

DEF-CAT: DBLP Expert Finder Utilizing Categories and Topics

By

PHILIP LEE FISHER-OGDEN
B.S. (University of California, San Diego) 2001

THESIS

Submitted in partial satisfaction of the requirements for the degree of

MASTER OF SCIENCE

in

Computer Science

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Committee in Charge

2008

Contents

List of Figures	iii
List of Tables.....	iv
Chapter 1: Introduction	1
1.1 Motivation	2
1.2 Objectives	3
1.3 Thesis Overview	5
Chapter 2: Background	6
2.1 Expert Finder Systems	6
2.2 Taxonomies	9
2.3 Information Retrieval.....	14
Chapter 3: Researcher/Topic-centric Expert Finder.....	21
3.1 Introduction	21
3.1.1 Who Knows What?	21
3.1.2 Definition of an Expert.....	22
3.1.2 Definition of Topics	24
3.2 Background.....	26
3.3 Approach	30
3.3.1 Expertise Sources	31
3.3.2 Data Model	34
3.3.3 Retrieving Expertise Areas	38
3.3.4 Retrieving Experts.....	44
3.4 Implementation	49
3.4.1 Prototype.....	49
3.5 Evaluation.....	55
3.6 Discussion	62
Chapter 4: Taxonomy-enhanced Expertise Browser.....	66
4.1 Introduction	66
4.2 Background.....	67
4.3 Approach and Implementation	69
4.4 Discussion	78
Chapter 5: Related Work	83
Chapter 6: Conclusions and Future Work.....	95
Bibliography.....	97

List of Figures

Figure 1. Taxonomy Concept Map (from [11])	9
Figure 2. Yahoo! Directory Sample	11
Figure 3. Classification Nodes for "Distance Browsing In Spatial Databases"	14
Figure 4. Inverted Index Creation Diagram.....	15
Figure 5. Example DBLP Record.....	33
Figure 6. Example Publication Entry from the ACM Portal	34
Figure 7. DBLP Entry Types	35
Figure 8. Publication Data Model	36
Figure 9. Publication Content Model	36
Figure 10. Expert Finder System Data Types	37
Figure 11. Hanan Samet's Self-Reported Research Areas.....	39
Figure 12. Keywords from Hanan Samet's ACM Author Profile Page.....	41
Figure 13. DEF-CAT System Overview Diagram	50
Figure 14. Publication Data Model	51
Figure 15. Content Type and Related Data Types	52
Figure 16. Topic Experts Diagram	54
Figure 17. Expertise Profile Diagram.....	54
Figure 18. Document Clustering of Search Results (clusty.com)	68
Figure 19. Faceted Navigation (nextag.com).....	69
Figure 20. DEF-CAT UI - Starting Page for Three Different Browse Modes.....	69
Figure 21. Taxonomy Browse Interaction Model	71
Figure 22. DEF-CAT UI - Taxonomy Browse	72
Figure 23. Group Navigation Interaction Model.....	74
Figure 24. DEF-CAT UI - Grouped Navigation	75
Figure 25. DEF-CAT UI - Profile View	76
Figure 26. DEF-CAT UI - Profile View Grouped Navigation	77
Figure 27. DEF-CAT UI - Profile View Timeline	77

List of Tables

Table 1. 1998 ACM CCS Top Level Nodes	12
Table 2. Tokenization Example	16
Table 3. Linguistic Preprocessing Examples	17
Table 4. Metadata for a Recent Publication by Hanan Samet	26
Table 5. Document Metadata Vocabulary Types	26
Table 6. DBLP Entry Types.....	32
Table 7. Build Topic Vocabulary Algorithm.....	41
Table 8. Key Phrases Pruning Algorithm	42
Table 9. Vocabulary Pruning Rules.....	43
Table 10. Example of Applying Topic Extraction to a Title	43
Table 11. Find Relevant Publications Algorithm.....	45
Table 12. Most Frequent Authors Algorithm.....	46
Table 13. First in Field Algorithm.....	47
Table 14. Expertise Areas for Hanan Samet #1	57
Table 15. Expertise Areas for Hanan Samet #2	57
Table 16. Evaluation Results for Hanan Samet's Expertise Areas.....	59
Table 17. Spatial Databases Experts.....	60
Table 18. Spatial Databases - Clusters and Top Authors	60
Table 19. An Intuitive Domain Model of Expert Finding Systems	85

Abstract

Expert finder systems answer two questions: “who knows what” and “what is known by whom”. Effectively answering those questions requires determining expertise areas from evidences of an individual’s work and designing a method of ranking experts against the topics and one another. We propose an approach to expert finding that leverages author-provided content like key phrases and taxonomy classifications to describe topics of expertise. We show how author-provided metadata can be turned into a topic vocabulary, enabling topic detection in publications that lack explicit metadata. We augment these topics with implicit subtopics and related topics, which we extract through document clustering techniques. Expertise profiles can be generated from extracted expertise areas, presenting contextual views of an author’s research publications and summarizing the areas of their contributions. We implement the components of our approach in DEF-CAT, the DBLP-Expert Finder Utilizing Categories and Topics, a system for expert finding and exploratory search across researchers and topics in the field of computer science. The 1998 ACM Computing and Classification Taxonomy provides a structured hierarchy of topics that we use for navigation purposes in our search system. We show how the taxonomy can be extended by applying our topic vocabulary and document clustering techniques to the publications in each node, producing cohesive subgroups of publications within a node.

Chapter 1: Introduction

Information needs arise in every situation in life. These needs vary from ones that can be met by oneself to those that require an expert with years of experience and education. How does one identify the person that can answer these information needs: What will the weather be like today? Where can I find the best price on that popular technology item? What parenting practices should I adopt for dealing with a child with disabilities? Where should I invest my money to appropriately balance risk and return? Satisfying these information needs requires identifying a set of experts, selecting from among them, and then accessing their expertise to answer the given information need.

The core question within all expert seeking ventures is “Who knows what”. Finding an accessible expert can be done through formal means, as in hiring a financial planner to answer your wealth management needs, or informal means, such as asking a question to the contacts in your social network. Once a set of potential experts has been identified, criteria are applied to assess their expertise and credibility. Ultimately, the determination of who knows what decides which experts should be contacted.

Expert Finder Systems (EFS) seek to automate the process of finding an expert on a given topic. To do so, these systems aggregate data and other artifacts that can be used to identify areas of expertise. Once the expertise areas are identified, these systems apply various techniques to produce rankings among experts for a given topic. The expertise determination process of an automated EFS models the process that humans go through when determining expertise (studied by McDonald and Ackerman [1]). But, the process does so with amounts of data that would easily and quickly overwhelm attempts to

manually identify experts. EFS unlock the ability to extract experts and expertise areas from data sets that are too large for human-based analysis.

1.1 Motivation

Expert finder systems can answer information needs in a variety of areas. The following hypothetical situations provide various motivations for building an automated expert finder system.

Before submitting a paper for publication, a researcher may wish to have it undergo a pre-publication review by an expert in the field. The pre-publication review can help with identifying any necessary modifications so that the chances of rejection in a later stage will be minimized. But, this requires identifying an individual who is an expert in the area and who is also willing to provide that pre-review. An expert finder system built from existing publications in the field can enable identifying candidates for the pre-publication review. To increase the chances of pre-review, the list of suggested experts could be limited to those who are in a given researcher's extended social network.

In the area of enterprise competency management, a company needs to know who knows what in order to effectively utilize its employees. As Becerra-Fernandez [2] suggests, an enterprise can easily assemble a team for a new project if it can find experts for the range of areas that the project covers. If the company needs to engage in workforce restructuring, it can establish core competencies for its workgroups and then use the expert finder system to identify who are the experts in those competencies. Those experts could be protected from layoffs, to avoid the loss of employees with the most valuable skills. Training needs could be determined by surveying expertise areas in a business unit and identifying key skills that are not adequately represented. Automated

expert finder systems enable more efficient management of competencies in an enterprise.

Communities of practice (CoP) are formed when individuals with similar interests coalesce and participate together. The participants could be researchers who have sought out domain experts, within their same discipline or across disciplines, to collaborate with. They could be thought leaders in a company who identified employees with similar interests and formed a think tank. Alternatively, they could be peers identified by an automated system like [3], in which McDonald describes an approach to recommending collaboration. Regardless of the type of participants, the CoP formation started with identifying experts for a given area or areas. Expert finder systems can identify experts in a given topic or topics, enabling the formation of a CoP.

Each of these situations requires converting evidence of expertise, for example documents, publications, correspondence, and web pages, into relevant expertise areas. Some organizations may have internal expertise brokers that can act on a member's behalf to identify experts. However, this requires the broker to stay current on every possibly expertise topic. This would consume and quickly overwhelm even the best of expertise brokers. Automated expert finder systems act as an enabling technology to offer a solution where an expertise broker could not.

1.2 Objectives

The primary goals of this thesis center on identifying expertise areas and extending a computer science taxonomy to facilitate browsing for experts. For the expertise areas, the goal is to synthesize associations between topics and experts, enabling expertise identification for a given topic or a given researcher. For the expertise

browsing, the goal is to extend an existing taxonomy to produce an exploratory interface to browse for experts in. Taken together, these goals form the foundation for the contributions from this work. We demonstrate an implementation of these goals called DEF-CAT, the DBLP-Expert Finder Utilizing Categories and Topics.

We offer research contributions that are applicable to finding experts in the Digital Bibliography and Library Project (DBLP) [4] data set, as well as being generalizable to expert finder systems for other domains. First, we show how author-provided metadata can be used to identify expertise areas. Specific to our document collection, we leverage author-provided key phrases to build a topic vocabulary for computer science. By combining that vocabulary with taxonomy nodes and document clusters, we provide a rich representation of identified expertise areas. Second, we show how to identify experts on a given topic using more than the naïve approach of most published authors. We do this by looking at not only the most frequent authors, but also those that were first in the field and those that are actively publishing in the field. We utilize document clustering to further extend this by providing context around each identified expert’s areas of contribution. Finally, we show how to leverage an extended form of a domain taxonomy to improve the expert finding process. We enhance nodes in the 1998 ACM Computing and Classification Survey (ACM CCS) taxonomy by extracting subtopics and related topics for each node. We synthesize these components together into a prototype system implementation, demonstrating how they enable expert finding and expertise profile generation while also supporting browsing for and discovering experts in an exploratory search UI.

1.3 Thesis Overview

The rest of this thesis is organized as follows: Chapter 2 reviews background concepts that are needed to understand our approach. Chapter 3 discusses identifying experts for a given topic and enumerating expertise areas for a given researcher. Chapter 4 investigates how to browse for experts and expertise areas by leveraging an extended version of the ACM CCS taxonomy. Chapter 5 reviews related work, positioning our approach in the context of what others have done, and compares and contrasts our system with other related systems. Chapter 6 concludes the thesis and discusses areas for future research.

Chapter 2: Background

In this chapter we review background concepts and related work to establish a baseline understanding of the concepts we use in future chapters. First, we present expert finder systems, including their motivations, and various implementations. We then discuss taxonomies and how they assist with knowledge organization, a key element in our exploratory expertise browsing interface. Finally, we review relevant concepts within the field of information retrieval.

2.1 Expert Finder Systems

Expert finder systems connect a seeker to a provider: one who requires information needs to find one that has knowledge of or access to that information. These systems provide this service through one or more of a diverse range of approaches. The approaches vary in the sources of expertise indicators that they utilize and in the amount of automation that they leverage.

One set of approaches to locating expertise focuses on accuracy in domain representation and reporting. Domain experts create knowledge taxonomies or skill descriptions and then potential experts self-report their competency in the areas (e.g., HP's CONNEX [5]). Many organizations have adopted the approach of maintaining a skills database (e.g., Microsoft's SPUD [6]), which catalogues expertise skills and individuals that are competent in those skills. If kept up to date, a skills competency database can be useful when looking to assemble a team of employees with a specific set of skills. However, the high cost to manually update the information in these types of systems has led to their decline in favor of more automated approaches.

An alternate set of approaches introduce automation and more heterogeneous sources of expertise information. Manually maintained profiles tend to become stale as the experts that provided the profiles frequently have higher priorities than updating their profiles. Automatically generated profiles remove this barrier by leveraging automated extraction techniques, though the accuracy of the identified results may be lower than self-reporting. Expert/Expert-Locator (EEL) [7] uses Latent Semantic Indexing techniques across document produced by various technical groups to identify the best group to answer a technical help request. NASA's Expert Seeker [2] draws evidence of expertise from multiple sources: HR databases, skills databases, and accomplishments listed in a performance evaluation system. By embracing automation and a wider range of information sources, these expert finder systems offer a more sustainable model and mechanism of maintaining expertise profiles.

The introduction of the expert finder task in the 2005 TREC Enterprise Track [8] has helped with generating active research into the problem of automatically associating experts with expertise areas. One such example come in Balog et al.'s designing of formal models for identifying expertise areas [9]. They use the TREC W3C-corpus and associated topic lists to empirically assess and evaluate their two models. The W3C-corpus is a web crawl of the W3C website combined with a list of candidate experts and topics, and researchers are challenged to identify the associations between experts and topics. Balog et al.'s first formal model collects all documents for a given candidate, and then uses the model to estimate the probability of this candidate knowing about a given query topic. If the topic in consideration is likely to be derived from the candidate's documents, then the association score is higher. In their second model, expertise in a

given topic is assessed by retrieving all relevant documents and then analyzing who are the candidates within those documents. Their evaluation of the two models concludes that the second model outperforms the first model in all assessments. It also has the advantage of being an online algorithm and can be implemented easily if an index of documents already exists. Balog et al.'s contributions, and others that have resulted from the TREC expert finder tasks, provide proven approaches to automatically associating topics with expert candidates.

Expert finder systems have not only been popular systems of study for academics, but many commercial systems have also been developed and deployed. The MITRE Corporation published an excellent review of existing commercial expert finder systems [10], comparing them to one another along the lines of these key feature clusters: sources, processing, search, results, and system properties. Sources means what range of input types does a system support, e.g., e-mails, web pages, pdfs, and so on. Processing covers features like entity extraction, language identification, and author identification, which surprisingly none of the reviewed systems supported. Search includes the types of searches supported, like keyword, Boolean, and natural language, and also browsing using a taxonomy. The results cluster highlights a system's support for more than just a ranked list of experts, for example, related documents and concepts. Finally, the system feature includes if customers are currently using the system, its interoperability, and any privacy mechanisms. This report is a valuable resource to anyone who is considering implementing an expert finder system within their organization, if not just for comparison then also for evaluation of what can be purchased instead of building a system from scratch.

2.2 Taxonomies

Taxonomy is broadly defined (in Figure 1 from [11]) as a knowledge map, a classification scheme, and a semantic representation. In the context of our work, we narrow the definition of taxonomy to be defined as a structured knowledge hierarchy. It provides order to information and defines hierarchical relationships within its elements. The taxonomy has a root node that indicates the general unifying theme among all nodes. Each subsequent non-root node is a specialization of its parent node and a generalization of any of its child nodes. Items can be classified into nodes in the taxonomy, indicating that the item shares similar characteristics with other items in the node.

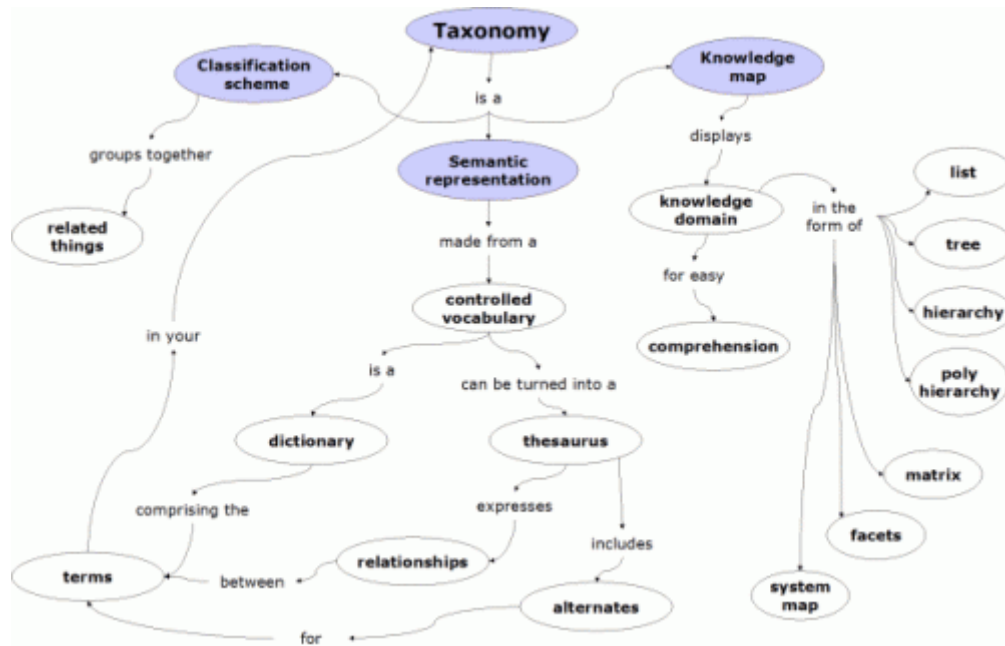


Figure 1. Taxonomy Concept Map (from [11])

Using taxonomy as an organization system for information enables simpler comprehension of the types of elements and their relationships than a structure-less view of those elements. Taxonomy's structure summarizes what the taxonomy creators

considered to be the important characteristics of elements and the similarities and differences between them. However, the use of taxonomies is not without its critics, for example, Follette and Houts [12] provide arguments on why a certain taxonomy should not be used. Taxonomies are inherently biased towards the ideas of their creators and may be too broad or too narrow depending on the context in which they are applied.

Taxonomies have been extensively developed and widely used in a broad range of fields. One of the most well known examples is biology's taxonomy of life, also referred to as the five kingdoms system. It was developed by the biologists Lynn Margulis and Karlene V. Schwartz, who extended work by previous taxonomists like Carolus Linnaeus and Charles Darwin [13]. Biological organisms are classified into the nodes of a hierarchical structure that consists of kingdom, phylum, class, order, family, genus, and species.

Another widely recognized taxonomy is the Yahoo! hierarchical directory of Internet web sites [14] (Figure 2). Its structure consists of fourteen top-level categories and multiple sub-categories, and its content is manually categorized by a team of editors. The Yahoo! taxonomy provides functionality to browse and search within a section of the taxonomy, allowing for more focused results to be returned. Prior to the rise in popularity of Google's ranking algorithm, directory-style Internet searching and browsing was the lingua franca of Internet searches (e.g., AltaVista's use of the LookSmart directory [15, 16]). This style still lives on in the Yahoo! directory and the Open Directory Project, though with diminished popularity.



Figure 2. Yahoo! Directory Sample

Taxonomy has proven useful in not only classification scenarios, e.g., the biological case, and information browse scenarios, e.g., the Yahoo! taxonomy, but also in the area of information creation. Bloom's taxonomy of educational objectives [17, 18] offers principles for producing effective instructional materials. The taxonomy divides the cognitive domain into six areas, ranging from lower-order thinking skills (i.e., remember, understand, apply) to higher-order thinking skills (i.e., analyze, evaluate, create). Educational practices that utilize higher-order thinking skills are more likely to increase engagement, comprehension, and retention. It is well respected and utilized within the field of education, i.e., teaching and learning, and of educational psychology.

In the field of computer science, the ACM Computing and Classification System (ACM CCS) [19] is a taxonomy used to classify publications in ACM journals. It has been developed and improved over time, with full details available online for the 1964, 1991, and 1998 versions. Prior to submitting a paper for publication by the ACM, authors can use the guidelines in [20] to annotate their publication with the relevant categories. ACM aims to guarantee accuracy and relevancy in the classifications of

accepted and published papers by having the authors categorize their work, as they are the most knowledgeable about their work. The benefits of accurate classification is that it “provides the reader with quick content reference, facilitating the search for related literature, as well as searches for your work in ACM’s Digital Library and on other online resources. It also ensures correct placement when a review appears in Computing Reviews” [20].

The ACM CCS taxonomy is structured as a three level tree, with an additional optional fourth level of subject descriptors. The most recent version of the taxonomy that is in use is the 1998 version [21]. The 1998 taxonomy consists of 1473 nodes split into 11 first level, 81 second level, 400 third level, and 981 fourth level descriptors. The 11 top level nodes are shown in Table 1. “General” and “Miscellaneous” nodes are included in the second and third levels of the taxonomy to allow for publications to be classified in a generic node or outside the scope of existing nodes. Nodes from previous taxonomy versions are retained for searching purposes and are marked as being unused or with a link to a node that acts as their replacement.

Table 1. 1998 ACM CCS Top Level Nodes

Top Level Node ID	Node Name
A.	General Literature
B.	Hardware
C.	Computer Systems Organization
D.	Software
E.	Data
F.	Theory of Computation
G.	Mathematics of Computing
H.	Information Systems
I.	Computing Methodologies
J.	Computer Applications
K.	Computing Milieux

Authors categorize their papers into one or more nodes in the taxonomy. The first node is considered the primary classification and should be the closest to the main contribution area for the paper. Analysis of the categories for existing similar publications is encouraged, to ensure consistency within a given category. Additional implicit descriptors can be added, where the implicit descriptors are proper names like JAVA and FORTRAN that are not explicitly defined in the taxonomy but may be relevant to a paper's core contributions.

