

Bootstrapping Ontology Learning for Information Retrieval Using Formal Concept Analysis and Information Anchors

Guo-Qiang Zhang, Adam D. Troy, and Keith Bourgoin

Department of Electrical Engineering and Computer Science
Case Western Reserve University
Cleveland, Ohio 44106, U.S.A.

Abstract. We present an innovative approach to information retrieval for domain-specific digital library collections. We use a combination of Formal Concept Analysis (FCA) and a notion of *information anchors* to facilitate information delivery to the end user. This approach (1) uses ranked objects in attribute concepts to facilitate topical queries for experts and expertise profiles; (2) formulates (keyword by keyword) context for concept lattice construction via a set of heuristics, including those based on information anchors for selecting descriptive phrases, (3) bootstraps the learning of domain-specific concept hierarchies using FCA, and (4) incorporates the learnt concept hierarchies and WordNet for content-based document classification. To demonstrate the feasibility and utility of this approach, we implemented a prototype online information retrieval system `memsworldonline.case.edu` (MWOL) for the emerging engineering discipline of MEMS (microelectromechanical systems) incorporating these ideas. MWOL has been actively used by a non-trivial group of MEMS practitioners; all user queries are processed in a fraction of a second as a result of inverse indexing strategy using Berkeley DB. Voluntary user feedback using online forms has been encouraging. However, no other systems with similar features are available for a comparative study at this point.

1 Background

The content of the World Wide Web can be divided into two broad categories. One is the “shallow Web”, referring to published Web pages directly accessible by endusers and search engines such as Google. The other, much bigger part of the Web, is the “deep Web”, accounting for an estimated 99% of digital contents in size [2]. These are contents that are not directly accessible: they are either protected by access control or require meaningful query to extract the content through well-defined interfaces. Digital libraries fall into the latter.

The scientific literature makes up the bulk of the contents for digital libraries. Over-time, the majority of the scientific literature published in the past has been digitized and made available through digital libraries. Almost all future publications will be accessible in digital format. Unlike the shallow Web, more contents of the deep Web are semi-structured, *i.e.*, certain metadata information is available across the scientific literature, such as *author, address, title, abstract, date, references*. Because of this, query

interfaces for digital libraries can potentially allow for higher-precision search, targeting contents in specific time-intervals and with specific keywords in title, author, or abstract.

However, traditional digital library retrieval interfaces have not taken advantage of the full potential offered by its semi-structured content. We believe that the structure of digital libraries makes it possible for the creation of a new generation of information retrieval interfaces that offer the user unprecedented precision and recall in content-based retrieval, as well as new ways for information exploration and discovery, such as *expert identification, social network dynamics, and emerging trend detection*. To this end, we present an innovative approach to information retrieval for domain-specific document collections. We use a combination of Formal Concept Analysis (FCA) and a notion of *information anchors* to facilitate information delivery to the end user. This approach (1) uses ranked objects in attribute concepts to facilitate topical queries for experts and expertise profiles; (2) formulates (keyword by keyword) context for concept lattice construction via a set of heuristics, including those based on information anchors for selecting descriptive phrases, (3) bootstraps the learning of domain-specific concept hierarchies using FCA, and (4) incorporates the learnt concept hierarchies and WordNet for improved document classification. To demonstrate the feasibility and utility of this approach, we implemented a prototype online information retrieval system `memsworldonline.case.edu` (MWOL) for the emerging engineering discipline of MEMS (microelectromechanical systems) incorporating these ideas. MWOL has been actively used by a non-trivial group of MEMS practitioners – since the announcement of the beta version to selected MEMS research groups at the end of October in 2005, users from 1475 distinct IP address have used MWOL to process 25824 entries as of Jan. 13th, 2006, and user base is growing at a steady rate. Considering that our users are domain experts and our system *ranks* their expertise in various MEMS topical areas, we feel that the positive comments we have received through our online web form, coupled with the lack of complaints so far is a rather positive sign. However, no other systems with similar features are available for a quantitative comparative study at this point.

