Quarterly*, 39, pp. 541-574, 1994.

65. Kevin Crowston and Marie Williams. Reproduced and emergent genres of communication on the World-Wide Web. *The Information Society*, 16(3), pp. 201-215, 2000.

66. Elizabeth Sugar Boese and Adele E. Howe. Effects of web document evolution on genre classification. In *Proceedings of the Conference on Information and Knowledge Management (CIKM '05)*, pp. 632 - 639, Bremen, Germany, November 2005.

67. Brett Kessler, Geoffrey Numberg, and Hinrich Schütze. Automatic detection of text genre. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pp. 32-38, Madrid, Spain, 1997.

68. Anne M. Turner, Elizabeth D. Liddy, Jana Bradley, and Joyce A. Wheatley. Modeling public health interventions for improved access to the gray literature. *Journal of the Medical Library Association*, 93(4), pp. 487-494, 2005.

69. National Library of Medicine. ClinicalQuestions Collection. URL: http://clinques.nlm.nih.gov, Accessed: January 1, 2008.

70. F. W. Lancaster. MEDLARS: Report on the evaluation of its operating efficiency. *American Documentation,* 20, pp. 119-142, 1969. Reprinted in Karen Sparck Jones and Peter Willett, Editors, *Readings in Information Retrieval*. Morgan Kaufmann Publishers, Inc., San Francisco, CA, pp. 223-246, 1997.

71. National Library of Medicine. Medical Subject Headings Home Page. URL: http://www.nlm.nih.gov/mesh/, Last updated: October 30, 2007, Accessed: January 2, 2008.

72. National Library of Medicine. Topical subheadings with scope notes and allowable categories. URL: www.nlm.nih.gov/mesh/topsubscope2004.html, Last updated: October 28, 2003, Accessed: January 2, 2008.

73. Susan L. Price and Lois M. Delcambre. Using concept relations to improve ranking in information retrieval. In *Proceedings of the AMIA 2005 Annual Fall Symposium*, Washington DC, 2005.

74. Christopher S.G. Khoo and Sung Hyon Myaeng. Identifying semantic relations in text for information retrieval and information extraction. In Rebecca Green, Carol A. Bean, and Sung Hyon Myaeng, Editors, *The Semantics of Relationships: An Interdisciplinary Perspective.* Kluwer Academic Publishers, Boston, MA, pp. 161-180, 2002.

75. Christopher S.G. Khoo, Sung Hyon Myaeng, and Robert N. Oddy. Using cause-effect relations in text to improve information retrieval precision. *Information Processing and Management*, 37, pp. 119-145, 2001.

76. Edgar B. Wendlandt and James B. Driscoll. Incorporating a semantic analysis into a document retrieval strategy. In *Proceedings of SIGIR 1991*, pp. 270-279, Chicago, IL, October 1991.

77. Geoffrey Z. Liu. Semantic vector space model: Implementation and evaluation. *Journal of the American Society for Information Science*, 48(5), pp. 395-417, 1997.

78. J. Farradane. Relational indexing. Part I. *Journal of Information Science*, 1(5), pp. 267-276, 1980.

79. J. Farradane. Relational indexing. Part II. *Journal of Information Science*, 1(6), pp. 313-324, 1980.

80. J. Farradane and D. Thompson. The testing of relational indexing procedures by diagnostic computer programs. *Journal of Information Science*, 2, pp. 285-297, 1980.

81. B.C. Brookes. Jason Farradane and relational indexing. *Journal of Information Science*, 12, pp. 15-18, 1986.

82. Priscilla Caplan. *Metadata Fundamentals for All Librarians*. American Library Association: Chicago, IL, 2003.

83. Getty Research Institute. Art & Architecture Thesaurus. URL: http://www.getty.edu/research/conducting_research/vocabularies/aat/, Accessed: January 4, 2008.

84. Marti A. Hearst. Clustering versus faceted categories for information exploration. *Communications of the ACM*, 49(4), pp. 59-61, 2006.

85. Louise Spiteri. A simplified model for facet analysis. *Canadian Journal of Information and Library Science*, 23, pp. 1 - 30, 1998. Reprinted by the Information Architecture Institute at http://iainstitute.org/pg/a_simplified_model_for_facet_analysis.php.

86. William C. Mann and Sandra A. Thompson. Rhetorical Structure Theory: Description and construction of text structures. In *Natural Language Generation: Proceedings of the NATO Advanced Research Workshop on Natural Language Generation*, pp. 85-95, Nijmegen, The Netherlands, 1987.