A paper's primary classification anchors its contributions in a core area, while secondary classifications provide necessary context around those contributions. As an example, consider one of the most highly cited publications in the ACM Transactions on Database Systems journal: *Distance browsing in spatial databases* by Hjaltason and Samet [22]. In this publication, the core contribution is that it shows "that the incremental nearest neighbor algorithm significantly outperforms the k-nearest neighbor algorithm for distance browsing queries in a spatial database that uses the R-tree as a spatial index". The authors chose to classify it (Figure 3) primarily in the spatial database category as it contributes a new approach to a central function of a spatial database system. They classified it secondarily in the tree data structure category as they showed how to efficiently implement distance browsing using a tree data structure.

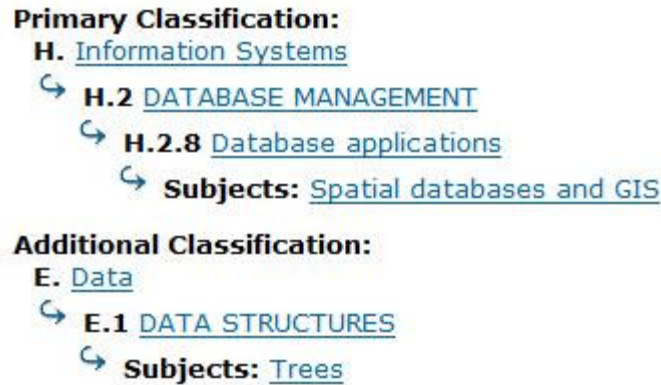


Figure 3. Classification Nodes for "Distance Browsing In Spatial Databases"

Taxonomy facilitates information organization and representation of interrelated and often complex concepts. In the field of computer science, the 1998 ACM CCS taxonomy provides structure to the complex subjects of a multitude of publications. Taken by itself, this taxonomy allows for coarse grain browsing of related publications, enabling discovery of related work and also understanding of the progression of contributions in that topic. However, many of the categories have over 200 publications, a size which inhibits the full potential of browsing and discovering. Additionally, the ACM CCS taxonomy is lacking nodes for recent popular topics like peer-to-peer systems. Apart from updating the entire taxonomy, providing additional context at each of the leaf nodes can help overcome this staleness by identifying subtopics like peer-to-peer systems in nodes like "C.2.4 Distributed Systems – Distributed applications". We address these problems in Chapter 4 when we present our approach to expertise browsing and exploration through an extended version of this taxonomy.

2.3 Information Retrieval

Information retrieval is a broad field of study in computer science that deals with the mechanisms of effectively retrieving information from typically large data sets (see

[23] for an excellent overview of the field). Our approach uses standard proven techniques in the field of information retrieval to retrieve documents relevant to an expertise query. In this section, we review the concepts in the field that are relevant to our approach.

Inverted Index. An inverted index is an index structure that allows efficient retrieval of documents. It is called an inverted index because it provides inverted access to its contents, that is, items are retrievable via any of their contained elements. Building an inverted index involves translating input items into searchable elements and building a persistent store of those elements. Retrieving answers to queries against that index requires determining how to process client queries against that data store. An overview of the inverted index creation process is provided in Figure 4.

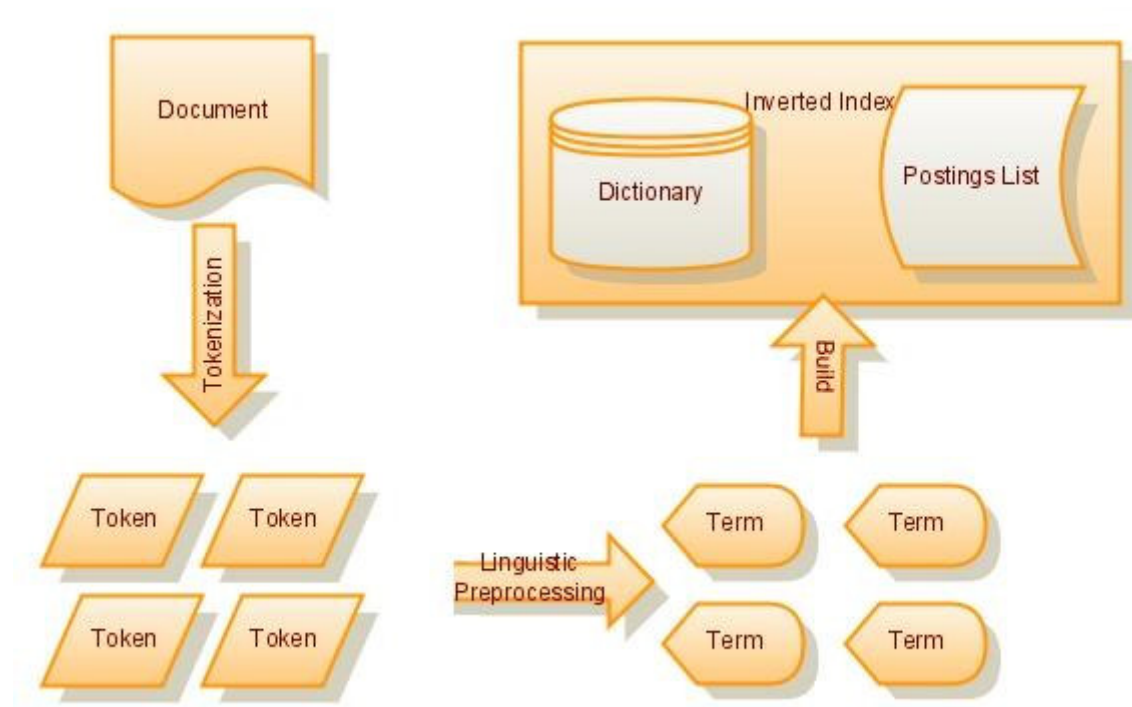


Figure 4. Inverted Index Creation Diagram

Tokenization. When indexing textual content, the text needs to be broken up into the smallest units of the textual elements. Segmenting the text into these units is the process of tokenization. Given an input text stream, tokenization will produce tokens from that stream. A common approach to tokenization is to segment on whitespace, shown in the example in Table 2.

Table 2. Tokenization Example

Tokenization Strategy	Input Text	Output Tokens
WhitespaceTokenizer	Distance Browsing in Spatial Databases	Distance, Browsing, in, Spatial, Databases

Linguistic Preprocessing. Tokens must be converted into indexing terms prior to index creation. The conversion process can be as simple as taking each token as-is or simply removing characters like punctuation or accented characters, or it can be as complex as reducing the token down into a normalized form. Common words like “a, an, and, for, the” are called *stop words* and may be removed. The intent of the various preprocessing techniques is to take the input tokens and produce terms that are relevant to the content being indexed, with little to no loss of important information.

Two techniques for reducing the tokens down are *stemming* and *lemmatization*. To stem a word is to reduce it down to a canonical form, for example by dropping a suffix if one exists. This form is the same between singular and plural terms and normalizes the term between multiple forms of the same root. Lemmatization is a related concept, which is like stemming except that it maintains the part of speech along with the reduced canonical form of a term. Lemmatization enables distinguishing between equal stems that are derived from unequal words like bank the noun, a financial institution, and bank the verb, an action involving inward tilting. Table 3 shows examples of some of these

techniques applied to the sentence “Ralf Hartmut Güting wrote ‘A foundation for representing and querying moving objects’”.

Table 3. Linguistic Preprocessing Examples

Preprocessing Technique	Input Tokens	Output
Strip punctuation	Ralf, Hartmut, Güting, wrote, 'A, foundation, for, representing, and, querying, moving, objects'	Ralf, Hartmut, Güting, wrote, A, foundation, for, representing, and, querying, moving, objects
Replace accented characters	Ralf, Hartmut, Güting, wrote, A, foundation, for, representing, and, querying, moving, objects	Ralf, Hartmut, Guting, wrote, A, foundation, for, representing, and, querying, moving, objects
Remove stop words	Ralf, Hartmut, Guting, wrote, A, foundation, for, representing, and, querying, moving, objects	Ralf, Hartmut, Guting, wrote, foundation, representing, querying, moving, objects
Apply stemming algorithm	Ralf, Hartmut, Guting, wrote, foundation, representing, querying, moving, objects	Ralf, Hartmut, Guting, wrote, foundat, repres, queri, move, object

Document representation. How we represent a document in the inverted index will have an impact on what types of retrieval methods we allow. Two common document representations are the *bag-of-words* representation and the *n-gram* representation. The bag-of-words representation views documents as order-less bags of words, only requiring the words contained within a document to be included in the model, but not their absolute or relative positions. The n-gram representation segments sequences into grams, which are typically word grams or character grams. For example, all character trigrams in the word “spatial” are “spa, pat, ati, tia, ial”. N-gram representations have proven themselves in many other areas, for example language recognition, and they also work well when a retrieval application needs to support wildcard searches.

For our purposes the bag-of-words representation of a document is more appropriate than the n-gram approach. However, representing a document as a bag-of-words requires more than just the decision to represent single word units rather than character or word grams. To sufficiently model a document and compute similarities between documents requires a more robust model: the vector space model.

Vector Space Model. In the vector space model [24], documents are represented by a vector of terms. The vector space consists of all the terms in the index's dictionary. A vector representation for a given document contains non-zero values in the dimensions that correspond to terms in that document.

Determining the numerical values for a given term/dimension in the vector space model involves determining term importance within the document and across the collection. We utilize a technique called term frequency-inverse document frequency (TF-IDF) to determine the values for each dimension. The term frequency is the frequency of occurrence of a given term in a document over the number of terms in the document. This ratio normalizes the term value across documents of different sizes. The inverse document frequency (IDF) acts as a measure of term importance. IDF is calculated by taking the log of the total number of documents over the number of documents that contain the given term. IDF boosts the importance of rare terms while also dampening the contributions of low information terms that occur in a large portion of a collection. Combined together, TF-IDF produces a value for a given term in a document that describes its relevance within the document and the entire collection.

Each document's term vector forms a numerical representation that can be used in a similarity function to determine two document's similarity to one another. The

similarity between two documents can be interpreted as the distance between the documents, with similar documents being closer together. Cosine similarity computes document similarity by determining the angle between each document's term vectors. The cosine similarity is equal to 1 when the documents are identical and is 0 when the documents have nothing in common.

Dictionary and Postings List. The core data structures in an inverted index are the dictionary and postings list. The dictionary consists of all unique terms output from the linguistic preprocessing step and may include additional information like the frequency of a given term in a document collection. The postings list catalogues occurrences of dictionary terms in documents. It can answer questions like “which documents contain the word ‘database’ in them”. Occurrence entries may include additional information beyond a document reference like the frequency of a given term in a document or the position within the document where the term was found. If this information is recorded, queries like “which documents have the words ‘spatial’ and ‘database’ in them, in consecutive order” can be answered. Together, the dictionary and postings list provide an efficient index structure for many information retrieval needs.

Retrieval. All of the previously mentioned IR concepts allow creation of an efficient, effective index structure for textual content. However, IR not only involves the storage of the information but also the mechanisms for retrieving that information. Retrieval is the process of taking an input sequence of terms from a client to the system and determining the relevant set of documents to return. Two well known and widely used retrieval models are the Boolean retrieval model and the ranked retrieval model. The Boolean retrieval model supports Boolean queries, where terms are combined

together using Boolean operators like AND, OR, and NOT. All documents that satisfy the Boolean query are returned. In contrast, the ranked retrieval model orders the documents that it deems relevant, returning a ranked list of results. The ranked retrieval model supports free text querying, providing easier query construction for the users of the system. Input queries are represented as vectors in the vector space model, and their similarity to indexed documents is determined by using the cosine similarity between elements.

Chapter 3: Researcher/Topic-centric Expert Finder

3.1 Introduction

This chapter formalizes the notion of an expert and presents our approach to solving the problem of identifying expertise areas and connecting them to researchers. Research into techniques for finding experts is very active, spurred on by events like the addition of an expert search task to the enterprise track of the TREC conferences in 2005 [25] and workshops like the 1st International ExpertFinder Workshop, 2007 [26], devoted to expert finding. To these efforts, we contribute an approach that:

- Automatically builds a vocabulary of topics for the computer science field, leveraging publication metadata contributed by thousands of researchers.
- Summarizes expertise of researchers through extracting frequent taxonomy nodes, topics and topic clusters from their publications.
- Produces contextual views of experts on a given topic, expanding on the simple approach of “most frequent authors” as experts.

3.1.1 Who Knows What?

The question of “who knows what?” requires connecting areas of knowledge with knowledgeable researchers. Apart from one another the two are relatively useless to someone seeking out an expert, but in conjunction they provide the necessary confluence of information to meet an expertise need. In this chapter we explore two problems central to connecting topics to experts:

1. Given a researcher, produce a list of their expertise areas.

2. Given a topic, identify the experts and rank them according to expertise on this topic.

3.1.2 Definition of an Expert

Webster’s dictionary defines an expert as “having, involving, or displaying special skill or knowledge derived from training or experience” [27]. An ideal expert finder system could assess the breadth of areas of expertise for a given researcher by mining each item in this definition: skills, knowledge, training, and experience. An ideal system would have access to each of these types of expertise evidence and then could extract expertise areas. By comparing the extracted areas and evidence against similar researchers, a researcher’s impact and context within a given field could be determined.

We take a pragmatic view of these ideals, to assess what is feasible to implement and evaluate. Skills databases offer structured evidence of expertise, but are hard to acquire and can suffer from stagnant or outdated information. Manually created profiles, like those listed on an author’s home page or posted to a site like Community of Science [28], often include explicit listings of these: research areas, academic degrees, positions held, awards, and publications. If a system had a way to acquire every manually reported profile then it could attempt to extract expertise areas from the manual profiles, achieving high recall when retrieving expertise areas for each researcher.

We make the simplifying assumption that focusing solely on a researcher’s publications appropriately balances accuracy with the overhead required to acquire various data sources. We focus our approach on expertise evidence in the form of refereed publications. Our approach defines an expert as a researcher who has

demonstrated publications on a given topic that satisfy one or more of the following criterion:

1. *Relevant* – publications that support a proposed area of expertise must be related to that area. This is the simplest of the three criteria, but is necessary to establish a baseline of publications to use for determining expertise areas or experts.
2. *Frequent* – repeated peer-reviewed publications on a given topic are strong indicators of expertise.
3. *Recent* – for career academics, recent publications are tied to active research interests. A researcher’s recent areas of focus are stronger indicators of active expertise than past focus areas, assuming that the recent areas are of similar or greater frequency.

However, our definition of expertise is missing support for high impact authors that have a low frequency of publications in our data set. For example, DBLP contains only nine publications from Claude Shannon [29], even though he is considered the father of information theory [30]. If we included citation information in our data set used to determine expertise, then we would have the necessary metadata to extract such high impact researchers. There is a wealth of existing work on using citation link analysis for finding experts (e.g., Tho, Hui, and Fong [31]), but not a lot of research into building expert topical profiles using clustering and topic vocabularies. We acknowledge this weakness in our system, but assert that our solution can complement other proposed solutions like those focused on citation link analysis.

3.1.2 Definition of Topics

Topics form the blocks from which expertise profiles are built. We define a topic as a specific research area indicated in a publication's author-provided key phrases, taxonomy nodes (see Section 2.2), or descriptive terms from document clusters. The set of all author-provided key phrases from a set of publications are an important part of building a complete vocabulary of topics for those publications' domain. In the following paragraphs, we motivate why we chose to leverage key phrases as one source of topics.

Establishing topics in a given domain is a non-trivial effort, requiring either input from domain experts or automated extraction or a combination of both. Domain experts can provide accurate, descriptive topics for a given domain but require time to develop the topics and then time to improve them as the field advances. Extracting topics manually can ensure that the topics are up to date, but may not produce topics as descriptive or consistent as domain expert determined ones.

When asked to identify their areas of expertise, researchers will typically provide a few high level topics that describe their research interests. For example, on the homepage of Professor Hanan Samet, the most highly published researcher on spatial databases and author of the foundational book on multidimensional data structures, he lists his research interests as "Spatial Databases, Data Structures, Geographic Information Systems, Image Databases, Computer Vision, and more" [32]. However, an expert finder system needs more detail than these high-level topics provide in order to build expertise profiles and perform accurate retrieval of experts. Fortunately, researchers' provide another source of expertise areas: the keywords that they use to describe their publications.

Publishers that request for index terms / keywords to be submitted as metadata alongside a publication are asking researchers to perform the cognitive task of identifying relevant topics for their publication. The phrases that a researcher uses to describe the key terms in a publication can consist of general topics, like the ones listed on Professor Samet's homepage, and specialized topics that can more accurately characterize the topic of the publication. As an example, let us consider a recent publication by Professor Samet (Table 4) [33]. None of the research interest topics listed on his homepage occurs in the title or abstract of this publication. The topic "Data Structures" could be inferred from the provided metadata, if we knew that quadrees are a type of data structure. Otherwise, the topics listed on the homepage provide a weakly related set of topics for this publication. A more strongly related set of topics exists in the author-provided keywords field. "Distributed data structures" and "Spatial data structures" are subtopics of "Data Structures", whereas "Peer-to-peer networks" is an entirely different topic, though it relates to the other topics in this paper's context. Clearly, author-provided key phrases provide a rich set of topics for publications. A reasonably accurate, though possibly expansive, set of topics for an expertise profile can be built from these key phrases. By combining them with other sources of topics, i.e., taxonomy nodes and terms describing document clusters, we achieve a comprehensive set of topics to use for describing expertise areas.

Table 4. Metadata for a Recent Publication by Hanan Samet

Title:	<i>Using a distributed quadtree index in peer-to-peer networks</i>
Abstract excerpt:	Peer-to-peer (P2P) networks have become a powerful means for online data exchange. ... In this paper, a distributed quadtree index that adapts the MX-CIF quadtree is described that enables more powerful accesses to data in P2P networks. ... Our index is easy to use, scalable, and exhibits good load-balancing properties. ...
Keywords:	<i>Distributed data structures, Peer-to-peer networks, Quadtrees, Spatial data structures</i>

3.2 Background

Vocabularies. Mathes [34] identified that vocabularies for document metadata typically originate from one of three sources: professionals, authors, or users.

Vocabularies vary by who creates them, how are they structured, and how they are maintained. A summary of these aspects is provided in Table 5, with more detailed discussion following. The three vocabulary types should not be considered exhaustive (i.e., automatic topic discovery techniques like Latent Dirichlet Allocation are omitted); we chose to only include a small number of relevant types.

Table 5. Document Metadata Vocabulary Types

Vocabulary Type	Controlled Vocabulary
Created By	Professionals (e.g. librarians)
Structure	Highly structured/controlled, explicit relationships defined
Maintenance	Least frequently maintained
Vocabulary Type	Index Terms
Created By	Authors
Structure	Key words or phrases, without explicit relationships defined
Maintenance	Maintained at the frequency of new publications
Vocabulary Type	Tags/Labels
Created By	Users/Readers
Structure	Key words or phrases, without explicit relationships defined
Maintenance	Most frequently maintained (at the rate of readers who tag content)

A controlled vocabulary is a managed set of terms used to describe concepts in documents. The National Information Standards Organization states that the “primary

purpose of vocabulary control is to achieve consistency in the description of content objects and to facilitate retrieval” [35]. Developing a controlled vocabulary involves identifying the canonical terms to describe concepts in the domain and then determining related terms and relationships between terms. Examples of relationships between terms are “USE”/“Used For (UF)” to indicate synonyms, “Broader Term (BT)” to indicate generalization, and “Narrower Term (NT)” to indicate specialization. The controlled vocabulary is used at index and retrieval time, improving the precision of results returned by ensuring that the terms searched for resolve to their related forms in the controlled vocabulary. The vocabulary is maintained by an expert group, which periodically reviews and updates the vocabulary.

Two prominent examples of controlled vocabularies are the Library of Congress Subject Headings (LCSH) [36] and the National Library of Medicine (NLM) Medical Subject Headings (MeSH) [37]. LCSH “provides the most comprehensive list of subject headings in the world ... [it’s] the key to accurate cataloging and topical searching” [36]. It is accepted as a worldwide standard for subject headings and includes all of the relationship types mentioned previously (USE, UF, BT, and NT). Similarly, MeSH is a cornerstone vocabulary in the realm of health sciences and is used to index Medline/PubMED, the largest collection of publications from biomedical journals. It includes both broad and specific headings, e.g., “Anatomy” versus “Ankle”, and is maintained by experts on staff with NLM.

Reader-provided keywords, or “tags”, are the most frequently updated source of vocabulary terms but are also the least structured. Systems that support this type of vocabulary (e.g., Delicious [38]) allow viewing tags for a given document, providing

concepts that a user found relevant to that document, or viewing all documents for a given tag, providing groupings of documents. Yet, there are no explicit relationships defined between tags, nor are there any attempts to ensure consistency within groups or resolve ambiguity between term usages like “bank” for riverbank versus “bank” for a financial institution.

To describe areas of expertise, our system builds a vocabulary of topics by combining together author-provided key phrases, taxonomy nodes, and descriptive terms extracted from document clusters. Our vocabulary type blends aspects of the previous two vocabulary types. The topics in the vocabulary are of quality closer to a controlled vocabulary than to user-provided tags, in that we include taxonomy nodes created by domain experts and index terms created by the authoring researchers in our set of topics. The frequency of updates to the vocabulary is closer to that of reader provided keywords, in that any new publication can provide any set of index terms to describe itself. However, author-provided topics do not have some of the advantages of the other approaches, such as the lack of ambiguity of the controlled vocabulary or the possibility for frequent updates to the terms associated with the publications like user-generated tags offers. Regardless, by using author-provided key phrases and taxonomy nodes we build a vocabulary of topics from the viewpoint of the researcher, which is the viewpoint we desire to be able to identify expertise and experts.