Related work. We refer to [29] for an informative survey of the applications of FCA in several areas of information retrieval. We briefly mention three pieces of work that are relevant to some components of our integrated approach. First, in [10], Cimiano, Hotho and Staab use FCA for learning concept hierarchies and compare their approach to other clustering and learning methods on several specific domains. Although both [10] and our work involve FCA, the particular context of the applications and the details of FCA's usage are different in many ways. Our work is motivated by the recognition that the incorporation of concept hierarchies can bring improved precision for document classification. On the other hand, Cimiano, Hotho and Staab consider the learning of concept hierarchies as a stand-alone topic, without explicitly pursuing any specific applications in [10]. This higher-level distinction entails differences in specifics. For example, Cimiano, Hotho and Staab make substantial use of Natural Language Processing (NLP) techniques for parsing *unstructured* text corpora but we take advantage of the *semi-structured* data derivable from the notion of information anchors, and we use off-the-shelf NLP tagging programs. Because of this, the FCA context we use for concept hierarchy learning is of the type `keyword×keyword`, instead of the more standard

document×keyword context in [10] – documents play an implicit role in our formulation of information anchors. This way, an iterative bootstrapping process naturally fits our approach. Moreover, our approach has been implemented as part of the MWOL system, which incorporates several different kinds of FCA contexts, related by the database join operation, for attribute-concept retrieval. In other work, Bloehdorn, Cimiano, and Hotho [3] explore ontology learning for use in text classification and clustering applications. Their experiments reveal that automatically learned ontologies provide classification and clustering improvements similar to those gained with manually created ontologies. Earlier work by Bloehdorn and Hotho is also relevant [4].

Second, since we rank objects in queried attribute concepts, numerical values are unavoidable. Our relevance ranking approach is somewhat related to multi-value concepts [12] or fuzzy concept analysis [15, 30]. In our approach presented in this paper, values for relevance ranks are either used for the ordered presentation of results to the user, or for thresholding to obtain a standard context, based on which a concept hierarchy is constructed. Thus the particular features of conceptual scales are not used in our approach at this point.

Third, keyword selection is an important topic in FCA-based information retrieval. This topic has been explored in a number of settings (e.g. [9, 16]). [9] demonstrates the advantage of noun phrases in selecting attributes for organizing the presentation of retrieved results using FCA and evaluates the performance of such a selection strategy using certain lattice-based measures. In contrast, we use the notion of information anchors in evaluating the quality of a keyword without an in-depth NLP analysis and feature this technique in a working information retrieval system. The idea of incorporating background knowledge not captured by co-occurring keywords for document clustering [16] is related to our incorporation of concept-hierarchies for ranking document relevance and classification. While the goal of clustering is to group related documents into clusters, the task of classification is to determine the relevance of a specific document to a specific query topic supplied by a user.

The rest of the paper is organized as follows. Section 2 introduces the notion of information anchors. Section 3 briefly overviews the roles of FCA in our approach, followed by heuristics for descriptive keywords in Section 4, and the bootstrapping of ontology learning in Section 5. Section 6 provides an overview of our search engine environment MWOL incorporating the techniques described in earlier sections.

Remark. For readability, the terms concept hierarchy, ontology, and taxonomy are used as synonyms. The relationships among these terms should really be carefully delineated, though they are beyond the scope of the current paper (for further discussions we refer to [13, 14]).

2 Information Anchors

We approach the problem of information retrieval (IR) for special-domain document collections using a combination of FCA and the informal notion of *information anchors*. Our notion of information anchors is motivated in part by the idea of *agents* in Artificial Intelligence (AI). While agent is a useful informal concept that helps the

understanding and implementation of AI systems by associating a certain degree of autonomy or boundary to subsystems that “process and react” to information, information anchors are entities that “possess and weave” information (by combining the active and the passive). This is again a highly *informal* notion. A typical example of information anchor is an article, which pulls together views and findings around a central theme. Another example of information anchor is a Web page, usually with hyper links attached to it.

There are other kinds of information anchors, concrete or abstract, that have not occupied a central role in traditional IR. These include researchers and institutions (or organizations), as well as hubs in social networks (CiteSeer [5] is an example of social network – citation network). Formal concepts from FCA can be considered as information anchors as well.