87. Simone Teufel and Marc Moens. Summarizing scientific articles: Experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4), pp. 409-445, 2002.

88. Chris D. Paice. Constructing literature abstracts by computer: Techniques and prospects. *Information Processing & Management*, 26(1), pp. 171-186, 1990.

89. Elizabeth D. Liddy, Kenneth A. McVearry, Woojin Paik, Edmund Yu, and Mary McKenna. Development, implementation and testing of a discourse model for newspaper texts. In *Proceedings of the Workshop on Human Language Technology*, pp. 159-164, Princeton, NJ, 1993.

90. Elizabeth D. Liddy. The discourse-level structure of empirical abstracts: An exploratory study. *Information Processing & Management*, 27(1), pp. 55-81, 1991.

91. Elizabeth D. Liddy and Sung H. Myaeng. DR-LINK system: Phase I summary. In *Proceedings of the Annual Meeting of the ACL*, pp. 93-112, Fredericksburg, VA, 1993.

92. Gretchen P. Purcell, Glenn D. Rennels, and Edward H. Shortliffe. Development and evaluation of a context-based document representation for searching the medical literature. *International Journal on Digital Libraries*, 1, pp. 288-296, 1997.

93. Jack G. Conrad and Daniel P. Dabney. A cognitive approach to judicial opinion structure: Applying domain expertise to component analysis. In *Proceedings of the 8th International Conference on Artificial intelligence and Law*, pp. 1-11, St. Louis, Missouri, 2001.

94. Chris D. Paice and Paul A. Jones. The identification of important concepts in highly structured technical papers. In *Proceedings of SIGIR '93*, pp. 69-78, Pittsburgh, PA, 1993.

95. Lois Delcambre and David Maier. Models for superimposed information. Peter P Chen, et al., Editors, *ER '99 Workshops on Evolution and Change in Data Management, Reverse Engineering in Information Systems, and the World Wide Web and Conceptual Modeling*, Lecture Notes in Computer Science 1727: Springer, pp. 264-280, 1999.

96. David Maier and Lois Delcambre. Superimposed information for the internet. In *ACM SIGMOD Workshop on The Web and Databases (WebDB)*, pp. 1 -9, Philadelphia, PA, June 1999.

97. William Hersh, Andrew Turpin, Susan Price, Dale Kraemer, Daniel Olson, Benjamin Chan, and Lynetta Sacherek. Challenging conventional assumptions of automated information retrieval with real users: Boolean searching and batch retrieval evaluations. *Information Processing & Management*, 37, pp. 383-402, 2001.

98. Andrew Turpin and William Hersh. Why batch and user evaluations do not give the same results. In *Proceedings of SIGIR 2001*, New Orleans, Louisiana, September 9-12, 2001.

99. Andrew Turpin and Falk Scholer. User performance versus precision measures for simple search tasks. In *Proceedings of SIGIR 2006*, Seattle, Washington, August 6-11, 2006.

100. William Hersh and Paul Over. Interactivity at the Text Retrieval Conference (TREC). *Information Processing & Management*, 37(3), pp. 365-367, 2001.

101. Family Medicine Research Center. ICPC-2: Introduction. URL: http://www.fmrc.org.au/icpc2/, Last updated: December 4, 2005, Accessed: September 10, 2007.

102. World Health Organization. International Classification of Diseases (ICD). URL: http://www.who.int/classifications/icd/en/index.html, Accessed: September 10, 2007.

103. UpToDate. URL: http://www.uptodate.com, Accessed: January 17, 2008.

104. U.S.F.S. Environmental policy and procedures handbook. URL: http://www.fs.fed.us/emc/nepa/nepa_templates/handbook/epp.htm, Accessed: January 17, 2008.

105. U.S.F.S. Index of /emc/nepa/nepa_templates. URL: http://www.fs.fed.us/emc/nepa/nepa_templates/, Accessed: January 17, 2008.

106. National Information Standards Organization. ANSI/NISO Z39.19-2003 - Guidelines for the construction, format, and management of monolingual thesauri. URL: http://www.niso.org/standards/index.html, Accessed: February 24, 2004.

107. Betsey L. Humphreys, Donald A. B. Lindberg, Harold M. Schoolman, and G. Octo Barnett. The Unified Medical Language System: An informatics research collaboration. *Journal of the American Medical Informatics Association*, 5(1), pp. 1-11, 1998.

108. R. B. Haynes, N. Wilczynski, K. A. McKibbon, C. J. Walker, and J. C. Sinclair. Developing optimal search strategies for detecting clinically sound studies in MEDLINE. *Journal of the American Medical Informatics Association*, 1(6), pp. 447-458, 1994.