Clustering. One of our core contributions involves clustering documents to synthesize expertise areas from the terms that describe each cluster. Clustering is a data mining technique to partition items to a set of clusters, where each item is more similar to the other items within its assigned cluster than it is to items in other clusters (see Tan,

Steinbach, and Kumar’s data mining book [39] for a good review of cluster analysis).

Clustering techniques require defining a way of calculating similarity or distances between items and determining when to merge or split a group of items. Each technique is generally a variation on the pattern of repeated groupings of similar items until some criterion function is optimized or a convergence criterion is met.

Clustering algorithms can be grouped into three classes: non-hierarchical, hierarchical, and spectral. Non-hierarchical approaches involve successive repartitioning of the space using criteria like distance to a cluster’s centroid. Hierarchical approaches are either divisive, or top-down, or agglomerative, or bottom-up. Hierarchical algorithms produce a tree-based hierarchy of the input items, which can be converted to non-hierarchical by choosing a level at which to “cut” the hierarchy into clusters of connected components. Spectral clustering leverages the eigenstructure of a similarity matrix to produce its clusters. Each class of clustering has its advantages and disadvantages, and achieving desired results requires being aware of these before choosing an algorithm to apply.

Guided by Ying’s and Karypis’s findings [40], we chose to focus on partitional methods rather than agglomerative or spectral methods. Their approach produces a hierarchy by performing repeated bisections of the data into clusters, optimizing a particular criterion function at each step. The result is a hierarchy built top-down and consisting of n clusters, where n is the number of documents in the data set. We use partitional methods to achieve a hierarchical clustering of documents, but then we convert the hierarchy into a flat clustering by cutting off the tree at a certain level.

Various criterion functions can be used to decide how to bisect a cluster in partitional approaches to hierarchical clustering. Ying and Karypis [40] suggest four main types of functions: *internal*, *external*, *graph-based*, and *hybrid*. Their results show that “the repeated bisection method with the I_2 criterion function leads to the best solution for most of the datasets”. I_2 is an internal criterion function that optimizes the within-cluster similarities, but does not consider differences between members of separate clusters like the external functions do. I_2 maximizes the similarity between a given document and the centroid vector of the cluster that it is assigned to. As discussed in Section 2.3, we use the cosine similarity to calculate similarity, so I_2 becomes a maximization of the following function, with k = number of clusters, S_r = cluster r , d_i = document i in cluster S_r , and C_r = the centroid vector of documents in cluster r :

$$I_2 = \sum_{r=1}^k \sum_{d_i \in S_r} \cos(d_i, C_r)$$

Partitional clustering using the I_2 criterion function allows us to easily and effectively produce clusters of related documents. By leveraging this technique, we are able to cluster documents into related subtopics. By extracting descriptive terms from each cluster, a set of topics can be built that complements those taken from the author-provided key phrases and taxonomy nodes. The clusters also provide a way for us to identify related topics and subtopics, enhancing our results when finding experts for a given topic.

3.3 Approach

In this section, we present our approach to determining expertise areas for a given researcher and experts for a given topic. We identify our sources of data, discuss our data

model, and review the techniques that we apply to extract topics from those sources. Then, we detail the specifics of our approach to determine expertise areas and topics experts.

3.3.1 Expertise Sources

Our approach uses DBLP [4, 41] as its main source of expertise information and also includes additional information available on the following publishers' websites: ACM [42], SpringerLink [43], and IEEE [44, 45]. DBLP is maintained by Michael Ley of the University of Trier in Germany. It originally focused on publications relating to "DataBases and Logic Programming", but has since expanded to incorporate publications in many other areas of computer science. Given this, the acronym DBLP can be interpreted as "Digital Bibliography and Library Project" [46].

The DBLP data set is provided as a periodic XML data dump of their main database [47]. A DTD describing the XML format is provided along side the dump. We obtained the February 17th, 2008 dblp.xml.gz data dump and DTD for this work. The ACM SIGMOD Anthology [48] is a parallel collection to DBLP, containing six volumes of database systems research publications. It contains not only metadata but also scans of the physical copies of many of the publications. Additionally, many of the entries in DBLP contain references to their electronic editions, for example URLs to ACM or Springer or IEEE. These electronic editions expand on the information in DBLP, often including abstracts, keywords, and subject areas. After evaluating the available resources, we opted to use DBLP and the associated electronic editions accessible on the Internet. We chose not to leverage the full text of publications that the ACM SIGMOD Anthology offers, due to the difficulty of applying OCR to scanned publications (in

contrast to Bergmark, Phempoonpanich and Zhao’s work [49], which does discuss extracting information from publication PDFs).

The DBLP data set consists of the eight types of entries (Table 6). The majority of entries (97.4%) are either conference or journal publications. Each entry contains one or more of the following metadata fields: *author*, *editor*, *title*, *booktitle*, *pages*, *year*, *address*, *journal*, *volume*, *number*, *month*, *url*, *ee*, *cdrom*, *cite*, *publisher*, *note*, *crossref*, *isbn*, *series*, *school*, *chapter*. An example of a DBLP record in BibTeX format is included in Figure 5. The most useful fields for our purposes are author, title, and ee, or a reference to the electronic edition of the publication, for example the ACM Digital Library entry for a publication.

Table 6. DBLP Entry Types

<i>Entry Type</i>	<i>Additional Description</i>	<i>Count</i>
Article	Journal article	351058
Book		1244
In Collection	Publication cited in a collection, e.g., a book	2569
In Proceedings	Publication published in conference proceedings	589207
Master’s Thesis		8
Ph.D. Thesis		89
Proceedings	Conference proceedings	9543
WWW	Author homepage links (and 39 other WWW links)	11389
Total:		965107

DBLP Record 'journals/vldb/TaninHS07'


BibTeX

```
@article{DBLP:journals/vldb/TaninHS07,
  author    = {Egemen Tanin and
               Aaron Harwood and
               Hanan Samet},
  title     = {Using a distributed quadtree index in peer-to-peer networks},
  journal   = {VLDB J.},
  volume    = {16},
  number    = {2},
  year      = {2007},
  pages     = {165-178},
  ee        = {http://dx.doi.org/10.1007/s00778-005-0001-y},
  bibsource = {DBLP, http://dblp.uni-trier.de}
}
```

Figure 5. Example DBLP Record

To augment the information in the DBLP data set, we leverage the publication information, linked to DBLP entries through the ee field, provided on the public websites for the ACM Digital Library [42], SpringerLink [43], IEEE Xplore [45], and IEEE Computer Society [44]. These sites provide abstracts for the majority of their publications. ACM provides author-specified keywords and subject areas, which are nodes in the ACM 1998 Computing Classification System [50]. An example entry from ACM is shown in (Figure 6). The other three sites contain author-provided index terms for publications, similar to ACM's keywords. Though not every publication has its abstract or keywords/index terms available, the coverage is sufficient for our purposes.

Scalable network distance browsing in spatial databases

Full text  Pdf (636 KB)

Source International Conference on Management of Data [archive](#)
 Proceedings of the 2008 ACM SIGMOD international conference on Management of data [table of contents](#)

Authors [Hanan Samet](#) University of Maryland, College Park, MD, USA
[Jaqan Sankaranarayanan](#) University of Maryland, College Park, MD, USA
[Houman Alborzi](#) University of Maryland, College Park, MD, USA

↑ **ABSTRACT**

An algorithm is presented for finding the k nearest neighbors in a spatial network in a best-first manner * * * shortest paths between all possible vertices in the network and then making use of an encoding that ta * * *

↑ **INDEX TERMS**

Primary Classification:
 H. [Information Systems](#)
 ↳ H.2 [DATABASE MANAGEMENT](#)
 ↳ H.2.8 [Database applications](#)
 ↳ **Subjects:** [Spatial databases and GIS](#)

Keywords:
[decoupling](#), [nearest neighbor](#), [scalability](#), [shortest path quadtree](#), [spatial networks](#)

Figure 6. Example Publication Entry from the ACM Portal

3.3.2 Data Model

In order to build a system that identifies experts and topics, we need to precisely define the models for those types. Our expertise sources provide defined data types from which we can build our two core types: Author and Topic. These types incorporate our definition of expertise, with indicators for frequency and recentness. Indicators for relevancy are not explicitly stored, as the topics extracted are restricted to only relevant ones.

DBLP distinguishes between eight different types of entries (Figure 7) and provides a varying number of metadata fields for each (see section 3.3.1). We unify those different types into one general model: Publication (Figure 8). Data external to DBLP, such as abstracts, keywords, or category/taxonomy information, is represented by our

Content type and related types (Figure 9). Content is composed of an abstract, zero or more categories, and zero or more keywords/phrases. The model for Category data includes the name of the category, whether or not it is active, and a link to its parent so that the full taxonomy tree can be represented.

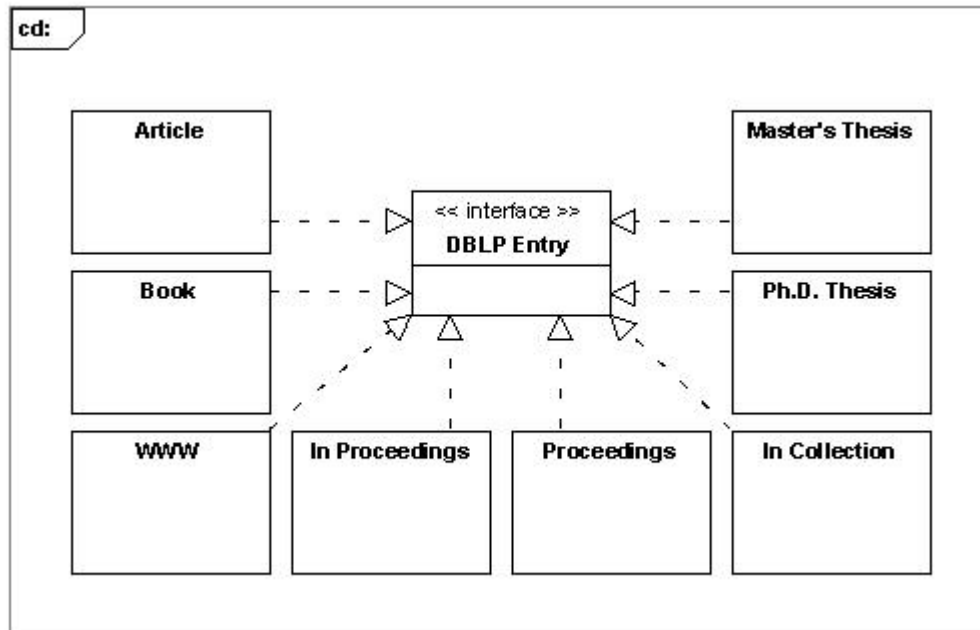


Figure 7. DBLP Entry Types

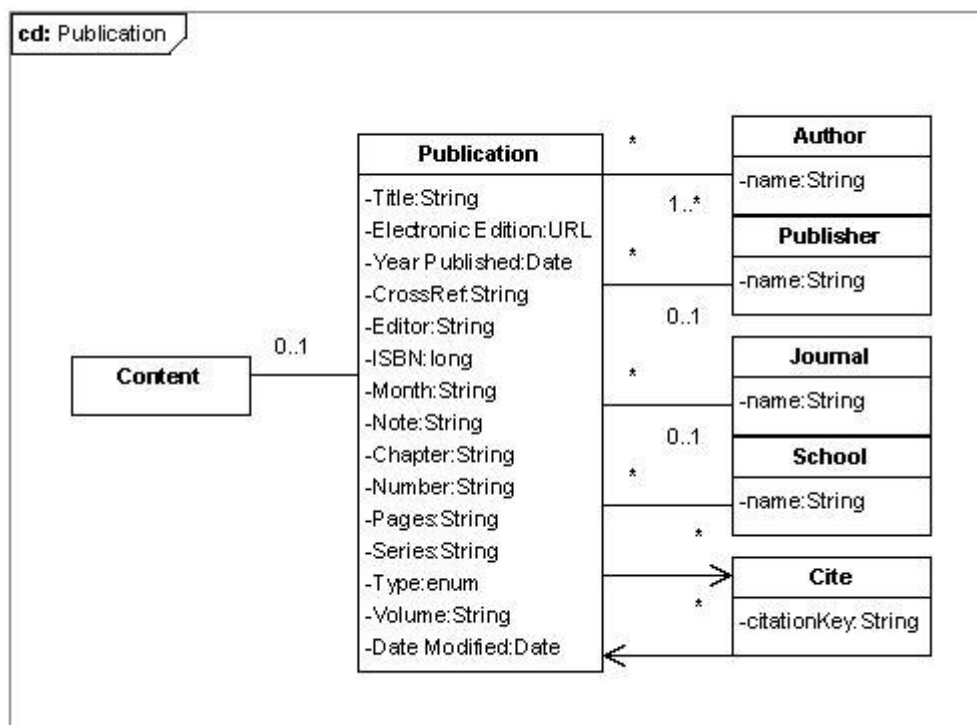


Figure 8. Publication Data Model

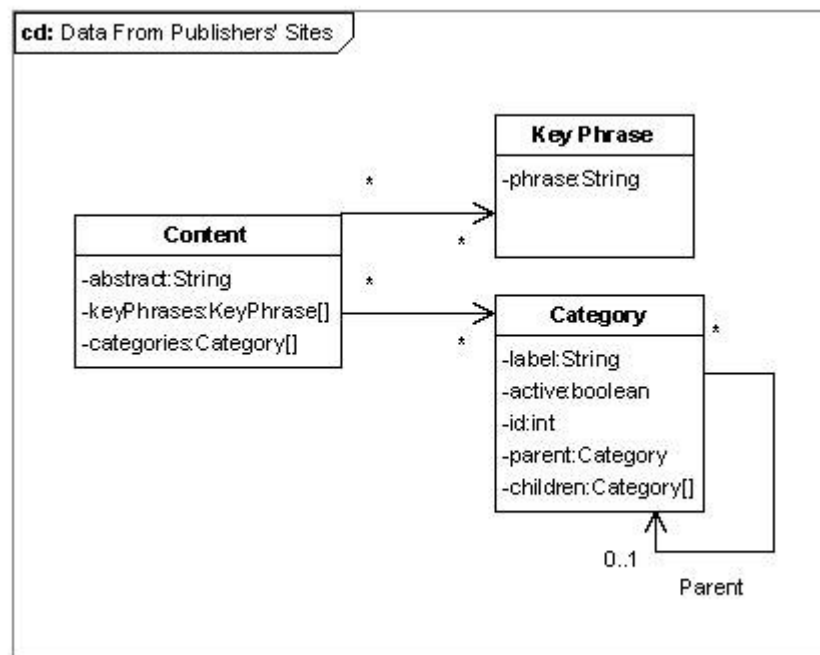


Figure 9. Publication Content Model

Our approach to expert finding requires us to develop some custom data types that extend the information provided by our expertise sources. We introduce an Expertise Profile type to explicitly hold the association between Authors and Topics. We also develop a Document Cluster type to represent clusters that we use in our approach for determining sub-topics and related topics. Finding experts for a given topic does not require any additional types, as Topics and Authors will suffice. Our final model, including attributes and operations on our types, is shown in (Figure 10). The following two sections provide concrete examples and further details on how we arrived at this model.

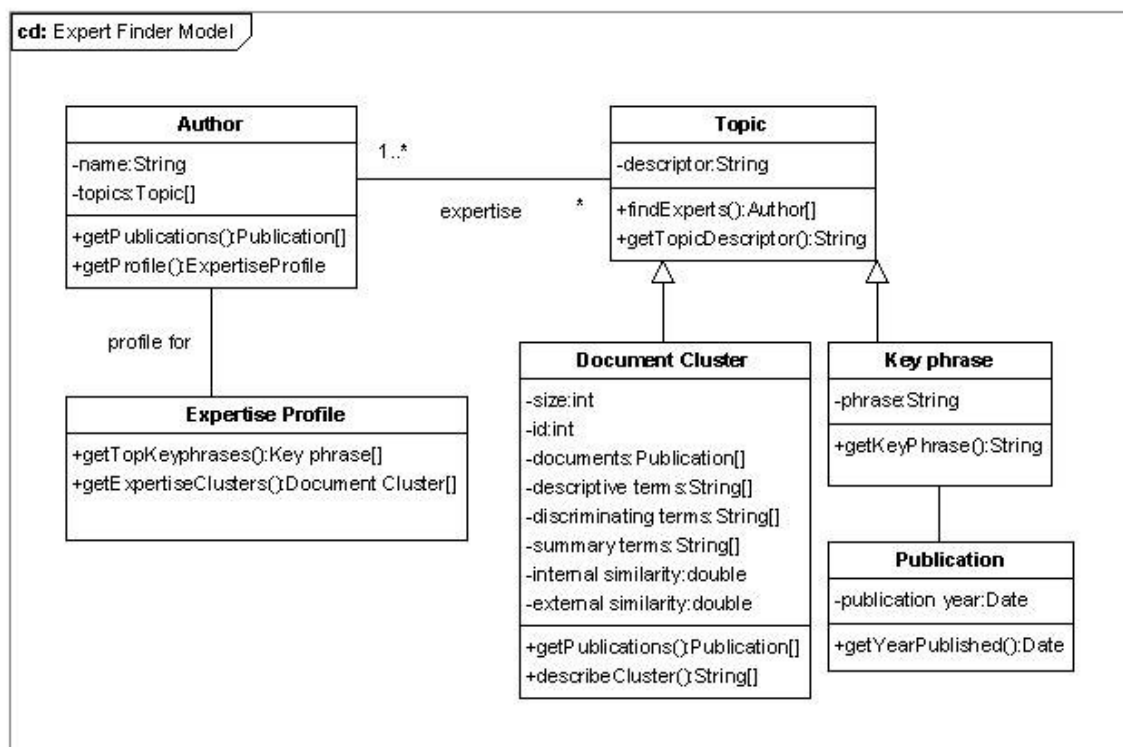


Figure 10. Expert Finder System Data Types

3.3.3 Retrieving Expertise Areas

We present our approach to retrieving expertise areas for a given researcher as a progression of approaches, in order of increasing complexity. The retrieval goal focused on here is the “what” part of our “who knows what” motivating question. We will use Professor Hanan Samet, a recognized authority on spatial databases, as the “who” in this section.

The simplest approach to identifying expertise areas involves asking researchers to identify their areas of expertise. This ensures high precision, or correctness of the results, as the researchers will not identify any areas that they were not versed in. Expert finder sites like Community of Science [28] rely on this self-identification, but many organizations like the MITRE Corporation [51] have discovered that this approach introduces unnecessary overhead to maintain the profiles. Information becomes quickly outdated and recall suffers, i.e., a low number of existing relevant results are actually returned. However, self-reported expertise areas on authors’ homepages (e.g., Figure 11) offer value as a ground truth for evaluating the precision of automatically identified areas.



Hanan Samet

Email: hjs@cs.umd.edu

Academic Degree: Ph.D., [Stanford University](#), 1975

•
•
•

Research Interests:

[Spatial Databases](#), [Data Structures](#), [Geographic Information Systems](#),
[Image Databases](#), [Computer Vision](#), and [more](#)



Browse the publications of Hanan Samet by subject classifications			
book	code optimization	compiler testing	computer-aided manufacturing
computer graphics	computer vision	digital government	equality algorithms
general	geographic information systems (GIS)	image approximation and compression	image database
•	•	•	•

Figure 11. Hanan Samet's Self-Reported Research Areas

Our first improvement on the self-reporting approach addresses the problem of data staleness. Researcher-created profiles tend to become quickly outdated and rely on a manual process to keep them current. We need a source that stays current and accurate. For this we utilize the public products of a researcher's research: publications. By using the refereed publications in DBLP as a source of information, we have access to a recent snapshot of a given researcher's activities.

An up to date snapshot of a researcher's publications provides implicit information on his or her expertise areas. Extracting explicit topics that represent expertise areas is considerably more challenging than the focus of our first improvement, obtaining up to date data. Our approach to the problem of identifying topics is based upon three key ideas:

1. Author-provided key phrases and taxonomy nodes for their publications provide a source of topics for that publication.
2. A vocabulary of relevant topics can be built from the set of all author-provided key phrases and can be applied to the entire data set to identify topics for all publications, not just those with author-provided key phrases and taxonomy nodes.
3. Document clustering and the resulting representative terms for each cluster can provide more specific topics to complement the topic vocabulary.

The next improvement in our approach incorporates key idea #1: author-provided key phrases and taxonomy nodes. These improve on the self-reporting approach by not requiring researchers to update a separate list of expertise areas. Systems like the ACM Author Profile Pages [52] incorporate this information as a part of a researcher's profile (e.g., Figure 12). The taxonomy nodes, for example "H.2.8 Database Applications – Spatial databases and GIS", provide unambiguous topics that can be used as expertise area for a researcher. In contrast, the quality of results produced by the author-provided key phrases is highly dependent on the author providing consistently good key phrases in their publications. Additionally, author-supplied keywords rely on an author's view of the important keywords relating to their publication. Their choice of keywords may be too vague or may miss key topics discussed in their paper.