The emphasis on a broader scope of information anchors has several advantages. For example, instead of literally returning literatures containing certain string patterns in the traditional sense of “search”, one can elevate it to the level of more organized return surrounding the information anchors of *researchers* and *institutions*. We call this *multi-perspective search*.

	Topic area	Researcher	Literature	Institution
Topic area		X	X	X
Researcher	X		X	X
Institution	X	X	X	

Table 1. Multi-perspective search.

Table 1 represents three search/query modes: query by topic, query by researcher, and query by institution. For each query mode, a user can select among several organized presentations for their search results. For example, for query by topic area, a user can input a free (i.e. not provided by the system) topical phrase such as “accelerometers” with the option of retrieving any of the following type of results, which are not available in traditional digital library interfaces or search engines:

- a ranked list of names of relevant authors/experts;
- a ranked list of relevant published literatures/citations; and
- a ranked list of relevant institutions associated with the topic area.

Another advantage is a new class of heuristics based on information anchors. For example, when it comes to the selection of descriptive keywords and phrases in Section 4, traditional document-based ranking alone does not provide satisfactory results. Instead, by passing through researchers as information anchors and then assessing the distribution and significance of word-association and occurrence profiles, we obtain much improved results.

Focusing on information anchors also allows us to look at individual research profile, institution research profile, community dynamics [34], and emerging trends [18].

3 Roles of Formal Concept Analysis

We briefly recall some basic terminologies of formal concept analysis and refer further details to [12]. A (*formal*) *context* is a triple (G, M, I) where G is a set of *objects*, M is a set of *attributes*, and $I \subseteq G \times M$ is an incidence relation. Given $O \subseteq G$ and $A \subseteq M$, we define $O' := \{m \in M \mid (g, m) \in I \text{ for all } g \in O\}$ and $A' := \{g \in G \mid (g, m) \in I \text{ for all } m \in A\}$. A *concept* of (G, M, I) is a pair (O, A) such that $O = A'$ and $A = O'$. O is called the *extent* and A the *intent* of the concept. Since the extent and intent of a concept determine each other uniquely, it suffices to refer to one of them. Of particular interest to us is the so-called *attribute concept* and *object concept*. A concept (O, A) is called an attribute concept if $A = \{a\}$ for some $a \in M$, and an object concept if $O = \{o\}$ for some $o \in G$. The central result of FCA is that contexts can be used to represent complete lattices. For any context (G, M, I) , the mapping $(\cdot)'' : 2^G \rightarrow 2^G$ constitutes a closure operator on the powerset 2^G .

Formal concept analysis plays an important role in several aspects of our approach. Since title, author, address, and abstract text are extracted for each entry in our MEMS collection, we already obtain a collection of contexts in the sense of FCA. Classification of the collection according to keywords provides another context, which can be combined with the other contexts using standard database join operation to implement additional features. For example, the query response of a list of authors for a topic area amounts to the derivation of the objects of an attribute concept for the context type `keywords×(abstract id)`, joined with a context type `(abstract id)×author`. However, such contexts are kept *implicit* in our system through the document classification function. Moreover, the ranking of objects in the retrieved attribute concept is a useful feature not considered as part of FCA.

Another use of FCA for our approach is for learning concept hierarchies with respect to the specific document collection in order to improve precision for document classification. A simple example can help illustrate this idea. A news article on a baseball game should be classified to sports, even though (and quite possibly) “sport” is not explicitly mentioned *anywhere* in the article. Humans are able to perform such classification correctly because of acquired taxonomical knowledge. For information retrieval systems to be able to do the same, external knowledge such as this must be used. Because of its philosophical, mathematical and algorithmic grounding (as well as numerous successful case studies for conceptual modeling), FCA is a natural candidate for automatic learning of concept hierarchies or taxonomies.

We describe the utility of these aspects of FCA in more detail in subsequent sections.