109. *Physicians Desk Reference*. 61 ed. Thomson PDR: Montvale, NJ, 2007.

110. Daniel C. Berrios, Russell J. Cucina, and Lawrence M. Fagan. Methods for semi-automated indexing for high precision information retrieval. *Journal of the American Medical Informatics Association*, 9(6), pp. 637-652, 2002.

111. Wanda Pratt, Marti A. Hearst, and Lawrence M. Fagan. A knowledge-based approach to organizing retrieved documents. In *AAAI-99: Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pp. 80-85, July 1999.

112. James J. Cimino, Stephen B. Johnson, Anthony Aguirre, Nancy Roderer, and Paul D. Clayton. The Medline button. In *Proceedings of the Sixteenth Annual Symposium on Computer Applications in Medical Care*, pp. 81-85, Baltimore, MD, 1992.

113. James J. Cimino, Anthony Aguirre, Stephen B. Johnson, and Ping Peng. Generic queries for meeting clinical information needs. *Bulletin of the Medical Library Association*, 81(2), pp. 195-206, 1993.

114. James J. Cimino, Gai Elhanan, and Qing Zeng. Supporting infobuttons with terminologic knowledge. In *Proceedings of the 1997 AMIA Annual Fall Symposium*, pp. 528-532, 1997.

115. Qing Zeng and James J. Cimino. Linking a clinical system to heterogeneous information resources. In *Proceedings of the 1997 AMIA Annual Fall Symposium*, pp. 553-557, 1997.

116. L. Rolling. Indexing consistency, quality and efficiency. *Information Processing & Management*, 17, pp. 69-76, 1981.

117. Marley W. Watkins and Miriam Pacheco. Interobserver agreement in behavioral research: Importance and calculation. *Journal of Behavioral Education*, 10(4), pp. 205-212, 2000.

118. Jean Carletta. Assessing agreement on classification tasks. *Computational Linguistics*, 22(2), pp. 249-254, 1996.

119. Barbara Di Eugenio and Michael Glass. The kappa statistic: A second look. *Computational Linguistics*, 30(1), pp. 95-101, 2004.

120. Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), pp. 1-47, 2002.

121. Yiming Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1-2), pp. 69-90, 1999.

122. David D. Lewis. Evaluating text categorization. In *Proceedings of the Workshop on Speech and Natural Language*, pp. 312-318, Pacific Grove, CA, 1991.

123. Justin Zobel. How reliable are the results of large-scale information retrieval experiments. In *Proceedings of SIGIR 1998*, pp. 307-314, Melbourne, Australia, 1998.

124. Ellen M. Voorhees and Chris Buckley. The effect of topic set size on retrieval experimental error. In *Proceedings of SIGIR 2002*, pp. 316-323, Tampere, Finland, August 2002.

125. Mark Sanderson and Justin Zobel. Information retrieval system evaluation: effort, sensitivity, and reliability. In *Proceedings of SIGIR 2005*, pp. 162-169, Salvador, Brazil, August 2005.

126. Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *Proceedings of SIGIR 1999*, pp. 42-49, Berkley, CA, USA, 1999.

127. Marie Adele Hughes and Dennis E. Barrett. Intercoder reliability estimation approaches in marketing: A generalizability theory framework for quantitative data. *Journal of Marketing Research*, 27, pp. 185-195, 1990.

128. William A. Scott. Reliability of content analysis: The case of nominal scale coding. *The Public Opinion Quarterly*, 19(3), pp. 321-325, 1955.

129. Matthew Lombard, Jennifer Snyder-Duch, and Cheryl Campanella Bracken. Content analysis in mass communication. Assessement and reporting of intercoder reliability. *Human Communication Research*, 28(4), pp. 587-604, 2002.

130. Klaus Krippendorff. Reliability in content analysis. *Human Communication Research*, 30(3), pp. 411-433, 2004.

131. Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), pp. 37-46, 1960.

132. Joseph H. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5), pp. 378-382, 1971.

133. Sidney Siegel and N. John Castellan. *Nonparametric Statistics, Second Edition*. McGraw-Hill: Boston, MA, 1988.

134. Rebecca J. Passonneau and Diane J. Litman. Intention-based segmentation: human reliability and correlation with linguistic cues. In *Proceedings of the 31st Annual Meeting of the Association of Computational Linguistics*, pp. 148-155, 1993.