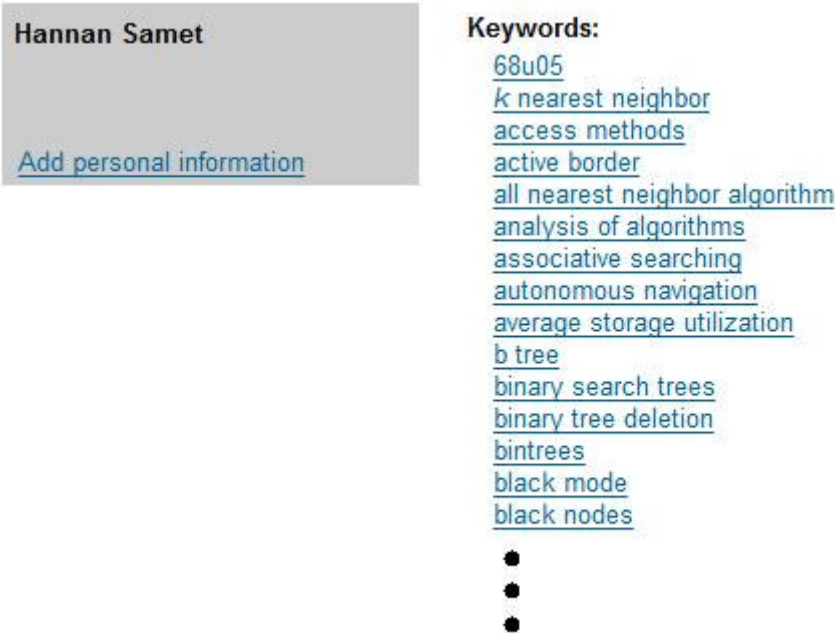


Figure 12. Keywords from Hanan Samet's ACM Author Profile Page

To maximize the advantages of author-provided key phrases, we build a vocabulary of topics from all provided key phrases, in line with key idea #2. Our algorithms for doing this are detailed in Table 7.

Table 7. Build Topic Vocabulary Algorithm
<i>Algorithm for building the topic vocabulary</i>
1 buildTopicVocabulary(publications) {
2 vocab = create empty vocabulary
3 // initial set = all author-provide key phrases
4 for pub in publications {
5 if (pub has key phrases from publisher site) // ACM, SpringerLink, IEEE
6 add all pub.keyphrases to vocab
7 }
8 // all author-provided key phrases include informative topics *and*
9 // low information keywords. So, prune the set of topics in vocab.
10 prunedVocab = removeLowInformationKeyphrases(vocab)
11 return prunedVocab
12 }

The algorithm in Table 8 applies a variety of heuristics to the set of key phrases to reduce the topic vocabulary down and eliminate low value phrases. To better illustrate how this is done, see the descriptions and examples shown in Table 9.

Table 8. Key Phrases Pruning Algorithm

<i>Algorithm for removing low information key phrases</i>	
1	removeLowInformationKeyphrases(vocabulary) {
2	prunedVocab = create empty vocabulary
3	
4	for phrase in vocabulary {
5	if (phrase is an acronym) {
6	phrase = expandAcronym(phrase)
7	}
8	
9	tokens = split phrase on whitespace
10	tokenCount = number of tokens
11	stemmedPhrase = stem tokens and re-join together into a phrase
12	phraseIdf = 0 // default value
13	if (tokenCount == 1) {
14	phraseIdf = retrieve IDF for single term from search index
15	}
16	
17	// Note: these heuristics for determining low information keywords were
18	// determined by experimentation. For a phrase to be added to
19	// the vocabulary, it must not satisfy any of the pruning criteria.
20	if (stemmedPhrase not already in vocabulary) {
21	if (tokenCount >= 2 OR
22	second letter of phrase is not capitalized OR
23	phrase is all capitalized letters OR
24	(tokenCount == 1 AND phraseIdf > MIN_IDF_THRESHOLD)) {
25	add phrase to vocabulary
26	}
27	}
28	}
29	
30	return prunedVocab
31	}

Table 9. Vocabulary Pruning Rules

Rule	Examples
Interesting phrases are typically of 2 or more tokens	spatial databases, image databases
Interesting phrases typically are capitalized words, or have their second letter not lower-cased	SAND, kNN, r-trees
Acronyms and their expanded forms should be considered equivalent. Acronyms are typically listed as “expanded form (acronym)” and can be parsed easily as such	OLAP, Geographic Information Systems (GIS)
Stemmed forms of the words are considered as equivalent	database = databases, access structures = access structure
Interesting words not picked up by prior rules can be detected using IDF as an indicator of importance.	steganalysis, steganography

After applying these rules to reduce the key phrases down to a canonical set, we use the Aho-Corasick string-matching algorithm [53] to identify phrase instances from this vocabulary in all publication’s titles and abstracts. This algorithm uses a finite state machine that can be built in $O(|k|)$, with k =set of keywords and $|k|$ indicating the length of the entire set. Then, it can identify instances of those keywords in input text in a single pass. However, its keyword recognition step is output-sensitive, being linear in the number of output chunks. Identified vocabulary phrases in a publication are added to that publication’s list of key phrases. See Table 10 for an example of applying this to a publication’s title to extract key phrases, where the publication had no author-provided key phrases prior to running the extraction algorithm. This completes the intent of key idea #2, enabling us to have topics for nearly every document in the collection.

Table 10. Example of Applying Topic Extraction to a Title

Title	Clustering moving objects via medoid clusterings
Authors	Hans-Peter Kriegel, Martin Pfeifle
Extracted Key Phrases	Clustering, Moving Objects

Our final improvement in our approach relates to key idea #3: document clustering to provide additional information. Sometimes, key phrases are not enough to accurately explain the context of a given publication, especially for general key phrases like “databases” or “data mining”. Clustering the documents elicits sub-groups of topics. These clusters can help summarize the range of topics contained in a document set. By extracting descriptive features, such as those terms that are more likely to occur in a given cluster than in other clusters, a human readable description can be generated for each cluster. Co-occurrence of descriptive terms can provide an easier to understand term grouping than just displaying each term separately. For example, a cluster described by “spatial, database, query, index, improvement” is more easily understood when presented as term groups: “spatial index improvement”, “query spatial database”.

In summary, we offer three distinct advantages over the naïve self-reporting approach. First, we automatically build a researcher’s expertise profile from their publications and associated metadata, e.g., author-provided key phrases and taxonomy nodes. Second, we leverage all author-provided keywords to build a topic vocabulary for the domain, and then we apply this topic vocabulary to all publications to extract additional expertise areas. Lastly, we utilize document clustering to distinguish specific sub-groups from the overall general topics in a set of documents.

3.3.4 Retrieving Experts

Given a topic, who are the experts on it? Who was first in the field and who is still active in the field? What are their expertise areas within that topic? Answering these questions will enable identification of experts on a given topic while also providing context to distinguish the levels of expertise. In our chapter introduction, we specified

three criteria for identifying and ranking experts on a given topic: relevancy, frequency, and recentness. We also noted that our approach introduces contextual views of the identified experts, providing richer information regarding expertise in the topic than a simple publication count. We present our approach as a progression of techniques, in order of increasing complexity.

The simplest approach to identifying experts is to retrieve all experts who have published anything on a given topic (see algorithms in Table 11 and Table 12). Ordering them by publication count descending will produce a ranking of authors for that topic. This approach minimally satisfies two of our three criteria: relevancy and frequency. It ensures relevancy at a broad level, as the only authors included are those that have mentioned the topic. It satisfies our frequency criterion because it utilizes publication count. But, it does not provide any context around the author's mention of the topic, for example by providing related topics or subtopics. A user of an expert finder system that uses this approach will be left with a high level list of frequent authors, but will be unable to answer questions like "who were the pioneers in this field".

Table 11. Find Relevant Publications Algorithm

<i>Algorithm for finding relevant publications</i>	
1	// identify relevant publications for the input topic
2	findRelevantPublications(Topic topic) {
3	searchFields = title, abstract, keyphrases
4	topicPublications = full text search for topic across searchFields in all documents
5	return topicPublications
6	}

Table 12. Most Frequent Authors Algorithm

<i>Algorithm for finding experts - ranked by publication frequencies</i>	
1	// frequency of publications ranking algorithm
2	identifyExpertsFor(Topic topic) {
3	publications = findRelevantPublications(topic)
4	Map<author, pubCount> authorPubs = empty map of authors to ints, default = 0
5	for (publication in publications) {
6	for (author in publication.authors) {
7	// increment author publication count
8	authorPubs[author]++
9	}
10	}
11	rankedAuthors = sort authorPubs by pubCount descending
12	return rankedAuthors
13	}

Each publication includes temporal information that can be used to improve upon the previously described approach to finding experts. By examining all publications on a given topic in order of when they were published, we can chronologically order the researchers and extract metrics related to their contributions. For example, we can update our “most frequent authors” view to be for recent publications, enabling a view into who is currently active in the field. Additionally, this chronology enables us to identify the set of early pioneering researchers on a given topic (see algorithm in Table 13), highlighting those who were “first in field”. These distinguishing characteristics provide a way to classify researchers as *pioneers*, while also incorporating recentness into a view on the most frequent authors.

Table 13. First in Field Algorithm***Algorithm for finding “first in field” authors***

```

1 // identify earliest authors to publish on the input topic
2 firstInField(Topic topic) {
3   publications = findRelevantPublications(topic)
4   totalPubCount = count(publications)
5   sortedPublications = sort publications by date ascending
6
7   // customizable percentage to include, set to 10% for easy
8   // evaluation purposes
9   firstInFieldPercentage = 10%
10  firstInFieldAuthors = empty set of authors
11
12  // loop through pubs in ascending order, calculating
13  // cumulative percentage and including authors up to the
14  // year at which firstInFieldPercentage of the pubs in
15  // the field have been published
16  count = 0
17  for (Publication pub in sortedPublications) {
18    count++
19    cumulativePercentage = count / totalPubCount
20    if (cumulativePercentage >= firstInFieldPercentage
21        && cutoffYear not set) {
22      cutoffYear = pub.getYear()
23    }
24    if (cutoffYear not set || pub.getYear() <= cutoffYear) {
25      firstInFieldAuthors.addAll(pub.authors)
26    } else {
27      break
28    }
29  }
30
31  // sort the first in field authors by publication frequency
32  // descending, to support only showing the top most frequent
33  // first in field authors
34  sortedAuthors = sort firstInfieldAuthors by frequency desc
35  return sortedAuthors
36 }

```

Our approach so far has identified the most frequent authors, chronologically ordered them, identified *pioneers*, and incorporated a concept of recentness. Our final improvement to the approach enhances the relevancy of the results by providing additional context around the identified experts for a topic. We leverage the clustering techniques described in section 3.3.3 to produce subtopics and related topics from the data set. The terms describing the clusters can be associated with the identified relevant authors to provide context around their contributions. For example, a top author on “spatial databases” may be in a cluster described as “moving objects”, which helps to clarify that author’s area of contribution in the field of spatial databases. With this additional information, we improve the relevancy of our results by presenting extra context around each researcher.

In summary, our approach to identifying experts for a given topic could be characterized as reducing the haystack, rather than finding the needle in the haystack. We do not provide an approach that produces the needle, i.e., a single ordered ranking of researchers for the topic. Rather, we provide multiple views on researchers in that topic, enabling the user of the system to answer more questions than the simple one of “who is an expert on topic x”. The recent top authors view enables answering the question of “who is active in topic x”, while the *pioneers* view can answer “who was first in the field of topic x”. The subtopics view, which results from clustering the results, establishes context around researchers that helps answer the question of “what area within topic x has this researcher been active in”. Together, these views summarize different types of experts for a given topic.

3.4 Implementation

We implemented a prototype system called DEF-CAT (DBLP Expert Finder utilizing Categories and Topics) for the aforementioned approaches to expert and expertise finding. In this section, we detail the specifics of the implementation, provide examples of output from the system, and then evaluate the implementation.

3.4.1 Prototype

A graphical overview of our prototype system is provided in Figure 13. The details regarding the core components are provided following that figure. The code for the prototype is hosted on Google Code [54], and is around 25,000 lines of code (25 KLOC).

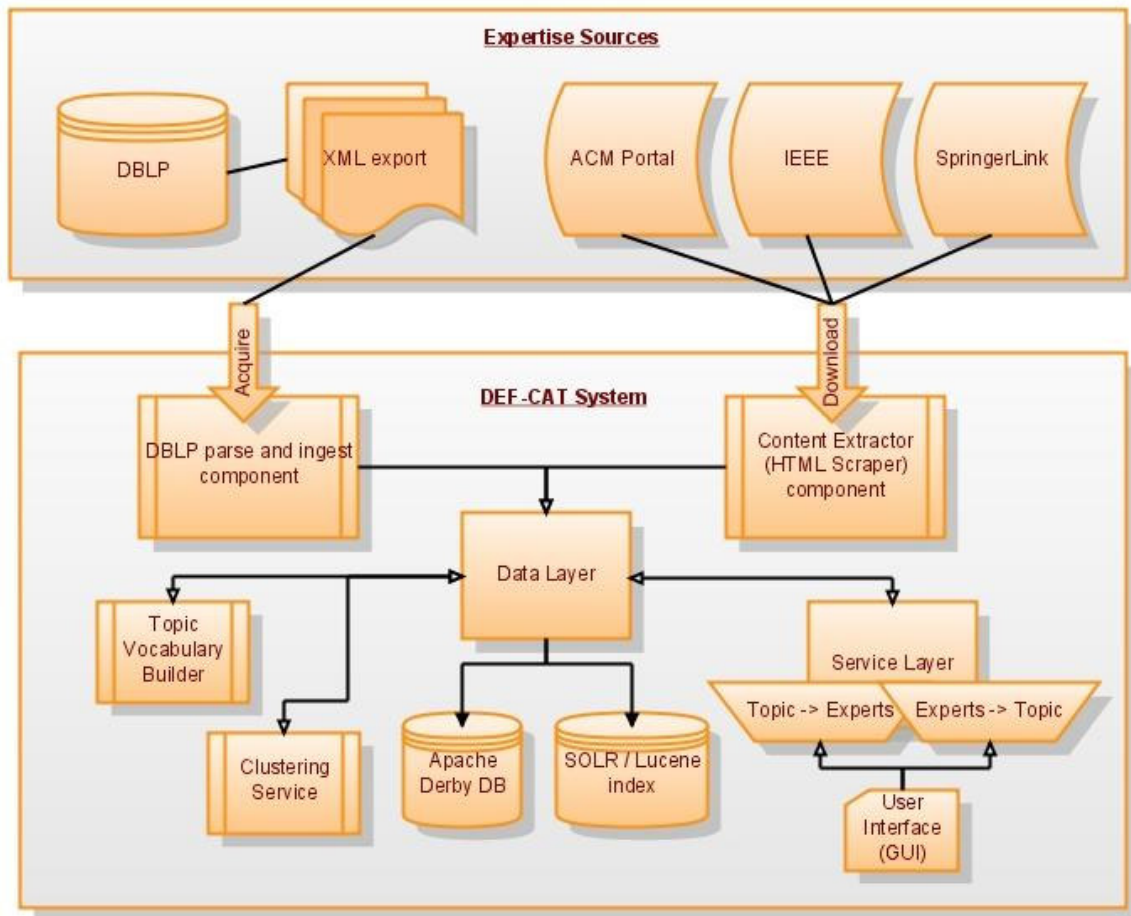


Figure 13. DEF-CAT System Overview Diagram

Data Parse and Ingest. The DBLP data set was ingested from its XML format into a customized schema (Figure 14, Figure 15) in an Apache Derby [55] embedded database. The DTD was converted to an XSD and then JAXB technology [56] was leveraged to unmarshall the XML into an object model. The object model was annotated with JPA [57] annotations to produce an object-to-relational mapping that allowed for easy persistence into a relational DB.

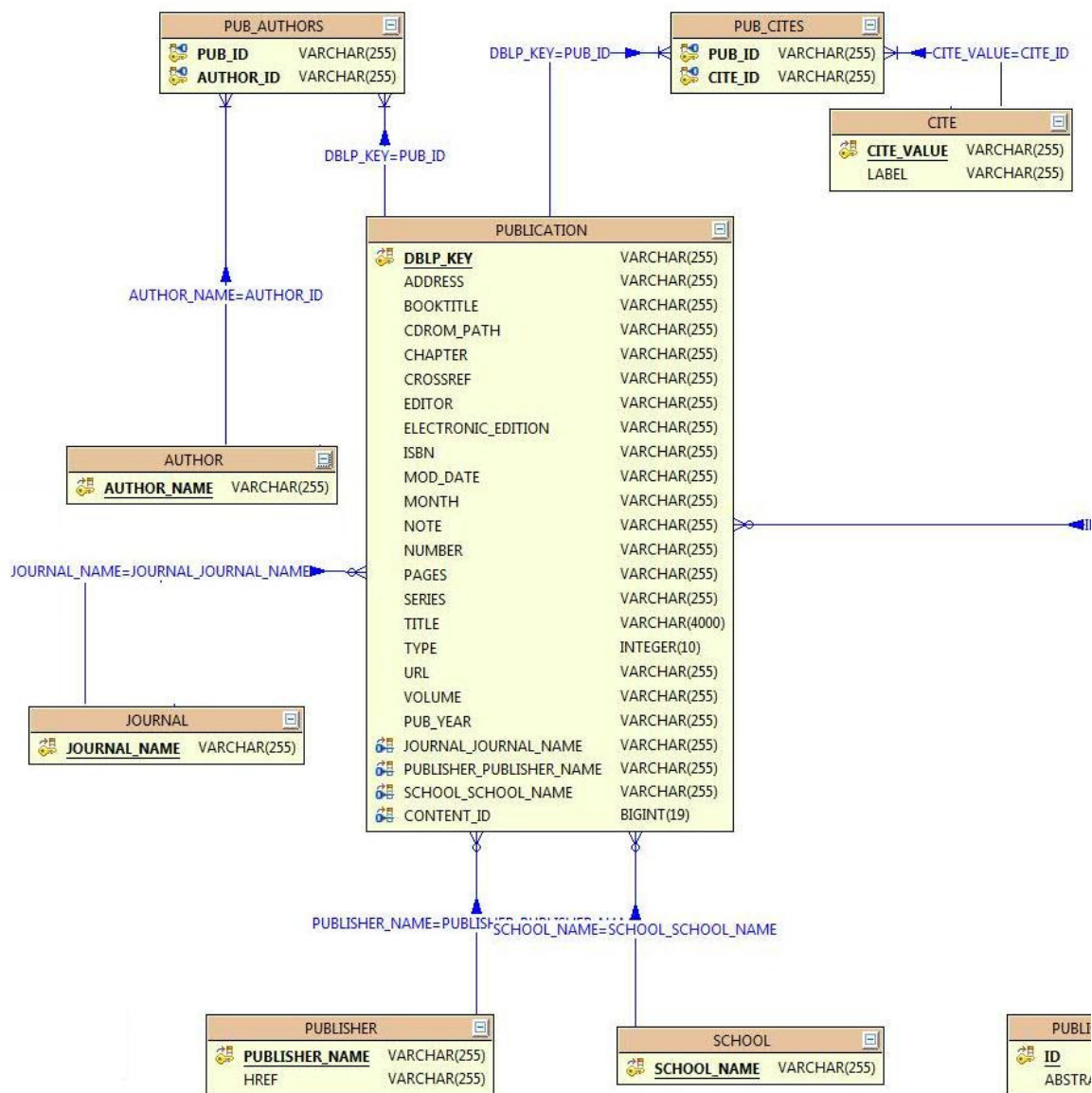


Figure 14. Publication Data Model

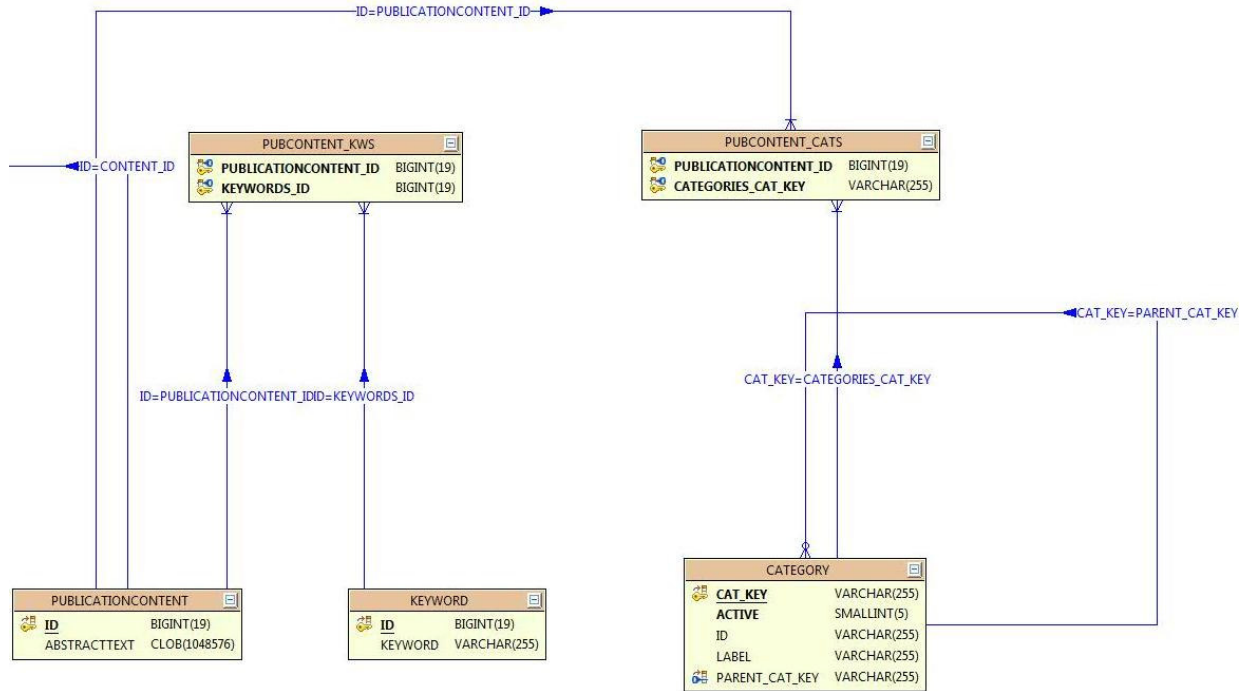


Figure 15. Content Type and Related Data Types

Content Download and Extraction. A web crawler downloaded local cached copies of the electronic edition of a publication, which were then accessed, parsed as HTML, and then scraped to extract the abstracts and keywords. Scrapers were implemented for ACM, IEEE, and Springer, enabling retrieval of abstracts and/or keywords for close to 292,000 entries (~30% of the total document collection).