4 Constructing a Descriptive Keyword Set

Before a concept hierarchy can be built we must select topical keywords to be used for the construction of a formal context. The keywords must be good indicators of topics within the domain of interest and must be sensitive to the specific document collection. Keywords used in this work fall into two distinct categories: (1) those assigned by the authors of the documents and (2) those extracted with our heuristic methods. We

utilize both types of keywords for the following reasons. Author assigned keywords are used not only because they are likely to be accurate descriptors of content but also because they often include higher-level keywords that are not specifically mentioned in the document itself. However, author assigned keywords often do not include the most specific keywords of the domain, particularly those that describe the specific research presented in the documents. For this reason, it is also important to extract additional keywords directly from the source.

We use “terms”, “phrases” and “words” interchangeably except where a more specific meaning is clear. In this context they refer to a string containing *one or more* words. “Keywords” is used specifically for those words selected by the following techniques as topical keywords for the domain of interest.

4.1 Author-Assigned Keywords

Although author assigned keywords are useful, they cannot be used in their raw state. The most glaring issue is the sheer number of keywords. In our corpus, only 20% of the documents have assigned keywords, yet these relatively few documents have about 10,000 unique keywords. A concept hierarchy constructed from such a large collection keywords would not be useful for the MEMS field. As a comparison, the National Library of Medicine’s MeSH (medical subject headings) contains just under 23,000 descriptors to comprehensively capture the rich medical field. Additionally, 86% of the author-assigned MEMS keywords are assigned only to a single document. By limiting author assigned keywords to the most common ones, we can limit them to a manageable amount. We use only those keywords which are assigned to a minimum number of documents, as a small percentage of the total number of documents, around 0.01% – 0.1%. Those keywords which are assigned to multiple documents have been given a sort of implicit census by the authors as being useful for the field.

4.2 Keywords Derived from Documents

We obtain the second set of keyword phrases from the document collection by first constructing a candidate list of all potential key words, and then distilling it using several heuristics.

We limit phrase length to four words, which seems to be a good compromise between being long enough to include most worthwhile phrases as well as limiting the number of possible phrases to a reasonable amount. Phrases are considered to be any string found in the text containing at most four consecutive words and not containing any stopwords. For this work we use a relatively aggressive list of stopwords, for this domain in particular we use the NASA ADS physics stopword list [25] which is used by the NASA digital library. Essentially, phrases are those series of consecutive words that occur in between stopwords up to four words long. Additionally each word in the phrase is stemmed using Burden’s stemming algorithm [8]. Burden’s is used rather than Porter’s because the Burden algorithm attempts to reduce words to their readable english root, where the Porter algorithm makes no attempt to produce proper english words. It is helpful to have actual english words in this context because the ontology generated with the stemmed words may need to be evaluated by a domain expert. For

example, given “displays”, the Porter algorithm produces “displai” where the Burden algorithm produces the true English root “display”.

Once a raw list of keyword candidates is formed, we perform further selection by requiring candidate keywords to pass a certain threshold using a combination of the three heuristics discussed below. There is a great deal of previous research dealing with automatic keyword extraction [17, 23, 27]. Our simple and yet robust heuristics takes advantage of *information anchors* to achieve a reasonably sized yet quite descriptive set of key terms.

The first type of heuristic is statistical. The intuition behind this heuristic is that *any established sub-topic of a field will be practiced by several distinct authors*. We take advantage of this idea by using *individual authors as information anchors* and comparing expected term occurrence with the actual term occurrence for each author. The occurrence of terms in each author’s collections of documents is analyzed independently. The actual occurrence value of terms is calculated as a percentage of the number of papers in which each term appears for each author who has authored a minimum number of documents. This minimum is necessary because an author with only a few papers could have artificially high occurrence values. Term frequency percentages from the entire document set are used as the expected term frequency values for comparison with the term frequencies each author. If greater than a set threshold of authors have the term with an occurrence value greater than the global occurrence value scaled by some constant it is accepted as a keyword passing this test. It is important to require enough authors to have the keyword in order to eliminate words that occur often simply due to author’s individual writing styles. The constant and threshold used here are interrelated. If a high scaling constant is used, fewer authors will be identified as having the keyword, so a lower threshold will need to be used. The threshold and constant values used here were selected through experimentation, although our approach is amendable to a variety of optimization techniques.