135. M. A. Hearst. Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd annual meeting of the Association for Computational Linguistics*, pp. 9-16, Las Cruces, New Mexico, 1994.

136. Doug Beeferman, Adam Berger, and John Lafferty. Statistical models for text segmentation. *Machine Learning*, 34, pp. 177-210, 1999.

137. Lev Pevzner and Marti A. Hearst. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1), pp. 19-36, 2002.

138. James Allan. HARD Track Overview in TREC 2004. High Accuracy Retrieval from Documents. In *NIST Special Publication 500-261: The Thirteenth Text REtrieval Conference Proceedings (TREC 2004)*, Gaithersburg, MD, November 16-19, 2004.

139. William Hersh, Aaron M. Cohen, Phoebe Roberts, and Hari Krishna Rekapalli. TREC 2006 Genomics Track Overview. In *The Fifteenth Text REtrieval Conference Proceedings (TREC 2006)*, Gaithersburg, MD, November 14-17, 2006.

140. Ellen M. Voorhees. Overview of the TREC 2005 Question Answering Track. In *NIST Special Publication 500-266: The Fourteenth Text REtrieval Conference Proceedings (TREC 2005)*, Gaithersburg, MD, November 15-18, 2005.

141. ChengXiang Zhai, William W. Cohen, and John Lafferty. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *Proceedings of SIGIR 2003*, pp. 10-17, Toronto, Canada, July 2003.

142. Jordi Turmo, Alicia Ageno, and Neus Catala. Adaptive information extraction. *ACM Computing Surveys*, 38(2) 2006.

143. Jim Cowie and Wendy Lehnert. Information extraction. *Communications of the ACM*, 39(1), pp. 80-91, 1996.

144. Dagobert Soergel. Indexing and retrieval performance: The logical evidence. *Journal of the American Society for Information Science*, 45(8), pp. 589-599, 1994.

145. Klaus Krippendorff. Measuring the reliability of qualitative text analysis data. *Quality and Quantity*, 38, pp. 787-800, 2004.

146. Klaus Krippendorff. On the reliability of unitizing continuous data. *Sociological Methodology*, 25, pp. 47-76, 1995.

147. Ultraseek. URL: http://www.ultraseek.com, Accessed: August 1, 2007.

148. Mathew Weaver. Enhancing a domain-specific digital library with metadata based on hierarchical controlled vocabularies, Dissertation in *Computer Science*. Oregon Health & Science University, 2005.

149. Mathew Weaver, Lois ML Delcambre, Marianne Lykke Nielsen, Susan L Price, David Maier, and Timothy Tolle. Supporting domain-specific digital libraries in government: Two case studies. In Catherine Larson, Editor *Digital Government: Advanced Research and Case Studies.* Springer, New York, 2007, forthcoming.

150. procedure. Dictionary.com. *Dictionary.com Unabridged (v1.1)*, Random House, Inc. http://dictionary.reference.com/browse/procedure(accessed: October 19, 2007.) 2007.

151. Victoria Uren. An evaluation of text categorisation errors. In *Proceedings of Workshop on the Evaluation of Information Management Systems*, pp. 79-87, London, September 2000.

152. Miguel E. Ruiz and Alan Aronson. User-centered evaluation of the Medical Text Indexing (MTI) System. URL: http://ii.nlm.nih.gov/resources/MTIEvaluation-Final.pdf, Accessed: December 6, 2007.

153. Autonomy. URL: http://www.autonomy.com, Accessed: April 5, 2007.

154. Ultraseek. Ultraseek FAQ: How does Ultraseek create scores for items in the search results? URL: http://www.ultraseek.com/support/faqs/1070.html, Accessed: October 6, 2006.

155. Ellen M. Voorhees. Evaluation by highly relevant documents. In *Proceedings of SIGIR '01*, New Orleans, LA, September 9-12, 2001.

156. Geoffrey R. Norman and David L. Streiner. *Biostatistics. The bare essentials*. Second ed. B.C. Decker Inc.: Hamilton, Ontario, 2000.

157. William Hersh, Andrew Turpin, Susan Price, Benjamin Chan, Dale Kraemer, Lynetta Sacherek, and Daniel Olson. Do batch and user evaluations give the same results? In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 17-24, Athens, Greece, 2000.

158. Scott A. Golder and Bernardo A. Huberman. The structure of collaborative tagging systems. *Journal of Information Science*, 32(2), pp. 198-208, 2006.

159. Harry Halpin, Valentin Robu, and Hana Shepherd. The complex dynamics of collaborative tagging. In *WWW 2007*, Banff, Alberta, Canada, May 8-12, 2007.