Search Index. A full text search index was built from the document set by utilizing Solr/Lucene [58]. The schema for that index mirrored the data model (Figure 14, Figure 15), but it also included a catch all field for all text fields (titles, abstracts, key phrases). This field was tokenized and stemmed to enable easy querying for all documents in the collection relating to given term(s).

Identifying Topics in Publications. We use the author-provided key phrases as a set of topics that we could look for in publications without author-provided key phrases or taxonomy nodes. We used the LingPipe [59] exact dictionary chunker, which

implements the Aho-Corasick string matching algorithm, to identify instances of these key phrases in publications.

Data Clustering. We utilized the CLUTO clustering toolkit, which is a C-based tool that is easily used when run from the command line or from the gCLUTO tool [60]. To access the results from our Java program, we wrote a custom solution file reader to translate the clustering results into Java. We experimented with a variety of types of clustering, such as agglomerative, partitional using repeated bisections, and graph-based. We also experimented with a variety of parameter settings, such as the number of clusters and what criterion functions to use. The best results, determined by repeated trials and manual review of the output, were achieved using partitional clustering via repeated bisections with I_2 as the criterion function. Our strategy to determine the number of clusters started with setting a minimum internal similarity for all clusters and a minimum overall number of clusters. Then, we ran repeated applications of the clustering algorithm, starting with the minimum overall number of clusters and then increasing by one until the minimum internal similarity criterion was met.

Topics -> Experts and Experts -> Topics. Using the data layer components and the data model from Figure 10, we implemented services to identify experts for a given topic and to produce an expertise profile for a given author. Topic experts are presented in three non-exclusive groups: most frequent, most recent and frequent, and first in the field (Figure 16). An expertise profile for a researcher includes their most frequent topics, i.e., taxonomy nodes or key phrases, and the subtopics/descriptive terms for clusters of their publications (Figure 17).

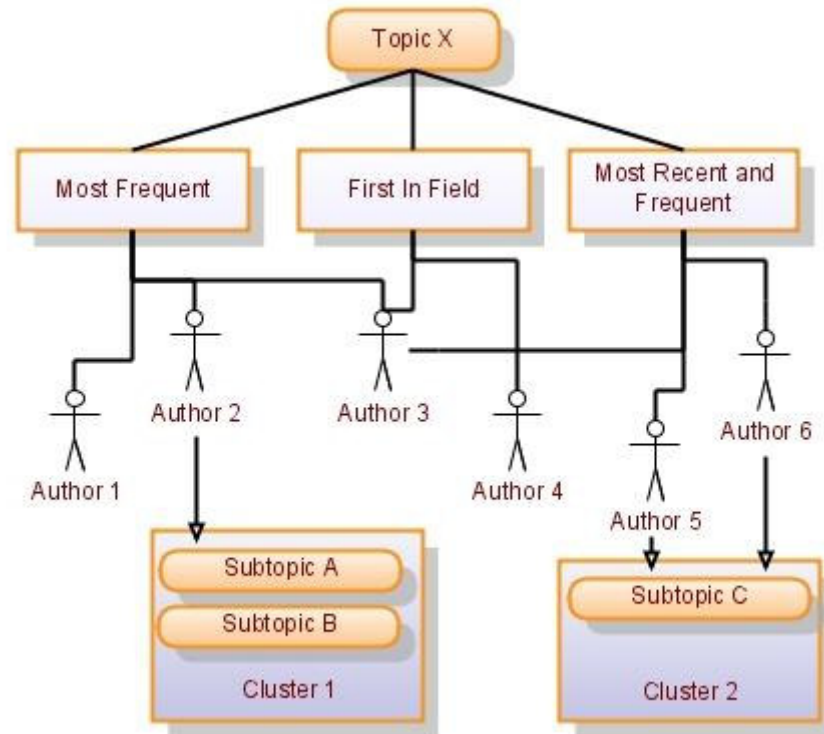


Figure 16. Topic Experts Diagram

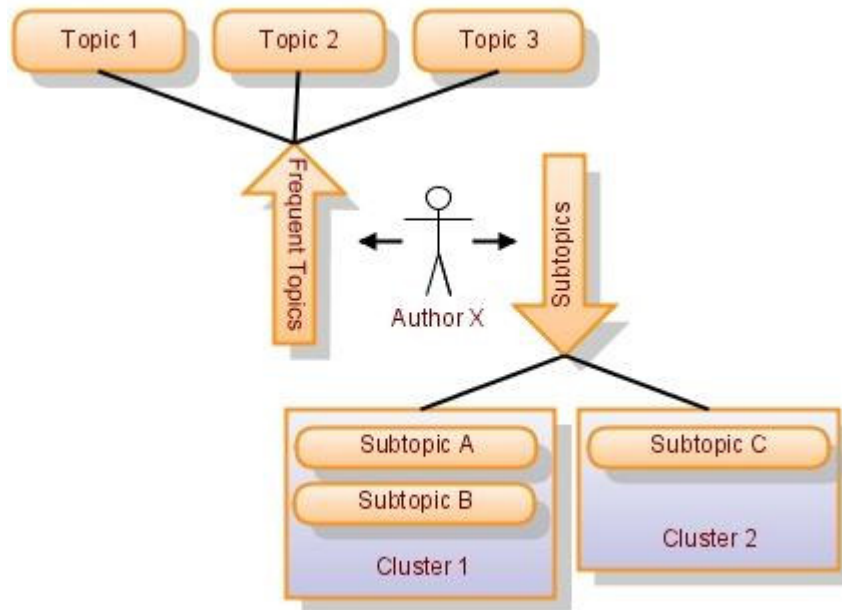


Figure 17. Expertise Profile Diagram

User Interface. We developed a web-based graphical user interface (GUI) for browsing the data and identifying expertise areas and experts. For our discussion of the

system and results in this chapter, we use a simple tabular output format from our system rather than the GUI. Information on that GUI and its underlying implementation can be found in Chapter 4.

3.5 Evaluation

A comprehensive evaluation of an expert finder system is difficult to perform without having a ground truth to assess accuracy of the results against. Determining a ground truth of expertise in a given field is equally as challenging, due to the subjective and contentious nature of ranking experts. Inaccuracies are easy to identify, but accurate results are hard to codify. The TREC-2005 Enterprise Track identified a ground truth of experts for assessment purposes of the W3C corpus, enabling quantitative evaluations of proposed approaches to finding experts. No such ground truth exists for our data, so we consider alternate types of evaluations.

To evaluate our system, we consider the following options. Author homepages frequently list areas of research interests that can be equated with expertise areas. They are typically broad in definition and may not be comprehensive, due to being manually maintained. However, they form a basis of topics that should be identified by any expert finder system that contains those researchers in it. Alternatively, researchers in a domain can be called upon to evaluate the accuracy of results retrieved for areas that they are knowledgeable in. They could assess the error rate and provide a subjective evaluation of the correctness and usefulness of the approach. Averaging across their assessments could help to smooth out any bias and quell disagreements between competing researchers over the way that they ranked each expert.

To contain the scope of this thesis, we elect to evaluate our approach by comparing results with self-identified areas on researcher’s homepages, leaving the evaluation by researchers in the domain as future work. We also detail how our system identifies experts in the field of spatial databases, to demonstrate the results produced by our techniques. Additionally, the process of generating results from our system will elicit the advantages and shortcomings of our approach, which we discuss in Section 3.6.

Researcher’s Homepages. To informally evaluate our system’s approach to identifying expertise areas for a given researcher, we compare our results with the self-identified research interests present on an author’s homepage. We use Hanan Samet as our test case, as he is a well known and respected researcher and he has manually extracted subject areas and classified his more than 171 publications into those areas on his homepage [61].

We assess the recall of our system, which is the number of Hanan Samet’s self-identified subject areas that we also identified. We consider our system as having identified an area if an exact match or semantically similar set of terms is present in the top 5 taxonomy nodes, the top 25 key phrases, or the clusters’ descriptive terms. We limit ourselves to this subset of data because that is what users of our system would have access to initially in the research profile view (see Section 4.3). We include 28 of his 34 subject areas, omitting 3 that are not specific enough to compare ourselves to (book, general, survey) and 3 that have none of their publications listed in DBLP (compiler testing, computer-aided manufacturing, similarity searching). The expertise areas for Hanan Samet that our system identified are given in Table 14 and Table 15.

Table 14. Expertise Areas for Hanan Samet #1

Taxonomy Nodes
H.2.8 Database Applications Spatial databases and GIS (14)
H.2.4 Systems <u>Query processing</u> (9)
I.3.5 Computational Geometry and Object Modeling <u>Curve, surface, solid, and object representations</u> (5)
I.3.3 Picture/Image Generation Display algorithms (4)
H.2.8 Database Applications <u>Image databases</u> (3)

Table 15. Expertise Areas for Hanan Samet #2

Clusters	Key Phrases
Cluster 0: label compon connect, octre neighbor (11)	Quadtrees (24), Data Structures (19), Spatial Databases (18), Spatial Data (15), geographic information systems (10), Nearest Neighbor (10), Geographic Information (9), Peer-to-peer (8), Query Processing (7), neighbor finding (6), information systems (6), digital government (6), SAND (6), R-Trees (6), the Internet (5), Data-parallel (5), octrees (5), Hierarchical Data Structures (4), Database Management System (4), spatial data structures (4), data sets (4), Line Segments (4), Peer-to-Peer (P2P) (4), tetrahedral meshes (4), Worldwide (4)
Cluster 1: gi, system multimedia, system geograph inform (13)	
Cluster 2: spatial data, spatial oper, join spatial, parallel data, parallel oper (16)	
Cluster 3: peer applic govern distribut, peer p2p applic distribut (16)	
Cluster 4: spatial sand access, databas spatial access, databas spatial brows (15)	
Cluster 5: quadtre window algorithm, quadtre region algorithm, quadtre region boundari (19)	
Cluster 6: line effici, structur data effici, structur data hierarch (24)	
Cluster 7: anim search, visual anim, similar search, queri visual (18)	
Cluster 8: mesh represent, content retriev, shape (16)	
Cluster 9: csg algorithm, heurist, optim algorithm, tree csg (18)	

Our highest recall is 93%, but that takes into account some filtering down of the subject areas that we include in the evaluation. For the entire 28 subject areas that we considered, our recall is 46%. However, this includes 9 areas that had less than 3 publications in them. If we exclude those areas, our recall is 68%. This includes 5 areas that had only 3 publications in them. If we exclude those areas, only including subject areas with greater than 3 publications, then our recall is 93%. The details on that assessment are given in Table 16. This high recall for a set of the subject areas with a reasonable amount of publications shows that our system is effective at identifying expertise areas for researchers like Hanan Samet. Our recall on the full set of subject areas could be improved if we considered more of the system's output than what is shown in the user interface, e.g., the top 25 key phrases, top 5 taxonomy nodes, or if we expanded our clustering settings to allow for smaller, more cohesive groups at the risk of producing too many groups. The results of this informal evaluation are promising, and future work could investigate more complete evaluations if needed.

Table 16. Evaluation Results for Hanan Samet's Expertise Areas

Author-identified Subject Area	In our results?	Evidence
computer graphics	TRUE	keyphrase: tetrahedral meshes, cluster 8: mesh represent
digital government	TRUE	key phrase: digital government
geographic information systems (GIS)	TRUE	taxonomy: H.2.8 Spatial databases and GIS, key phrase: GIS, cluster 1: GIS
image database	TRUE	taxonomy: H.2.8 Image databases
nearest neighbor finding	TRUE	key phrase: Nearest Neighbor, neighbor finding
parallel processing	TRUE	key phrase: Data-parallel, cluster 2: parallel data, parallel oper
peer-to-peer (P2P)	TRUE	key phrase: Peer-to-Peer (P2P), cluster 3: peer p2p applic distribut
pictorial query specification	TRUE	taxonomy: H.2.8 Image databases, cluster 8: content retriev, cluster 7: queri visual
programming languages	FALSE	
solid modeling	TRUE	taxonomy: I.3.5 Computational Geometry and Object Modeling, cluster 9: csg algorithm, tree csg
spatial algorithms	TRUE	key phrase: Nearest Neighbor, neighbor finding, cluster 0: label compon connect, octre neighbor, cluster 5: quadtre window algorithm, quadtre region algorithm, quadtre region boundari
spatial database	TRUE	taxonomy: H.2.8 Spatial databases and GIS, key phrase: Spatial Databases, DBMS, SAND, cluster 2: spatial data
spatial data structures	TRUE	key phrase: Quadtrees, Data Structures, Spatial Data, Hierarchical Data Structures, spatial data structures, R-Trees, octrees, cluster 2: spatial data, spatial oper, join spatial, cluster 6: line effici, structur data effici, structur data hierarch
visualization	TRUE	cluster 7: visual anim, queri visual

Spatial Database Experts. As discussed in Section 3.4, our system builds a contextual view of researchers in a given field, focusing on recent, frequent, and relevant researchers combined with a listing of pioneers in the field. We present the results from our system for experts and topics in the field of spatial databases. Our system's determination of frequent, recent, and pioneering researchers for spatial databases is

shown in Table 17. Groupings of relevant topics and the top 5 researchers that are in those groupings are shown in Table 18, with descriptive terms included for each cluster.

Table 17. Spatial Databases Experts

Top Authors		Most Recent Top Authors		Pioneers/first in field	
authorName	pubs	authorName	pubs	authorName	pubs
Hanan Samet	36	Markus Schneider	11	Hans-Peter Kriegel	14
Hans-Peter Kriegel	34	Hanan Samet	10	Ralf Schneider	8
Markus Schneider	17	Hae-Young Bae	8	Thomas Brinkhoff	7
Dimitris Papadias	16	Dimitris Papadias	8	Hanan Samet	6
Martin Ester	15	Xiaofang Zhou	6	Jack A. Orenstein	4
Xiaofang Zhou	13	Roger Zimmermann	6	Jan Paredaens	4
Bart Kuijpers	11	Alejandro Pauly	6	Oliver Günther	4
Jörg Sander	11	Cyrus Shahabi	6	Martin Ester	3
Thomas Brinkhoff	11	Nikos Mamoulis	5	Max J. Egenhofer	3
Jae-Woo Chang	10	Wei-Shinn Ku	5	Ralf Hartmut Güting	3
Jianwen Su	10	Cláudio de Souza Baptista	5	Shashi K. Gadia	3
Yufei Tao	10	Yannis Theodoridis	5	Stephan Heep	3
Dirk Van Gucht	9	Yufei Tao	5	Terry G. Glagowski	3
Frantisek Brabec	9	Victor Teixeira de Almeida	5	Venkat N. Gudivada	3
Luc Vandeurzen	9	Ralf Hartmut Güting	5	Xiaowei Xu	3
Shashi Shekhar	9				
		Year Range: 2003 - 2007		Year Range: 1985 - 1996	

Table 18. Spatial Databases - Clusters and Top Authors

Cluster #	Descriptive Terms	Members from Top 5 Authors
0	cluster algorithm mine, cluster densiti algorithm dbscan	Hans-Peter Kriegel, Martin Ester
1	estim queri, join distanc queri, join oper queri	Hanan Samet, Hans-Peter Kriegel, Dimitris Papadias
2	linear queri algebra, languag queri, constraint linear	Hanan Samet, Dimitris Papadias
3	symposium proceed, relationship qualit, topolog relationship	Markus Schneider
4	discoveri pattern associ, mine pattern associ, mine rule associ	Hanan Samet, Hans-Peter Kriegel, Martin Ester
5	imag retriev icon represent, imag retriev color	Hanan Samet
6	queri move object locat, queri network	Hanan Samet, Hans-Peter Kriegel, Markus Schneider, Dimitris Papadias
7	tree index access, tree index structur search	Hanan Samet, Hans-Peter Kriegel, Dimitris Papadias
8	recognit imag featur gabor, video imag	Hanan Samet, Hans-Peter Kriegel, Martin Ester
9	data system geograph model integr	Hanan Samet, Hans-Peter Kriegel, Markus Schneider, Dimitris Papadias

Comparing the top authors for different time ranges brings insights into the types of experts in the field. From the pioneers list, we can identify that Ralf Schneider contributed significantly to the initial work in the field, but did not continue publishing in the field after those initial years. Comparing the top 5 authors for all time with the top 5 pioneers shows that Hanan Samet and Hans-Peter Kriegel were both foundational in the field and long lasting with their publication contributions. Comparing the top 5 authors for all time with the top 5 most recent shows that Markus Schneider and Dimitris Papadias are quite active in the field and would be good candidates to contact if someone had a current expertise need or wished to collaborate on current work in the field. Additionally, Markus Schneider has received a spatial data related NSF CAREER Award (NSF-IIS-0347574), an award given to junior faculty with proven track records of outstanding research and education, confirming our system's assessment of him as a top recent expert.

Overlaying the clusters with the top 5 researchers for the entire document set clarifies the subtopics within the field that each researcher has contributed to. Four out of five of the top 5 researchers have published on querying moving objects, as indicated by cluster 6. In contrast, Martin Ester has not published on querying moving objects, but has published on spatial data mining topics like clustering algorithms, as indicated by cluster 0. Cluster 3 in the overlay distinguishes Markus Schneider from the other top 5 researchers by his research on topological relationships of spatial data. Similarly, cluster 5 sets apart Hanan Samet from the other frequent authors by his contributions on image retrieval. The clusters view of top researchers is essential to our system's approach to expert finding, for without it the user would only have insight into the recent and frequent

authors in the field but would have no indication of what relevant subtopics they were active in.

3.6 Discussion

Our approach to expert and expertise finding favors recall over precision.

Coverage is valued above correctness in our system, with a preference to provide context around information rather than filter out information that may not be correct. We believe that a focus on recall and providing context is appropriate for an expert finder system, especially when the system includes an exploratory interface like ours does (see Chapter 4). Users of the system can easily validate the correctness of identified expertise areas by further refining the results to see if the publications match the identified expertise areas. For the rest of this section, we will discuss the runtime, the advantages, and the shortcomings of our approach.

Runtime analysis. It is important to understand the potential bottlenecks in the approach that our system uses, to be able to predict expected performance while using the system and know the costs of applying our approach to a new domain or system. Our approach has one main component whose runtime must be considered, that being the topic extraction component.

Prior to extracting topics using the Aho-Corasick algorithm, the topic vocabulary is built into a finite state machine, which takes $O(|k|)$, with k =set of keywords and $|k|$ indicating the length of the entire set. Extracting the topics is output-sensitive, being linear in the number of output chunks. So, the algorithm itself is efficient but the runtime required to apply it is dependent on how many topics exist in the documents. Our system implementation took note of this and we allocated a few days to run the topic extraction

algorithm over the almost one million DBLP publications. After the extraction has been applied and the results persisted, no further runtime cost is incurred.

Advantages. The efficacy of an expert finder system depends on if it can assist users with finding the experts in an efficient manner. Similar to search engines, if users continue to be shown unrelated results or have to page through enormous result sets then they will be less likely to continue using that search engine. We believe that our approach to finding experts appropriately balances providing important context to the results without overwhelming the users. Our evaluation showed that our expertise profiles accurately identify most of the topics that a researcher is interested in. The inclusion of taxonomy nodes, clusters, and extracted topics provides three alternate sources of expertise areas. Our assessment showed that some of Hanan Samet's self-identified topics were present in only one of each of the sources, validating the importance of having all three sources shown to the user. Our evaluation example of finding experts in the field of spatial databases showed that our approach is useful for discovering experts that are frequent, recent, and relevant. We were able to draw additional insights from the results by comparing the frequent top authors with the recent, and the overlap between the top authors views helped to clarify the position of each of the researchers.

Shortcomings. Despite its many advantages, our approach is not without shortcomings. In both the area of finding experts for a given topic and topics for a selected researcher, our system falls short in a few aspects.

When identifying experts for a given topic, our system does not produce an absolute ranked list of results, instead favoring to present the most frequent, most

frequent recent, first in field, and clustered views of researchers. Though we believe that our multifaceted views enhance discovery and exploration, in contrast with traditional expert finder systems our approach lacks the ability to produce ranked results of the identified experts.

When identifying expertise areas for a given researcher, our system may lack key data points or may cluster together unrelated publications. The expertise areas derived from the taxonomy nodes and author-provided key phrases are unequivocal in terms of what topic they represent, as opposed to the clusters which may group together unrelated items. However, both the taxonomy nodes and author-provided key phrases may miss key data points. A small percentage of the publications in DBLP have taxonomy nodes specified for them, which can mean that a researcher could be an expert in a certain taxonomy node but have no publications in that node. Additionally, there is no guarantee that the author-provided key phrases are comprehensive. A researcher could be an expert in a topic that no other researchers have listed as a key phrase, and then that topic would go unreported in that researcher's expertise profile. The last major shortcoming in this area is the potential to miss a topic when clustering documents. Balancing the quality of the clustering with the number of clusters means that the accuracy of the results can suffer if the number of clusters needs to be kept low. We chose to keep the number of clusters low to fit within our interactive user interface, but this can lead to unrelated documents being put into the same group and a potential hiding of documents with unique topics.