The second type of heuristic is grammatical. Certain parts of speech, particularly nouns and noun phrases are more likely to be useful topical keywords. Noun phrases are phrases that include the noun and any noun modifiers. Though most topical keywords are noun phrases, there are some exceptions. In the MEMS domain for instance, manufacturing processes such as “wafer bonding” are an import subtopic, though they are not described by noun phrases. For part of speech tagging we use the MontyLingua Suite of natural language tools [21]. The MontyLingua part of speech tagger is based on Brill’s tagger [7] while additionally incorporating “common-sense” knowledge. We use MontyLingua to identify all nouns, noun phrases, and sub-phrases in the corpus.

The final heuristic relies on term positions within the documents. The intuition behind this heuristic is that the most important topical keywords for any particular domain will occur in prominent places within several of the documents in corpus. There are several possible ways to exercise this intuition. The simplest method, and the one used in this work is to consider phrases occurring in the title as the most important phrases in a document. Additionally, titles are less likely to contain extraneous, non-useful terms. Whereas this method is discrete in the sense that a phrase is either prominent or not, another possible method is to consider prominence as a continuous variable where the prominence of a phrase is directly related to the its distance from the beginning of the

document. The logic behind this is simple: the most important topics in a document are generally stated early in the text, though this method is more useful for short documents or document summaries such as abstracts, as is the case here. A final possible group of techniques consist of sentential structure analysis tools [33] and summary methods to identify the most prominent phrases. We chose not to use this due to its increased complexity and dependence on the particular methods chosen.

5 Bootstrapping Ontology Learning

The ontology learning process takes place in four iterated steps (Fig. 1). First, the context is generated from the document collection with keyword assignments. Next, the context is subjected to formal concept analysis, which generates the concept lattice and in turn the ontology. At this point, the ontology may or may not be fine tuned by a domain expert. Lastly, the documents are reclassified based on information revealed by the new ontology. This series of steps is then repeated until an acceptable ontology is produced. Keyword selection has been described in Section 4. The remaining steps are discussed in more detail below.

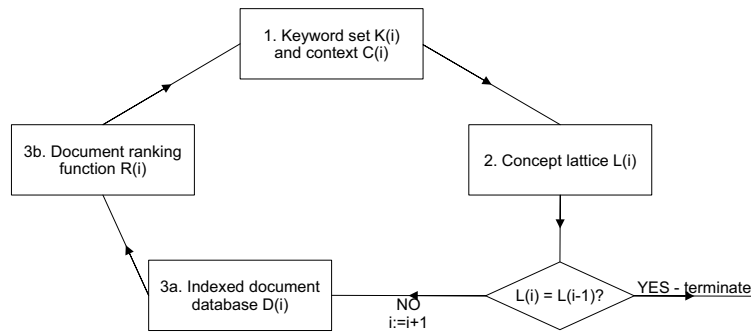


Fig. 1. The iterative steps involved in bootstrapping.

5.1 Context Generation

Once keywords have been selected, the context can be generated. We evaluated several methods for context generation using various information anchors. They include: (1) using documents as objects and keywords as attributes, (2) using keywords as objects and co-occurring words as attributes, and (3) using keywords as both objects and attributes. In the first method, each document in the collection is an object and any keywords that have been assigned to the document are considered to be attributes of the object representing the document. The second method consists of using each keyword identified above as an object and words that occur in documents that have been assigned the keyword and meet a certain threshold as the attributes of each object. The final method uses

keywords as objects where each object has a particular keyword as an attribute if that keyword occurs along with the object keyword and meets a specified support threshold. The first method uses the papers as information anchors while the second and third methods use keywords as information anchors. By building the context using information anchors we are able to gain more information about the keywords. Additionally, the final two context types are produced using a sort of “fuzzy” context that is then converted into a standard binary context. In the fuzzy context, rather than objects simply either having a certain attribute or not, the relationship between them is indicated by a continuous value which is defined by the number of co-occurrences between the object and attribute, similar to the idea of conditional probability (a sample continuous context is shown in Table 2). This fuzzy context is then converted into a standard binary context by using some threshold for attribute membership (the corresponding binary context for a threshold of 0.2 is shown in Table 3).