Appendix A: Indexing Study Forms

Appendix A.1 Keyword Indexing Data Collection Form

Indexer Code _____ Document Number _____

Indexing Technique: **Keyword**

Remember to click the ***Start*** button before you start this task.

For this task we would like you to list both the concepts that you would like to index for each document and the keywords that you chose to represent the concepts in the index.

Please list the concepts that you would like to index for this document:

_____ _____
_____ _____
_____ _____
_____ _____
_____ _____
_____ _____
_____ _____
_____ _____
_____ _____

Please list the terms that you chose for this document and the source of the terms (the Almen thesaurus, ICPC, ICD-10, or free term)

Source	**Term**

Source	**Term**

Remember to click the ***Finish*** button when you have completed this task.

Appendix A.2 Document Classification Form

Indexer Code _____ Document Number _____

Indexing Technique: **Semantic Components**

Remember to click the ***Start*** button before you start this task.

Please place an ***X*** by the one document type you think best describes this document. The table below describes each of the document types.

_____ Clinical Problem: Documents about a Clinical Problem or Condition

_____ Procedure: Documents about Diagnostic or Therapeutic Procedures

_____ Services and Rights: Documents about Government Payment for Healthcare

Remember to click the ***Finish*** button when you have completed this task.

Document Type	**Short Name**	**Description**
Documents about a Clinical Problem or Condition	Clinical problem	Documents that are primarily about a particular clinical problem such as a disease, a symptom, or other clinical condition. Examples: - a normal condition, such as pregnancy - an abnormal condition, such as malnutrition or injury - a disease, such as diabetes - a group of related diseases or problems, such as knee injuries (could include information about several specific injuries) - a symptom, such as chest pain
Documents about Diagnostic or Therapeutic Procedures	Procedure	Documents that are primarily about a particular procedure, or possibly a group of related procedures, that are used to diagnose, treat or otherwise evaluate (e.g. determine the severity of) clinical problems. The documents are intended to convey practical information, usually to patients or their family, about how the procedure is performed, what the purpose and outcome will be, what to expect, etc. Examples: - surgical operations, such as coronary artery bypass surgery - radiologic examinations, such as colonoscopy - other types of procedures, such as a hearing exam.
Documents about rights and services to patients	Services and rights	Documents that describe a service that is offered to patients in general or with specific indications. The documents inform about possible services offered to all patients in Denmark, including the right to subsidised medication and “frit sygehusvalg”, and services offered to patients with specific indications, e.g. diabetes, dementia, obesity.

Appendix A.3 Semantic Components for the Clinical Problem Document Class

Document Number _____ Indexing Technique: **Semantic Components**

For the purposes of this experiment, please assume that this document is of the type: **Documents about a Clinical Problem or Condition** regardless of which document type you chose. Please determine which segments of text in the document, if any, contain information about each of the semantic components. Use pens to mark the text and label each segment with the name of the semantic component. Please be sure to label each marked segment. Text associated with a semantic component may be discontinuous or overlap segments for other semantic components. Some text in the document may not belong to any of these semantic components.

Remember to click the ***Start*** button before you start this task.
Remember to click the ***Finish*** button when you have completed this task.

Documents about a Clinical Problem or Condition	
Semantic components	
Name	**Description**
Evaluation	**How to diagnose or evaluate the problem.**
	Information about how to evaluate a patient who has, or might have, the clinical problem. Examples: - how to diagnose the disease - how to determine its severity or clinical stage - the differential diagnosis of a symptom is (what diseases could cause this symptom) - what screening tests are appropriate - what tests should be performed in patients who have this problem.
Management	**How to treat, manage or control the problem.**
	Information about how to treat or manage a patient who has the clinical problem. Examples: - formal disease management guidelines - how to prevent complications - how to reduce the severity or impact of the disease on the patient - how to monitor progression of a disease - recommended diet, education, or counseling - what medications or procedures are appropriate - what doses of medications to give
Referral	**How to refer a patient with the problem to a specialist or special service.**
	Information about how and when the family practitioner should refer a patient for specialist care. Examples: - criteria for referral (such as severity of disease, presence of certain complications) - how to make a referral (what number to call, where to mail documents) - what tests to do before the referral - what records to send to the specialist or special clinic
About	**About the problem.**
	General information about the condition, not necessarily for care of a particular patient. Examples: - natural history of a disease if not treated - the usual clinical course of patients with this problem - population statistics about how frequently the problem occurs - common co-occurring conditions or complications of the problem - etiology (causation) of the disease or condition.