In summary, we defined our notion of an expert and then presented our approach to identifying experts and expertise areas. We utilize taxonomy nodes and author-provided key phrases as topics for some of the documents. For other documents, we

identify topics by extracting instances of a vocabulary built from author-provided key phrases. Then, we summarized a researcher's expertise areas using explicit metadata like nodes from the ACM CCS taxonomy, extracted topics from author-provided key phrases, and descriptive terms from clusters of their documents. To find experts on a given topic, we provide contextual views of researchers in that area, identifying researchers that have frequent or recent or relevant publications on the topic. We have laid a foundation for finding experts and expertise areas. The following chapter will build off of this work, enabling exploration and discovery of the experts and expertise areas.

Chapter 4: Taxonomy-enhanced Expertise Browser

4.1 Introduction

As described in the motivation for Cutting et al.’s Scatter/Gather [62], users access a document collection with *searches*, that target specific information, and *browsing*, that has no predefined goal but engages in exploration and discovery. Designing a search interface that supports browsing is much more complex than designing a simple keyword search interface. To support exploratory browsing, an interface must provide enough relevant contextual information to support the navigation steps of a user who typically has “a complex information problem, and a poor understanding of terminology and information space structure” [63].

The capability of connecting expert seekers to experts requires a way for the seeker to know what expertise areas they need, or a way to guide the seeker in the process of discovering those areas. In the simplest case, the seeker knows exactly what topic they need to find an expert on and can answer a question like “Who is an expert on spatial join algorithms?”. In a more complex case, the seeker wants to explore within a topic to discover experts in related concepts, for example with a question like “What are some current topics in the field of spatial databases and who are the experts on those topics?”. Presenting the user with the right granularity of information in the right context ensures easier discovery of the experts matching an expertise seeker’s needs.

In this chapter, we present an exploratory search interface to our enhanced DBLP document collection. This interface incorporates browsing by nodes in an extended taxonomy, by topics, and by researchers. It offers a way to find experts with exact topic searches or through browsing within the taxonomy and related topics.

4.2 Background

Grouped Navigation. Search interfaces can assist users with understanding search results by organizing them into meaning groups. These groups can help summarize subareas of the search results, and can also serve as a pathway for further refinement of the search. Two common grouping techniques used in user interfaces are *clustering* and *facets*. Clustering is the application of data mining algorithms to the data to create groups of items more similar to one another than to items in different groups. Facets are metadata, like the author of a document, which form explicit groups of items that share a common facet value.

Clustering techniques are data mining algorithms that produce a partitioning of a data set, with items in the same partition being similar to one another. When applied to search results, clustering can help elicit common themes in the returned documents. Search engines like Clusty (powered by Vivisimo) use document clustering as one of its core navigation techniques (e.g., Figure 18).

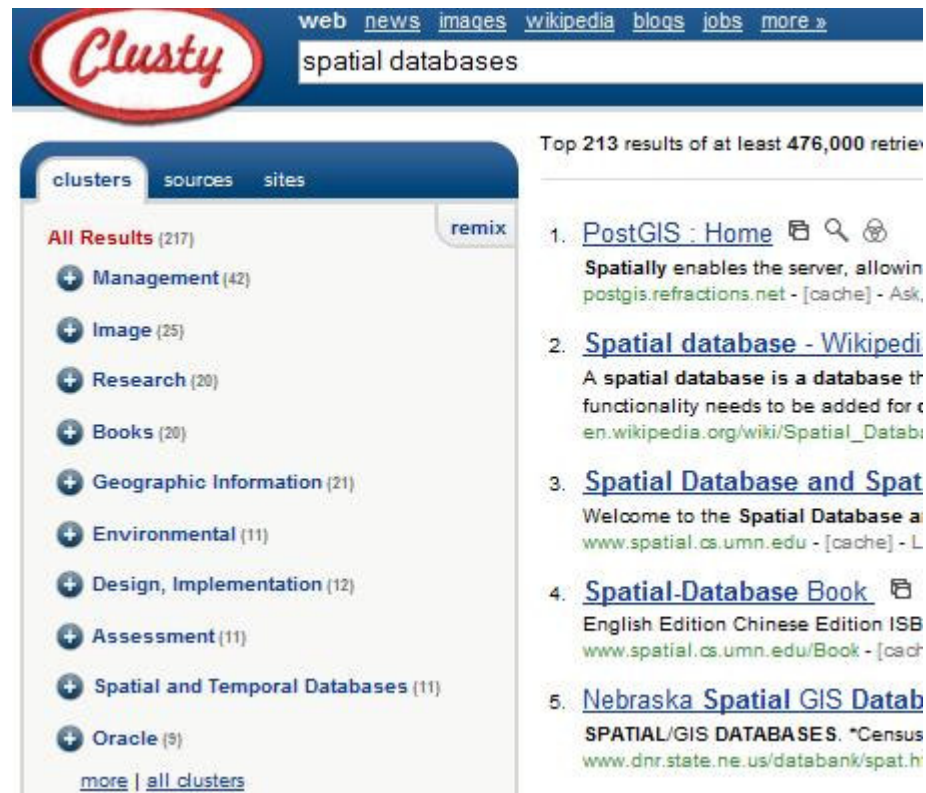


Figure 18. Document Clustering of Search Results (clusty.com)

Search interfaces that support faceted navigation are those that allow for refinement of search results based on one or more facets available in the data. A facet is any type of metadata associated with the documents. For example, many product shopping search engines like NexTag.com (Figure 19) and Shopping.com provide for faceted navigation on metadata like price and manufacturer. The interaction pattern with a faceted search engine involves a mixture of searching along with refining by selecting a facet value with which to further reduce the results. Free text search is integrated with categorical browsing, create an interface that organizes the results and “act[s] as scaffolding for exploration and discovery” [64]. Additionally, Yee et al.’s usability studies [65] showed that users prefer the additional context and easier browsing experience offered by faceted navigation over standard free text search interfaces.



Figure 19. Faceted Navigation (nextag.com)

4.3 Approach and Implementation

We combine the presentation of our approach and implementation (DEF-CAT), to allow visual descriptions from the implementation to complement the textual descriptions of the approach. Our approach to building an exploratory browse interface for our extended version of DBLP focuses on following three use cases. The starting user interface for these modes is shown in Figure 20.



Figure 20. DEF-CAT UI - Starting Page for Three Different Browse Modes

1. *Taxonomy Browse* - a user browses for experts in a given node in the 1998 ACM CCS taxonomy.
2. *Grouped Navigation* - a user navigates within a result set using clusters and facets, using these groups to explore subtopics and related topics.
3. *Profile View* - a user searches for and views an expertise profile for a researcher listed in DBLP.

Taxonomy Browse. Taxonomy organizes topics within a specific domain.

Hierarchical taxonomies provide structure and relationships that lend themselves easily to navigation and exploration. Starting at any node in the taxonomy, a more general topic can be discovered in the parent node and related topics can be explored in the sibling nodes. If the chosen node has children, then subtopics of that node can be navigated to in those child nodes. By browsing around the taxonomy, a user can assess the range of topics in the domain and can easily broaden or restrict their results.

The 1998 ACM CCS taxonomy organizes the field of computer science into 1473 nodes. ACM publications in DBLP can be classified into one or more of those nodes by the publication's authors. By populating the taxonomy nodes with the documents classified in each, we achieve a browse-able organization of those documents. The granularity of topics in the leaf nodes is best described as general, with nodes having topics like "H.2.8 Spatial databases and GIS", and "C.2.4 Distributed databases". Browsing these general topics provides a good macro view of the document collection, with the expected outcome of a user selecting a node to explore in finer detail. The interaction model for taxonomy browsing is described in the Figure 21 flow chart.

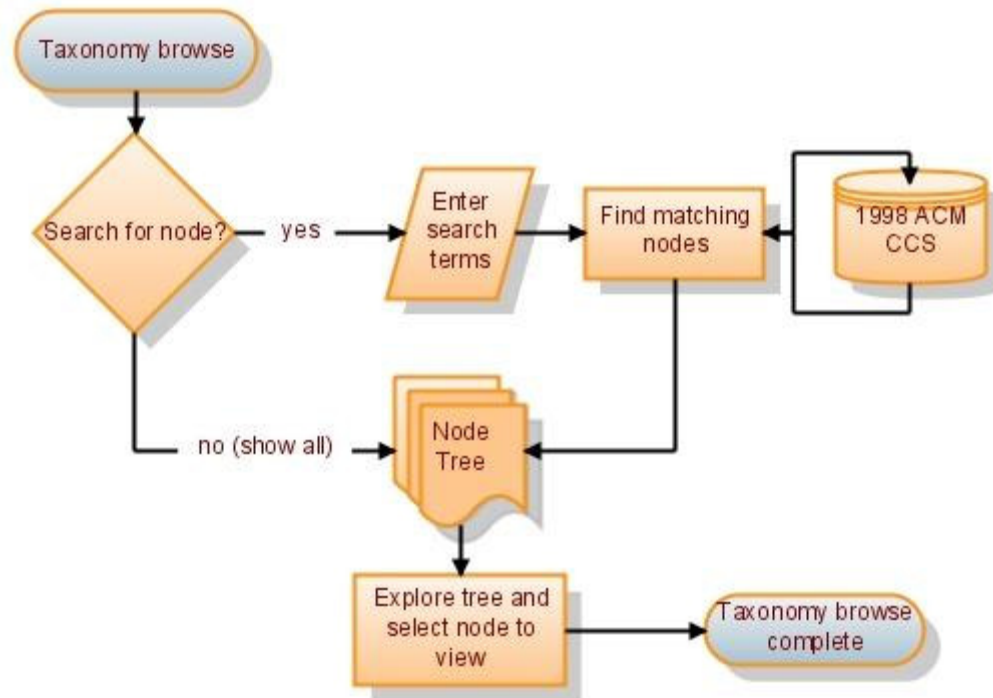


Figure 21. Taxonomy Browse Interaction Model

To encourage exploration of the taxonomy, we support two modes of interaction: browsing the entire taxonomy, or a keyword-search filtered version. A user can start at the root of the tree and navigate down through its children, or they can enter search terms and then browse the nodes that contain those terms. For example, to browse the taxonomy for topics related to databases, the user selects “Taxonomy” as their search type, enters “database” into the search input field, and then can browse through the filtered taxonomy for a specific topic. They can select one of the nodes from the taxonomy to transition to the grouped navigation view. The implementation of this browsing for database topics example is shown in Figure 22.

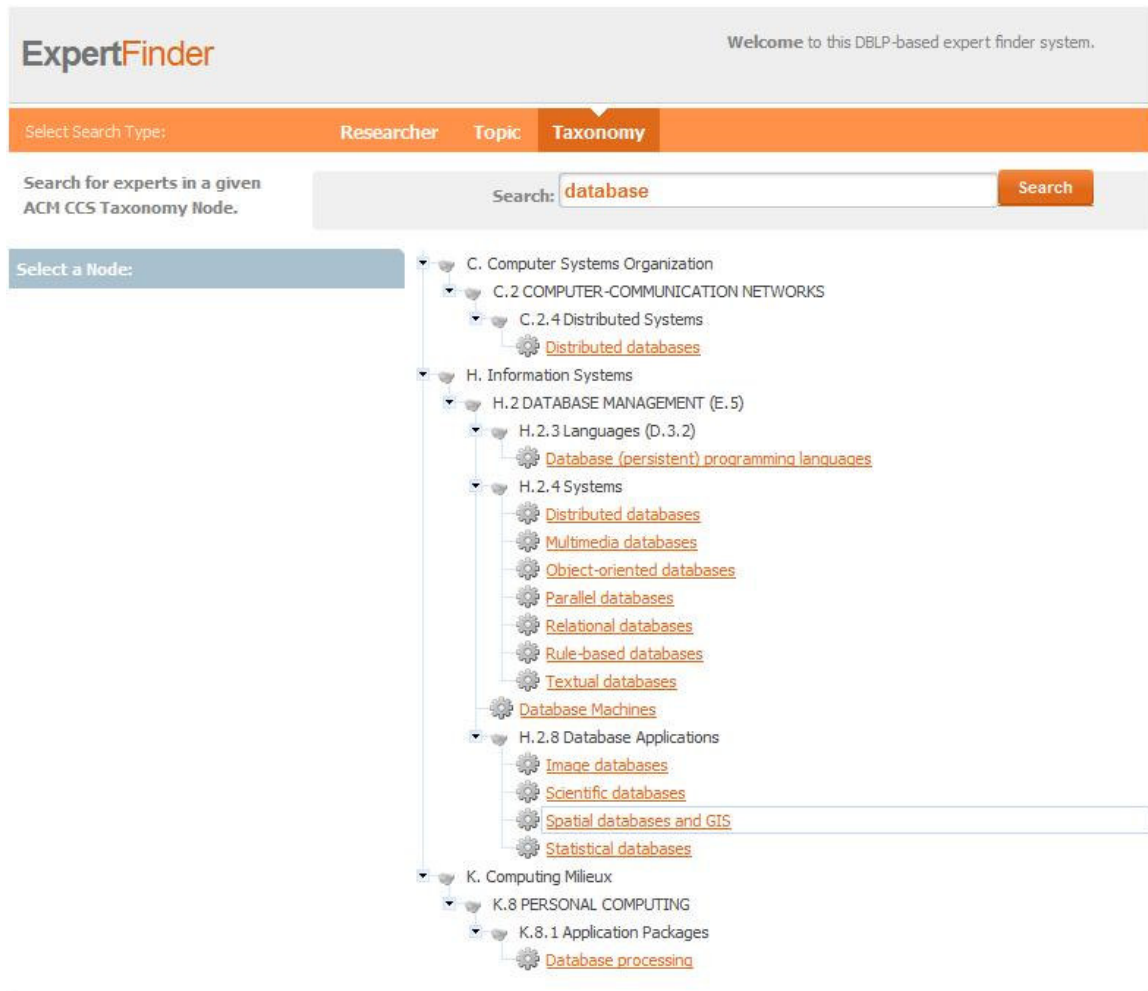


Figure 22. DEF-CAT UI - Taxonomy Browse

We restrict the tree browsing to the taxonomy as it is, without any clustered extensions to the leaf nodes. We do provide the clustered results in the grouped navigation view, but we left them out of the tree view because we believe users interact best in clear stages: first, select the broad topic from the taxonomy and then choose a subtopic or related topic from one of the clusters. By doing this, we have created an extended form of the taxonomy where each node is extended by its related clusters. These clusters provide topics for exploration at a much finer granularity than the ACM CCS taxonomy nodes.

Grouped Navigation. Motivated by Yee et al.'s usability studies [65] referenced in Section 4.2, we augment the listing of documents for a given taxonomy node or topic with contextual groups. Our usage of group navigation allows the user to explore within a document set, slicing the data into subgroups and gaining insight into the range of topics and authors.

The user can enter into the grouped navigation node by either selecting a taxonomy node in the taxonomy browse view or by entering a topic of their choice into a search input field. After selecting a taxonomy node or topic, the user is transitioned into a grouped navigation view, which combines document listings with groups that can be navigated on. The interaction model for this view is described in Figure 23. Within each group, the navigable options are displayed in descending order of the number of documents that relate to that option. For example, the authors facet shows the top authors by publication count in that node. By selecting an option, the document set is refined to show only the documents that relate to that option. The user can click the back button to expand their results again, and can refine by other groups to continue their exploration process.

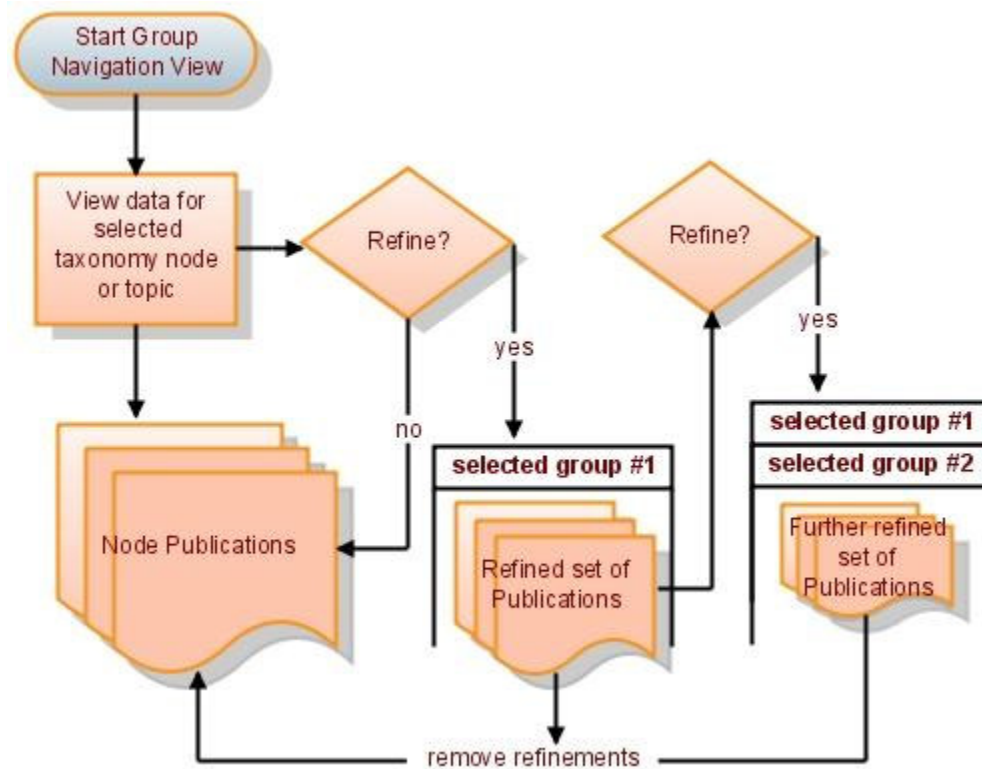


Figure 23. Group Navigation Interaction Model

The implementation of the group navigation interface was modeled after the common group navigation UI layout found on e-commerce sites like BestBuy.com. The navigable groups are shown in a left sidebar, with the search results displayed as a list placed into a center section. Groups are sorted in descending order of occurrence, and the document set can be refined to a particular group by clicking on the hyperlink for that group. We provide the following groups in this view: taxonomy nodes, clusters, key phrases, and authors. Each group addresses a different pattern of interacting with the document set. Examples of these interactions will be provided in the discussion in Section 4.4. Figure 24 demonstrates the grouped navigation interface for the taxonomy node “H.2.8 Database Applications – Spatial databases and GIS”.

ExpertFinder Welcome to this DBLP-based expert finder system.

Select Search Type: **Researcher** **Topic** **Taxonomy**

Search for experts in a given ACM CCS Taxonomy Node. Search:

Node matching:

H.2.8 Database Applications Subjects: Spatial databases and GIS

Taxonomy Nodes:

- H.2.8 Database Applications [Spatial databases and GIS](#) (269)
- H.2.4 Systems [Query processing](#) (29)
- H.2.8 Database Applications [Data mining](#) (24)
- H.2.1 Logical Design [Data models](#) (19)
- H.2.3 Languages (D.3.2) [Query languages](#) (10)
- [more...](#)

Clusters:

- Cluster 0:** [legaci](#), [spatio tempor type](#), [spatio tempor aggreg](#) (19)
- Cluster 1:** [parallel](#), [join spatial filter](#), [join algorithm spatial](#) (18)
- Cluster 2:** [outlier structur spatial](#), [tree index structur spatial](#) (24)
- Cluster 3:** [method extract](#), [cluster method pattern](#), [cluster mine pattern](#) (25)
- Cluster 4:** [spatiotemporal relationship](#), [topolog](#)

Publications: (269)

- Spatial join techniques.**
Edwin H. Jacox, Hanan Samet
ACM Trans. Database Syst. 32(1):7 (2007)
[\[More\]](#)
- Report from the clean slate network research post-sigcomm 2006 workshop.**
Peter B. Key, Jon Crowcroft
Computer Communication Review 37(1):75-78 (2007)
[\[More\]](#)
- An efficient and accurate method for evaluating time series similarity.**
Jignesh M. Patel, Michael D. Morse
SIGMOD Conference 2007:569-580
[\[More\]](#)
- An OLAP system for network-constrained moving objects.**
Xiaofeng Meng, Karine Zeitouni, Tao Wan
SAC 2007:13-18
[\[More\]](#)
- Structural similarity in geographical queries to improve query answering.**
Patrizia Grifoni, Fernando Ferri, Anna Formica, Maurizio Rafanelli, Arianna D'Ulizia
SAC 2007:19-23
[\[More\]](#)
- Snapshot density queries on location sensors.**

Figure 24. DEF-CAT UI - Grouped Navigation

Profile View. We define a researcher's profile as a contextualized view of their contributions to the field, derived from their publications listed in DBLP. Our profiles have items analogous to what many researchers list on their home pages: publication history and topics of research interest. However, we extend our profiles beyond what is listed on author home pages to include frequent co-authors, cluster groups, taxonomy nodes, and a timeline view of the author's publications. The timeline view, advocated for by Alonso, Baeza-Yates, and Gertz [66], provides an alternate representation to the results that can facilitate temporal exploration of the document set. The interaction model

of this view is equivalent to the interaction model (Figure 23) in the grouped navigation view. Users can refine the document set by any number of related groups, for example extracted topics, a certain co-author, or a specific taxonomy node. Our implementation for the profile view is shown in Figure 25, Figure 26 (reorganized to fit on one page), and Figure 27.

ExpertFinder Welcome to this DBLP-based expert finder system.