	Materials	Fabrication	Devices	MEMS	polysilicon
Silicon	0.21	0.27	0.32	0.13	0.07
Accelerometers	0.09	0.26	0.42	0.16	0.08
Wafer Bonding	0.13	0.33	0.31	0.25	0.05
Nickel	0.27	0.16	0.2	0.08	0.01
Etching	0.15	0.39	0.3	0.14	0.08

Table 2. A portion of a continuous context for concept hierarchy learning in MEMS.

	Materials	Fabrication	Devices	MEMS	polysilicon
Silicon	1	1	1	0	0
Accelerometers	0	1	1	0	0
Wafer Bonding	0	1	1	1	0
Nickel	1	0	1	0	0
Etching	0	1	1	0	0

Table 3. Corresponding portion of binary context after thresholding the context in Table 2.

We have chosen to use the third method for several reasons. Most importantly, as discussed, it utilizes a support threshold for the relationship between the keywords. Using the first method, even keywords that occur together only one time must be incorporated in the lattice (and the ontology in turn) as having a relationship of some sort. Keywords that co-occur only a limited number of times are not likely to be related in any significant way. Additionally, this context allows for relationships between keywords that may not be symmetrical. In the MEMS domain, for example, nearly 100% of occurrences of the term “carbide” co-occur with “silicon”, indicating a strong relationship, while less than 5% of the occurrences of “silicon” co-occur with “carbide”,

indicating a weak relationship (if any). Finally, the drawback of the second method is that we have no assurances of the quality of the words that are used as attributes. Any word co-occurring with a keyword that meets the support threshold can be used as an attribute, while all of the keywords have been carefully selected.

5.2 FCA and Ontology Construction

After constructing the context the concepts and concept lattice can be extracted using formal concept analysis. Several algorithms exist for formal concept analysis including algorithms by Ganter, Bordat, Lindig, and others [11, 6, 20, 19]. Here we have chosen to use Lindig’s algorithm due its efficiency in generating the concepts and concept lattice simultaneously, both of which are needed for this work and its relative ease of implementation. In comparison, Ganter’s algorithm first generates the concepts and then the lattice can be constructed from the concepts. Lindig’s algorithm begins with the meet of the lattice, generates its upper neighbors then continues in the same fashion, generating upper neighbors until the lattice is complete. Additionally, it is possible to reverse order of lattice construction and by doing so only generate the first several levels of the lattice if desired. We implemented Lindig’s algorithm in Python for this study. In the early stages of implementation we considered using OWL (Web Ontology Language [28]) to represent the generated concept hierarchies. As development progressed we found that OWL would be an over-kill at this stage, although we plan to revisit this in the future.

5.3 Tuning by Domain Expert

At this point it is possible to interject human expertise into the process. Human intervention is most useful during the first iteration of the process, when the ontology is completely automatically generated. This is the most feasible point at which human knowledge can be included. Compared to the size of the corpus, the generated context will be relatively small, so a domain expert can manually review. On the other hand, the size of the document collection is too large for the accuracy of document classification to be manually validated.

There are two particularly important and manageable contributions a domain expert can make at this point. First is the inclusion of the top level of the ontology (see Fig. 2 for a MEMS top level concepts).



Fig. 2. A stable top level MEMS ontology. Synonyms derived from WordNet are not displayed.

In the MEMS literature it is the top level topics that are most often not mentioned in documents to which they belong. These top level topics are implied by the keywords that are mentioned in the document. Knowledge of such implications is often assumed

of the readers by the author, particularly in technical scientific literature. The inclusion of the top level domain can greatly improve the quality of the ontology. The second key contribution is the identification of synonyms and conceptual synonyms. Synonyms are difficult to extract from the texts because they do not often occur together; an author simply chooses to use one or the other. Synonyms will often, on the other hand, occur in the same level of the ontology because they will have similar upper and lower neighbors.