Select Search Type: **Researcher** Topic Taxonomy

Search for a researcher by name and discover their expertise areas. Search: **Hanan Samet** Search

Researcher:
Hanan Samet

Taxonomy Nodes:

- H.2.8 Database Applications**
[Spatial databases and GIS](#) (14)
- H.2.4 Systems**
[Query processing](#) (9)
- I.3.5 Computational Geometry and Object Modeling**
[Curve, surface, solid, and object representations](#) (5)
- I.3.3 Picture/Image Generation**
[Display algorithms](#) (4)
- I.3.5 Computational Geometry and Object Modeling**
[Geometric algorithms, languages, and systems](#) (3)
- [more...](#)

Clusters:

Cluster 0: [label compon connect, octre neighbor](#) (11)

Publications: (171)

- Spatial join techniques.**
Edwin H. Jacox, Hanan Samet
ACM Trans. Database Syst. 32(1):7 (2007)
[\[More\]](#)
- Using a distributed quadtree index in peer-to-peer networks.**
Aaron Harwood, Egemen Tanin, Hanan Samet
VLDB J. 16(2):165-178 (2007)
[\[More\]](#)
- Execution time analysis of a top-down R-tree construction algorithm.**
Houman Alborzi, Hanan Samet
Inf. Process. Lett. 101(1):6-12 (2007)
- Client-Based Spatial Browsing on the World Wide Web.**
Hanan Samet, Frantisek Brabec
IEEE Internet Computing 11(1):52-59 (2007)
[\[More\]](#)
- A fast all nearest neighbor algorithm for applications involving large point-clouds.**
Jagan Sankaranarayanan, Hanan Samet, Amitabh Varshney
Computers & Graphics 31(2):157-174 (2007)

Figure 25. DEF-CAT UI - Profile View

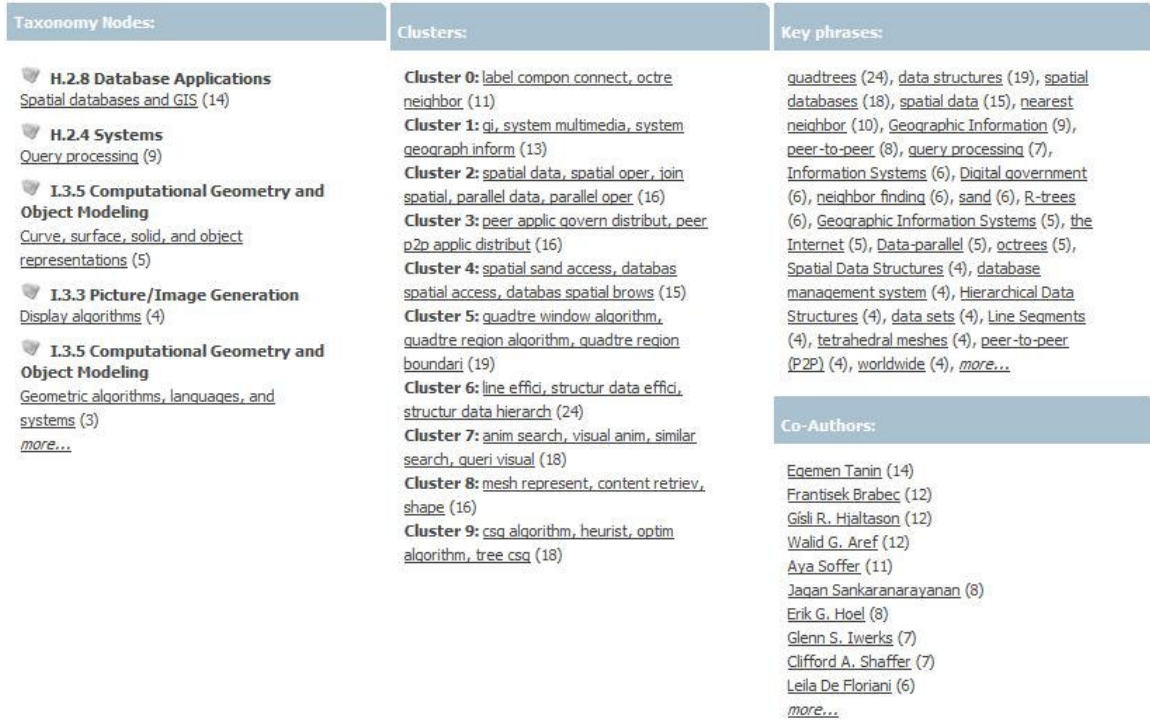


Figure 26. DEF-CAT UI - Profile View Grouped Navigation

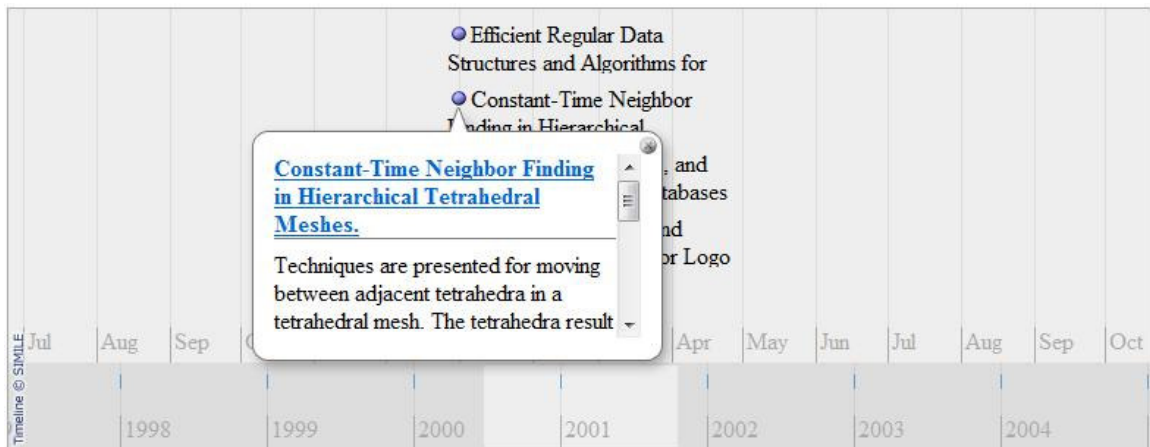


Figure 27. DEF-CAT UI - Profile View Timeline

Implementation specifics. We have demonstrated our implementation visually in the preceding parts of this section. We now provide details on how we achieved the described implementation.

The user interface was developed using the Java Server Faces (JSF) framework [67]. JSF is a recent web technology developed as a part of the Java Enterprise Edition

technology suite. JSF is a component-based web framework, aiming to simplify development of Java-based web interfaces. We developed the UI pages using Facelets [68], a JSF-centric view technology that supports templating and reusable components. The HTML/CSS template was modeled after a template designed by Luka Cvrk of solucija.com and posted on the Open Source Web Design site [69]. The icons were provided by Mark James in the Silk icon library [70]. The timeline in the profile view was implemented using the MIT SIMILE Project's Timeline [71]. To provide a rich user experience, we leveraged dynamic web components from a JSF library called RichFaces [72]. RichFaces provides UI components like trees, collapsible panels, and scrollable data tables. We developed the UI using an iterative approach, adding functionality in stages of increasing complexity.

4.4 Discussion

A usability study or other formal evaluation of our exploratory interface to DBLP is outside the scope of this thesis work. Instead, we perform a walkthrough, similar to Wharton et al.'s Cognitive Walkthrough usability method [73], of a task scenario to investigate the advantages and shortcomings of our system.

Task scenario. Our task scenario is to explore experts and topics on applying data mining techniques to spatial databases. Starting from the taxonomy browse view, we search for “database” and are presented with 12 matching leaf nodes containing topics like “Distributed databases”, “Spatial databases and GIS”, and “Scientific Databases”. “H.2.8 Database applications - Spatial databases and GIS” will suffice, and there is a parent node “H.2 Database Management” that could be explored further if our selection

proves to be too specific. We select this node to view publications classified into it by their authors.

After selecting the node, we are presented with the grouped navigation view of 269 publications in that node. We have not yet identified research on spatial data mining, and reviewing 269 publications to find this research is a large task. Instead, we turn to the grouped navigation provided by the system. The taxonomy node group shows “H.2.8 Database applications – Data mining” as co-occurring with the spatial database taxonomy node in 24 publications. Examples of publications that are classified into both of those taxonomy nodes are Zhang et al.’s *Fast mining of spatial collocations* [74] and Son et al.’s *A Spatial Data Mining Method by Clustering Analysis* [75]. The clusters group has one cluster of 25 publications that is immediately recognizable as data mining: “method extract, cluster method pattern, cluster mine pattern”. Examples of publications in this refined data set are Harel and Koren’s *Clustering spatial data using random walks* [76] and Guo, Peuquet, and Gahegan’s *Opening the black box: interactive hierarchical clustering for multivariate spatial patterns* [77]. In the group of extracted author-provided key phrases, the top 25 items only contain one relevant to data mining: “data mining”, with 14 publications for it. Among the top authors by publication count in the area of spatial database data mining are Hans-Peter Kriegel, Martin Ester, and Michelangelo Ceci, who published papers like *Spatial Data Mining: A Database Approach* [78]. Each of these groups enhances the ACM CCS taxonomy node related to spatial databases by providing a richer description of the range of topics in publications classified under that node.

Advantages. The task scenario walkthrough elucidated multiple advantages of our system. We classify these advantages into two areas: those that benefit exploring the document collection and those that benefit finding experts.

Compared with simple keyword search-and-display systems, our system offers unique advantages for exploring the collection. To retrieve the initial set of documents to explore, our system supports free text/keyword searches like other systems, but it also supports using the taxonomy to select a subset of documents. Using the taxonomy provides an initial set of general categories to choose between, and obviates the need for the exploratory user to know exactly what topic to enter into a free text search system. After the initial set of documents has been retrieved, our system offers the advantage of summarizing features present in the data set like frequent authors and key phrases, and co-occurring taxonomy nodes. Using these features, the user can more easily get an overview of the related groups of documents present in the collection than if they were required to parse through all of the documents. Additionally, related topics and subtopics are easily explored through the co-occurring taxonomy nodes, the clusters, and the key phrases. In our task scenario, the related taxonomy nodes and clusters were the most useful for discovering publications on spatial databases and data mining.

The naïve way to identify experts in any search system is to find the most frequent authors for a given topic. Ignoring the debate on the merits of equating frequency with expertise, our system improves expert identification for a given topic over simple keyword search-and-display systems. We do this primarily through the grouped navigation view and profile view. After the user has selected a taxonomy node or entered a topic to search for, they can immediately identify the most frequent authors in the

resulting document set. If the topic is too broad, they can refine the document set by navigating down into a co-occurring taxonomy node, a cluster, or a key phrase. They can verify the results by viewing an author's profile to see if their expected topic shows up in the view, or they could refine the document set down to the author's publications that our system identified as relating to the topic. We facilitate the user's process of determining when their topic and result set are specific enough, while also allowing them to verify the identified experts through alternate views like the profile view.

Shortcomings. No system is without flaws, and understanding those flaws can help identify areas of improvement for future work.

Our most prominent shortcoming is the problem of balancing the relevancy of the information we provide with the limited screen real estate of our web-based GUI. The grouped navigation left sidebar is quite expansive when showing taxonomy nodes, clusters, topics, and frequent authors. The abundance of refinement options can be overwhelming, especially given that most users expect Google-style keyword and ranked results simplicity in search systems. For the users that are willing to utilize the grouped navigation options, the relevancy of what is displayed cannot be guaranteed for all interactions. For example, in our task scenario of exploring spatial databases and data mining, the key phrases shown in the UI were limited to the top 25 that the system deemed relevant. However, the only relevant phrase in that list was "data mining", which did not offer any significant additions to what the user could find by navigating in the related taxonomy node or cluster. To address this weakness, an interaction model that required the user to select a related taxonomy node or cluster first could improve the key phrases that are displayed.

In summary, we designed a taxonomy-enhanced expertise browser that supports three modes of interaction: taxonomy browse, grouped navigation, and profile view. The taxonomy browse interface guides the user through the taxonomy by allowing for full text searches across all nodes and then tree walking to refine or expand the nodes. The grouped navigation view facilitates discovery of various groups, like subtopics or frequently published authors, in the document collection. The profile view summarizes an author's contributions to the research field and also supports group navigation within those publications. By combining the taxonomy browse view with grouped navigation view, we achieve an extended form of the ACM CCS taxonomy where leaf nodes are extended by the clusters of related document contained within them.

Chapter 5: Related Work

In Chapter 2, we built a foundation of conceptual background knowledge so that the reader could understand the context around our approach. Now that we have presented our approach, we will revisit the questions of who has done similar work to ours and how does our approach relate to their approach. We review related work in the areas of modeling expert finder systems, designing the system architecture of an expertise finder, building an expertise profile, extracting competency areas, and browsing for experts.

Modeling Expert Finder Systems. Yimam-Seid and Kobsa review existing expert finder system and suggest a domain model for expert finders, segmenting the domain into seven aspects that systems vary on [79]. They start by defining two motivations for expert seeking:

1. *Information need* – an expert is sought as a source of information.
2. *Expertise need* – an expert is sought who can perform an organizational or social function.

Yimam-Seid and Kobsa accurately position expert finder systems as essential organizational information systems, capable of supporting collaborative work and knowledge management within an enterprise. Their domain model for expert finding systems summarizes the key components of any such system. Their model helped us structure our approach and implementation, delineating components to be built for each area and helping to clarify the interfaces between components. They identify seven domain factors in expert finder systems:

1. *Basis for expertise recognition*: a system must identify its sources of expertise indicators and be able to recognize and gather these sources.
2. *Expertise indicator extraction*: indicators of expertise are extracted from the retrieved sources through domain knowledge dependent or independent techniques.
3. *Expertise models*: a model of a person's expertise areas, with a possible degree of uncertainty on the strength of association between a person and an element in the model. The origin of a model may be dynamic or static, with the static approaches being either stored and managed by personal agents or associated with document clusters. An aggregated expertise model combines multiple sources of expertise indicators, at the cost of some potential staleness in the data included in the model.
4. *Query mechanisms*: querying can be explicit, like user generated queries, or implicit, for example extracted from document browse and read history.
5. *Matching operations*: matching is the process of building associations between someone seeking expertise and those that offer it. It involves retrieval techniques like keyword extraction and determining similarity between documents and queries.
6. *Output presentation*: the identified experts are ranked prior to being displayed and related information may be returned to add relevant context to the profiles of each expert.

7. *Adaptation and learning operations*: expert finder systems can be self-learning systems that improve their results by collecting usage information and appropriately leveraging it.

A summary of their analysis of the expert finder domain is given in Table 19 (from [79]).

Table 19. An Intuitive Domain Model of Expert Finding Systems

Basis for Expertise Recognition	Expertise Indicator Extraction Operation	Expertise Modeling	Query Mechanism	Matching Operations	Output Presentations	Adaptation and Learning operations
* Explicit	* Domain knowledge driven	* Query-time generated	* Explicit query for expert	* Exact / overlap matching	* Ranked list of names	* Adaptation using user's expertise models
* Implicit	* Domain knowledge independent	* Personal-agent based	* Induce information / expert need	* Statistical /similarity-based matching	* Ranked list plus personal details	* User modeling
		* Association to aggregated expertise model		* Inference matching	* Experts contextualized in their social network	* Gathering relevance feedback
					* Documents / organizational groups containing relevant experts	* Community expertise evaluation / rating

Yimam-Seid and Kobsa [79] implemented an expert finder system that utilized the seven domain factors that they identified. Their approach, called DEMOIR, uses an aggregated expertise model, in contrast with the query time model that can suffer from high latency at retrieval time and the agent-based model that may not have enough sources of information. They build a centralized data warehouse of integrated expertise sources, which are populated by distributed source gatherers and converters. They

segment their warehouse into an aggregated expertise model that represents the domains of various experts and expert models that represent the experts themselves.

Similar to their DEMOIR approach, our system used an aggregate expertise model that leveraged a range of data sources to build a model of the computer science domain. We segment our system into the topics and taxonomy that represent the domain and the expert model that represents the researchers in DBLP. Their research in [79] omits specifics of how to build up the aggregated domain model for expertise areas. In contrast, our research focuses on the specifics of how to build a model of expertise areas by leveraging author provided key phrases and an existing taxonomy. Despite the differences, their contributions benefitted us. Their expert finder domain model, with its seven factors, provided us with the conceptual constructs to evaluate other expert finder systems. When we encountered a new system, we were able to look at the seven factors and analyze how that system chose to approach each factor.

Expertise Recommender. McDonald and Ackerman performed a field study [1] to determine how social, rather than technological, seeking out of experts occurs. They discovered that knowledge seekers in the company that they studied engaged in three phases of expertise location:

1. *Expertise identification*: identifying candidates who probably have experience and expertise in a given area
2. *Expertise selection*: choosing an identified candidate to approach for assistance

3. *Escalation*: iterating through identification and selection, leveraging what was learned in each iteration, until a satisfactory solution is found for the expertise need

Their approach to expertise finding, described in [80], follows the model that they discovered through their field study. They create an architecture for expertise recommendation called ER-arch, which contains three core supervisors: a *profiling supervisor*, an *identification supervisor*, and a *selection supervisor*. The profiling supervisor creates and maintains profiles of the individuals that are candidates for expertise selection. The identification supervisor retrieves all potential candidates that are relevant to a client-initiated expert search. The selection supervisor ranks the candidates according to their expertise in the requested topics and may refine the set by removing some of the candidates from it. An additional component called *Interaction Management* manages the interaction with the client by performing duties like tracking the requests/results returned by the system and building historical data that can be used to improve future results. This allows for improved usability by presenting a user who has previously searched for an expert on a given topic with their previously returned results when they search for that topic in the future. Their system implementation, called ER, produced a working prototype of these concepts that was used in a customer technical support scenario.

In comparison with our system, ER-arch's underlying patterns match the expertise browse patterns that we utilize. Their escalation pattern, which involves iterating through the expertise finding process until a desired accuracy and granularity are reached, parallels our faceted browsing interface, which supports result set refinement and

browsing until a satisfactory set of results are returned. In both systems, the user is in control of the desired level of accuracy that they wish to achieve and can refine or expand their results as needed. Like our approach, they automatically build expertise profiles rather than requiring the experts to self-identify their areas of expertise.

In contrast to our system, ER did not support building topical expertise profiles. Their system supported search by support need or code area and retrieval of the relevant experts (i.e., “who knows about Y”), but it did not support answering “what does person X know”. Our approach builds topical expertise profiles by leveraging author-provided key phrases. We also augment those profiles by including clustered groups that indicate subtopics or related topics that the author is knowledgeable in. However, they offer functionality that we do not by leveraging the social network within their company when selecting experts. Experts within one’s social network are ranked higher than those outside of it, as there is a greater likelihood of collaboration when one is friends with an expert. In [3], McDonald studied the use of social networks versus no social networks when recommending an expert. Our approach did not leverage a social network to rank retrieved experts, as our model of client interaction supposes that we do not know who the client is that is seeking an expert. We could extend our approach to recognize each user if they were in DBLP already, and then could rank experts higher that are within their co-authorship extended network.

Expertise Profiles. Balog’s and de Rijke’s research on how to determine expert topical and social profiles [81] inspired our investigation into automatic generation of topical profiles. They automatically build expertise topical profiles from the TREC W3C-corpus [82] (provided as a part of the TREC-2005 Enterprise Track [8]) by using

information retrieval techniques and post-processing filtering refinements. They define a topical profile as a collection of $\langle \text{knowledge area}, \text{competency} \rangle$ pairs. Their building of topical profiles is broken down into two stages: first, the identification of knowledge areas and second, the determination of competency levels. They propose two models for building a profile:

1. *Document centric*: retrieve all documents related to the knowledge area.

Then, calculate a researcher's competency in that area by summing the relevance of their documents, within the retrieved set of documents, to the area.

2. *Keyword centric*: equate a researcher's competency in an area with the ratio of co-occurrence between keywords extracted from a candidate's documents and those extracted from all documents in a given knowledge area. Keywords are identified as the top 20 words within a document, where the ranking is determined by the TF-IDF weighting formula.

These two models are used to determine ranking of a set of candidate experts for a given topic. They also are used to identify expertise areas for a given candidate, after filtering out topics for which the candidate is not one of the top f candidates for that area.

Balog's and de Rijke's approach to building expertise profiles has similarities to our approach to building profiles. Given a set of documents, they apply extraction techniques to identify the relevant keywords within those documents. Their extraction techniques use TF-IDF as a measure of importance, and they keep the top 20 keywords in each document. We also extract topics from documents in our approach. Rather than taking all high scoring TF-IDF terms as the important terms, we use author-provided key

phrases as the important elements to look for. Our approach produces semantically recognizable phrases (e.g., spatial database) compared to theirs which produces individual terms (e.g., checkpoint, markup). Phrases are easier to comprehend than individual terms, as they provide better context in which to understand the specific knowledge area.

Even though Balog’s and de Rijke’s topic profiles inspired our techniques for building topic profiles, there are differences that distinguish our approach from theirs. The most prominent difference is that they assume that the knowledge areas are predefined, scoping down their implementation to only address how to measure a person’s competency in an area. They did this because 50 topics were available as a part of the 2005 TREC Enterprise Search Track [8]. Our approach does not presume that the topics are provided, but rather enables topic identification through extracting author-provided key phrases and subsequent removal of low information phrases. We use these extracted key phrases to build a topic vocabulary, which we use to describe areas of expertise. We also identify related topics and sub-topics through document clustering, augmenting the identified expertise areas with further information to describe specializations in each area. For example, our approach is able to identify experts in the area of “spatial databases”, but then it can go further and denote a person as an expert in the related topic of “moving object queries”, assuming that they had publications in a cluster described by those terms.

Extracting Competency Areas. Zhu et al.’s CORDER [83] approaches the problem of determining knowledge areas by looking at the co-occurrence of extracted entities in a document set. They present an unsupervised approach to discovering relationships and evaluate its usefulness in the context of identifying researchers’

competency areas. They utilize named entity recognition (NER) to identify entities of types like person, organization, and knowledge area. They apply their techniques to a heterogeneous document set obtained by crawling an organization's web site. Co-occurrence, distance, frequency, and page relevance contribute to the relation strength between two named entities. Their automated approach removes the need to define a taxonomy of skills prior to assessing competency in those skills. However, the disadvantage of their automated approach is that the quality of its extracted areas is highly correlated with the effectiveness of the NER. If the NER does a poor job of recognizing knowledge areas, for example subjects and topics, then low quality competency areas will result.

Similar to CORDER, our approach utilizes automatic extraction of knowledge areas. But, we believe that the relevance and quality of our results is better than what can be produced by an NER approach. One of our sources of knowledge areas is the author provided key phrases that annotate their publications. These keywords describe the concepts, both broad and specific, that are highly relevant for a given publication. Named Entity Recognition techniques may be able to extract key phrases close to what an author would provide, but the cognitive capacity of the human mind will generally out perform these techniques. Another one of our sources of knowledge areas is the terms that describe document clusters. We equate these with subtopics and related topics to the main topic that was searched for. This automated approach is similar to CORDER's automated NER usage to identify knowledge areas. However, the descriptive terms for each cluster describe the commonalities between the documents in the cluster, whereas CORDER's knowledge areas are produced solely by co-occurring entities within a given

document. It is very difficult to identify subtopics of a given knowledge area when relying entirely on co-occurrence within a document. Our approach that retrieves all documents for that area before clustering them into groups can more skillfully identify the subtopics within those documents.

Expertise Browsing. In addition to addressing the problem of identifying expertise areas, an expert finder system must provide a way to navigate through its data. Seekers of expertise must be able to query the system for experts and need to be able to browse through the expertise areas that have been identified for people in the system. The general approach that existing work has taken is to support a search and refine browsing paradigm. The user starts by searching for a topic or researcher and then the system returned contextual views of the resulting documents that can be further refined based on available metadata. Existing approaches vary on how they present the navigable items and on whether or not they leverage algorithms and external data to enhance the available navigable items.

ACM introduced its Author Profile Pages in April 2008 [84], enhancing their digital library with people-centric views of contributions to the field of computing. The Author Profile Page for a given researcher includes their publications, co-authors, subject areas from the ACM CCS taxonomy, author-provided key phrases, and a link to the author's home page. Personal information can be added or edited by the person that the profile describes. ACM also lists bibliometrics in the profiles, including metrics like publication years, publication count, and citation count. No additional algorithms or external data are provided to enhance the navigable items.

Faceted DBLP [85] is an enhanced search and browse interface for DBLP. Its data comes from DBLP++ [86], an extended version of DBLP that includes key phrases and abstracts from public web pages. They developed and used Diederich and Balke’s Semantic GrowBag algorithm [87], which utilizes higher order co-occurrence to build association graphs of topics, to automatically build lightweight topic categories for a document collection. Users can view the generated topic graphs to identify subtopics and related topics and then can select one or more to navigate into. They support the faceted navigation paradigm, so search results can be refined by one or more facets like publication years, publication types, venues, authors, and the generated topics facet.

Our system shares many similar characteristics with the ACM Author Profile Pages and Faceted DBLP. Like ACM and Faceted DBLP, we leverage associated metadata like authors, taxonomy nodes, and key phrases for summarization and navigation purposes. Our interaction model and layout is similar to Faceted DBLP in that we provide grouped navigation options, like authors, alongside the documents listing. Also, we utilize author-provided key phrases as topics that can be explored, just as ACM and Faceted DBLP do.

Though we share many similarities with the ACM and Faceted DBLP approaches, it is our differences that set us apart. In contrast with the ACM approach, we offer an interaction model and interface that is geared towards easy refinement and exploration of documents. The ACM approach segments related groups like keywords and taxonomy nodes into separate pages, forcing the user into a pattern of navigation that discourages iterative refinement of a document set. To view all of an author’s publications that mention two different keywords requires navigating to the keywords page twice, and

seeing two different pages of the refined document sets compared with our single page interface. In contrast with both ACM and Faceted DBLP, we build a topic vocabulary from author-provided key phrases and then identify instances of that vocabulary in all documents, rather than restricting ourselves to displaying key phrases only on the documents from which they originate. We also improve upon the author-generated topics by removing duplicates and low information phrases after extracting instances of the topic vocabulary in a publication. We believe that the topic vocabulary produced by our algorithms is richer and more suited to document collection navigation and summarization than simply displaying key phrases that the authors have provided for a given publication. Additionally, the clustering algorithm that we apply produces groups that cannot be identified from the explicit information that the GrowBag approach uses to build its categories. In conjunction with taxonomy nodes and key phrases, the clusters offer more complete coverage of topics and related items in a document collection.

Chapter 6: Conclusions and Future Work

Expert finder systems enable retrieval of experts for a given topic and expertise areas for a given researcher. When users come to these systems with an information need, they are provided with one or more experts that could satisfy that need.

Alternatively, when users wish to assess the range of expertise for a person listed in the system, the system answers with its understanding of that person's areas of expertise. These systems answer the question of "Who knows what" and the related question of "What is known by whom".

We have contributed a new approach to expert finding in the field of computer science. To identify expertise areas, we start by leveraging author-provided key phrases to build a topic vocabulary for computer science. We combine that vocabulary with taxonomy nodes and terms describing clusters of related documents to build a rich representation of expertise areas. To identify experts on a given topic, we find researchers who have published on that topic and promote those that are frequent, recent, and relevant to the topic, along with those that were pioneers in the field. We have contributed a user interface that enables exploratory search for expert and expertise areas. Our user interface leverages the ACM CCS taxonomy by supporting navigation through its general topics, while also extending it by extracting subtopics and related topics for each node. We have assessed our approach as effective in identifying experts and expertise areas and facilitating expertise browsing through an extended taxonomy.

Though we have developed a complete approach to finding experts in the field of computer science, there are many possible points to extend or improve upon in our approach. We extended ACM CCS taxonomy nodes in our user interface by providing

contextual views of the topics within each node, where the topics were implicit in the clusters or explicit in our generated topic vocabulary or were related taxonomy nodes. However, ACM publications continue to be classified according to the fixed 1998 version of the taxonomy that we used. Future work could investigate creating the next version of the taxonomy by leveraging our approach to generate relevant subtopics for each node. Additionally, the accuracy of our approach could be improved by performing a formal evaluation using domain experts. Involving domain experts could help us better understand the deficiencies in our algorithms and provide the ground truth needed to enable tuning of our approach. Finally, future research could look at improving our use of clustering to extract subtopics and related topics from a document set. Our system's use of clustering was effective at identifying some of the subtopics present in a document set, but it favored keeping the number of clusters low for presentation purposes at the expense of not discovering all possible subtopics. Future work could determine how to ensure a minimum similarity level of documents in each cluster without producing an overwhelming number of clusters for users to parse through. Any of these extensions would enhance what we consider to be an effective baseline approach to expert finding in the field of computer science.

Bibliography

1. McDonald, D.W. and M.S. Ackerman, *Just talk to me: a field study of expertise location*. Proceedings of the 1998 ACM conference on Computer supported cooperative work, 1998: p. 315-324.
2. Becerra-Fernandez, I., *Searching for experts on the Web: A review of contemporary expertise locator systems*. ACM Transactions on Internet Technology (TOIT), 2006. **6**(4): p. 333-355.
3. McDonald, D.W., *Recommending collaboration with social networks: a comparative evaluation*. Proceedings of the SIGCHI conference on Human factors in computing systems, 2003: p. 593-600.
4. Ley, M., *DBLP (Digital Bibliography & Library Project)*, <http://dblp.uni-trier.de/>.
5. Davenport, T.H., *Knowledge Management Case Study: Knowledge Management at Hewlett-Packard, Early 1996*. Austin: Universidad de Texas. URL:<
<http://www.bus.utexas.edu/kman/hpcase.htm>>. Consultado el, 1996. **27**.
6. Davenport, T.H., *Ten Principles of Knowledge Management and Four Case Studies*. Knowledge and Process Management, 1997. **4**(3): p. 187-208.
7. Streeter, L.A. and K.E. Lochbaum, *Who Knows: A System Based on Automatic Representation of Semantic Structure*. RIAO, 1988. **88**: p. 380-388.
8. Craswell, N., A. de Vries, and I. Soboroff, *Overview of the TREC-2005 Enterprise Track*. TREC 2005 Conference Notebook, 2005: p. 199-205.
9. Balog, K., L. Azzopardi, and M. de Rijke, *Formal models for expert finding in enterprise corpora*. Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, 2006: p. 43-50.
10. Maybury, M.T., *Expert Finding Systems*. 2006, The MITRE Corporation.
11. greenchameleon.com. *Taxonomy_concept_map1_thumb.gif* (GIF Image, 500x322 pixels). 2006 [cited July 2008]; Available from:
http://greenchameleon.com/uploads/Taxonomy_concept_map1_thumb.gif.
12. Follette, W.C. and A.C. Houts, *Models of scientific progress and the role of theory in taxonomy development: a case study of the DSM*. J Consult Clin Psychol, 1996. **64**(6): p. 1120-32.
13. www.encyclopedia.com. *Taxonomy (Science of Everyday Things)*. 2002 [cited July 2008]; Available from: <http://www.encyclopedia.com/doc/1G2-3408600144.html>.
14. dir.yahoo.com. *Yahoo! Directory*. 2008 [cited July 2008]; Available from:
<http://dir.yahoo.com/>.
15. www.pandia.com. *Pandia - New Alta Vista, Part One*. 2000 [cited July 2008]; Available from: <http://www.pandia.com/searchworld/2000-28-alta-vista.html>.
16. web.archive.org. *AltaVista: Main Page*. 1999 [cited July 2008]; Available from: <http://web.archive.org/web/19990125093146/www.altavista.com/>.
17. Bloom, B.S., *Taxonomy of Educational Objectives: The Classification of Educational Goals*. 1956: Longman Higher Education.
18. Krathwohl, D.R., *A Revision of Bloom's Taxonomy: An Overview*. Theory Into Practice, 2002. **41**(4): p. 212-218.
19. www.acm.org. *ACM Computing Classification Systems*. 1999 [cited July 2008]; Available from: <http://www.acm.org/class/>.

20. Machinery, A.f.C. *HOW TO CLASSIFY WORKS USING ACM'S COMPUTING CLASSIFICATION SYSTEM*. 2006 [cited July 2008]; Available from: http://www.acm.org/class/how_to_use.html.
21. www.acm.org. *1998 ACM Computing Classification System*. 2008 [cited July 2008]; Available from: <http://www.acm.org/class/1998/>.
22. Hjaltason, G.R. and H. Samet, *Distance browsing in spatial databases*. ACM Trans. Database Syst., 1999. **24**(2): p. 265-318.
23. Manning, C.D., P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. 2008: Cambridge University Press.
24. Salton, G., A. Wong, and C.S. Yang, *A vector space model for automatic indexing*. Communications of the ACM, 1975. **18**(11): p. 613-620.
25. Craswell, N., A. de Vries, and I. Soboroff, *Overview of the TREC-2005 Enterprise Track*.
26. lsd.cs.uga.edu. *EFW'07 - Home - 1st International ExpertFinder Workshop 2007*. 2008 [cited June 2008]; Available from: <http://lsdis.cs.uga.edu/~aleman/efw2007/>.
27. www.merriam-webster.com. *expert - Definition from the Merriam-Webster Online Dictionary*. 2008 [cited June 2008]; Available from: <http://www.merriam-webster.com/dictionary/expert>.
28. www.cos.com. *Community of Science (COS) - Funding resource, expertise database and abstract management system*. 2008 [cited June 2008]; Available from: <http://www.cos.com/>.
29. *DBLP: Claude E. Shannon*. 2008 [cited June 2008]; Available from: http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/s/Shannon:Claude_E=.html.
30. Rogers, E.M. and T.W. Valente, *A History of Information Theory in Communication Research*. Between Communication and Information, 1993: p. 47.
31. Tho, Q.T., S.C. Hui, and A.C.M. Fong, *A citation-based document retrieval system for finding research expertise*. Information Processing and Management, 2007. **43**(1): p. 248-264.
32. www.cs.umd.edu. *Home Page of Prof.Hanan Samet*. 2008 [cited June 2008]; Available from: <http://www.cs.umd.edu/~hjs/>.
33. Tanin, E., A. Harwood, and H. Samet, *Using a distributed quadtree index in peer-to-peer networks*. The VLDB Journal, 2007. **16**(2): p. 165-178.
34. Mathes, A., *Folksonomies-Cooperative Classification and Communication Through Shared Metadata*. Computer Mediated Communication, LIS590CMC (Doctoral Seminar), Graduate School of Library and Information Science, University of Illinois Urbana-Champaign, December, 2004.
35. ANSI/NISO. *Guidelines for the construction, format, and management of monolingual controlled vocabularies (ANSI/NISO Z39.19-2005)* July 25, 2005 [cited June 2007]; Available from: <http://www.niso.org/standards>.
36. www.loc.gov. *Tools for Authority Control--Subject Headings*. 2008 [cited June 2008]; Available from: <http://www.loc.gov/cds/lcsh.html>.
37. www.nlm.nih.gov. *Medical Subject Headings - Home Page*. 2008 [cited June 2008]; Available from: <http://www.nlm.nih.gov/mesh/>.

38. del.icio.us. *del.icio.us*. 2008 [cited June 2008]; Available from: <http://del.icio.us/>.
39. Tan, P.N., M. Steinbach, and V. Kumar, *Introduction to Data Mining*. 2005: Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.
40. Ying, Z. and G. Karypis, *Evaluation of hierarchical clustering algorithms for document datasets*, in *Proceedings of the eleventh international conference on Information and knowledge management*. 2002, ACM: McLean, Virginia, USA.
41. Ley, M. and P. Reuther, *Maintaining an Online Bibliographical Database: The Problem of Data Quality*. Extraction et gestion des connaissances (EGC'2006), Actes des sixiemes journees Extraction et Gestion des Connaissances, Lille, France, 2006: p. 17-20.
42. portal.acm.org. *ACM Digital Library*. 2008 [cited June 2008]; Available from: <http://portal.acm.org/dl.cfm>.
43. www.springerlink.com. *SpringerLink Home - Main*. 2008 [cited June 2008]; Available from: <http://www.springerlink.com/home/main.mpx>.
44. www.computer.org. *IEEE Computer Society*. 2008 [cited June 2008]; Available from: <http://www.computer.org/portal/site/ieeecs/index.jsp>.
45. ieeexplore.ieee.org. *IEEE Xplore: Guest Home Page*. 2008 [cited June 2008]; Available from: <http://ieeexplore.ieee.org/Xplore/guesthome.jsp>.
46. *DBLP FAQ: What is the meaning of "DBLP"?* 2008 [cited June 2008]; Available from: <http://www.informatik.uni-trier.de/~ley/db/about/faqdblp.html>.
47. *DBLP XML data dump*. 2008 [cited 2008]; Available from: <http://dblp.uni-trier.de/xml/>.
48. www.sigmod.org. *ACM SIGMOD Anthology*. 2008 [cited June 2008]; Available from: <http://www.sigmod.org/sigmod/anthology/index.htm>.
49. Bergmark, D., P. Phempoonpanich, and S. Zhao, *Scraping the ACM Digital Library*. SIGIR Forum, 2001. **35**(2): p. 1-7.
50. www.acm.org. *1998 ACM Computing Classification System*. 2008 [cited June 2008]; Available from: <http://www.acm.org/class/1998/>.
51. Mattox, D., M. Maybury, and D. Morey, *Enterprise Expert and Knowledge Discovery*. International Conference on Human Computer International (HCI 99), 1999: p. 23-27.
52. www.acm.org. *Author Profile Pages in the ACM Digital Library* Association for Computing Machinery. 2008 [cited June 2008]; Available from: http://www.acm.org/membership/author_pages.
53. Aho, A.V. and M.J. Corasick, *Efficient string matching: an aid to bibliographic search*. Communications of the ACM, 1975. **18**(6): p. 333-340.
54. Fisher-Ogden, P. *dblp-expert-finder - Google Code*. 2008 [cited June 2008]; Available from: <http://code.google.com/p/dblp-expert-finder/>.
55. db.apache.org. *Apache Derby*. 2008 [cited 23 June 2008]; Available from: <http://db.apache.org/derby/>.
56. java.sun.com. *Java Architecture for XML Binding (JAXB)*. 2007 [cited 23 June 2008]; Available from: <http://java.sun.com/developer/technicalArticles/WebServices/jaxb/>.
57. java.sun.com. *Java Persistence API*. 2008 [cited 23 June 2008]; Available from: <http://java.sun.com/javaee/technologies/persistence.jsp>.

58. lucene.apache.org. *Solr, an open source enterprise search server*. 2007 [cited June 2008]; Available from: <http://lucene.apache.org/solr/>.
59. alias-i. *LingPipe Home*. 2008 [cited July 2008]; Available from: <http://alias-i.com/lingpipe/>.
60. Rasmussen, M. and G. Karypis, *gCLUTO An Interactive Clustering, Visualization, and Analysis System*.
61. www.cs.umd.edu. *Publication for Hanan Samet*. 2008 [cited July 2008]; Available from: <http://www.cs.umd.edu/~hjs/hjscat.html>.
62. Cutting, D.R., et al., *Scatter/Gather: a cluster-based approach to browsing large document collections*, in *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*. 1992, ACM: Copenhagen, Denmark.
63. White, R.W., et al., *Supporting exploratory search*. *Communications of the ACM*, 2006. **49**(4): p. 36-39.
64. Hearst, M.A., *Clustering versus faceted categories for information exploration*. *Commun. ACM*, 2006. **49**(4): p. 59-61.
65. Yee, K.-P., et al., *Faceted metadata for image search and browsing*, in *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2003, ACM: Ft. Lauderdale, Florida, USA.
66. Alonso, O., M. Gertz, and R. Baeza-Yates, *On the value of temporal information in information retrieval*. *SIGIR Forum*, 2007. **41**(2): p. 35-41.
67. java.sun.com. *JavaServer Faces Technology*. 2008 [cited July 2008]; Available from: <http://java.sun.com/javaee/javaserverfaces/>.
68. facelets.dev.java.net. *facelets: JavaServer Facelets*. 2008 [cited July 2008]; Available from: <https://facelets.dev.java.net/>.
69. www.oswd.org. *Open Source Web Design - Download free web design templates*. 2008 [cited July 2008]; Available from: <http://www.oswd.org/>.
70. www.famfamfam.com. *famfamfam.com: Silk Icons*. 2008 [cited July 2008]; Available from: <http://www.famfamfam.com/lab/icons/silk/>.
71. simile.mit.edu. *SIMILE | Timeline*. 2008 [cited July 2008]; Available from: <http://simile.mit.edu/timeline/>.
72. www.jboss.org. *RichFaces JSF component library*. 2008 [cited July 2008]; Available from: <http://www.jboss.org/jbossrichfaces/>.
73. Wharton, C., et al., *The cognitive walkthrough method: a practitioner's guide*. 1994.
74. Zhang, X., et al., *Fast mining of spatial collocations*, in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2004, ACM: Seattle, WA, USA.
75. Son, E.-J., et al., *A spatial data mining method by clustering analysis*, in *Proceedings of the 6th ACM international symposium on Advances in geographic information systems*. 1998, ACM: Washington, D.C., United States.
76. Harel, D. and Y. Koren, *Clustering spatial data using random walks*, in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. 2001, ACM: San Francisco, California.
77. Guo, D., D. Peuquet, and M. Gahegan, *Opening the black box: interactive hierarchical clustering for multivariate spatial patterns*, in *Proceedings of the*

- 10th ACM international symposium on Advances in geographic information systems*. 2002, ACM: McLean, Virginia, USA.
78. Ester, M., H.P. Kriegel, and J. Sander, *Spatial Data Mining: A Database Approach*. Proc. 5th Int. Symp. on Large Spatial Databases, Berlin, Germany, 1997: p. 47-66.
 79. Yimam-Seid, D. and A. Kobsa, *Expert Finding Systems for Organizations: Problem and Domain Analysis and the DEMOIR Approach*. Journal of Organizational Computing and Electronic Commerce, 2003. **13**(1): p. 1-24.
 80. McDonald, D.W. and M.S. Ackerman, *Expertise recommender: a flexible recommendation system and architecture*. Proceedings of the 2000 ACM conference on Computer supported cooperative work, 2000: p. 231-240.
 81. Balog, K. and M. de Rijke, *Determining expert profiles (with an application to expert finding)*. IJCAI'07: Proc. 20th Intern. Joint Conf. on Artificial Intelligence, 2007: p. 2657–2662.
 82. research.microsoft.com. *W3C Test Collection*. 2005 [cited July 2008]; Available from: <http://research.microsoft.com/users/nickcr/w3c-summary.html>.
 83. Zhu, J., et al., *Mining Web Data for Competency Management*. Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on, 2005: p. 94-100.
 84. www.acm.org. *Author Profile Pages announcement*. April 2008 [cited July 2008]; Available from: <http://www.acm.org/press-room/news-releases/pdfs/author-pages.pdf>.
 85. Diederich, J. *Faceted DBLP*. 2008 [cited July 2008]; Available from: http://dblp.l3s.de/?q=&newQuery=yes&resTableName=query_resultmAaSfe.
 86. dblp.l3s.de. *About FacetedDBLP*. 2008 [cited July 2008]; Available from: <http://dblp.l3s.de/dblp++.php>.
 87. Diederich, J. and W.T. Balke, *The Semantic GrowBag Algorithm: Automatically Deriving Categorization Systems*. Proc. of ECDL, Sept, 2007.