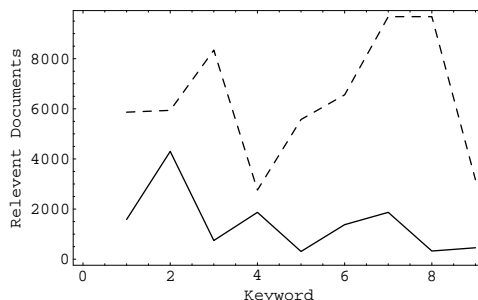


Fig. 3. Differences in term relevance before and after the incorporation of a concept hierarchy.

5.4 Reclassification

After having built and possibly tuned the ontology, the new knowledge of concept relationships must be returned to the document collection. This is done by reclassifying the documents based on the new ontology. Reclassification occurs in two stages. First, keyword assignments are propagated up through the ontology. This is simply based on properties of ontology. If a document is relevant for a given topic, it would then certainly be relevant for all super-topics of the given topic. As an illustration, if a document is relevant to “accelerometers”, it should also be relevant to “devices”, the super-topic of “accelerometers”. Second, a new set of classification rules is generated based on the newly assigned keywords and the documents are reclassified. After the process of reclassification, a new ontology can be constructed based on the newly classified document collection by repeating the steps above. Figure 5.3 shows the number of documents relevant for nine sample search terms before using the learned ontology and after using the ontology. This results in increased number of documents for some terms and relatively smaller increases for other. Such an increase can arise for multiple reasons. Topics that are often implied will gain many more relevant documents. Also those topics that have many sub-topics will gain more relevant documents. In rare cases it might be possible for the number of relevant documents to decrease after reclassification. As example, a portion of an ontology is shown in Figure 4.

Initially, before using the ontology, only the documents that actually have the string “silicon” are relevant to silicon. When the ontology is used, documents including “polysilicon” and “silicon carbide” would also be relevant. With this new knowledge,

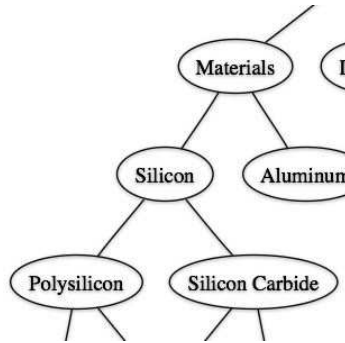


Fig. 4. A concept neighborhood of “silicon”.

classification rules would look for more than just the term “silicon”, therefore some the documents that were originally the most relevant before the ontology, may now be some of the least relevant if they do not include additional elements from the new classification rules. In other words, when concept neighbors are taken into account, the impact of the central node is decreased. In our experiments we used 4 iterations for the process described in Fig. 1. The convergence or optimal number of iterations are topics warranting further study.

When visualization is desired, such as for evaluation of the concept lattice by a domain expert, ConExp [35] was used. Though not particularly fast, it allows for interactive browsing of the lattice.

6 Prototype System

The beta release of MWOL (memsworldonline.case.edu) provides a simple and yet powerful interface for the query types given in the first row of the Table 1. To make the intended retrieval results more explicit, these queries are marked by “Experts in ...”, “Organization for ...”, and “Literature on ...”.

The “Experts in ...” option is selected at entry point by default. If a user now enters “accelerometers”, the output will show a ranked list of experts together with their relevance scores. If the user then clicks on an individual’s name, say “Khalil Najafi”, the related publications by the individual are displayed as a ranked list. Clicking on any paper items leads to the display of the title, abstract, and citation for this article.

We chose MEMS because this emerging engineering discipline is relatively new (less than three decades); it has a dedicated community of researchers and practitioners; and it has focused venues for its scientific publications. There are no other digital libraries or search engines focused on this area so such a digital resource would be useful in itself.

Our data source consists of about 20 journals and conference proceedings in MEMS. About 30,000 abstracts and 35,000 authors from these sources are currently used for the beta release (yes there are more authors than documents in our database; some

of the authors can be identified as the same person moving from one institution to another). The system runs on an Apple X-serve with an 80GB HD and 2GB of RAM. The responses for all queries are *instantaneous* (i.e., without noticeable delays other than possibly network traffic) due to various performance optimization techniques. The implementation details will be presented elsewhere.

Acknowledgment. We thank our MEMS domain-expert Mehran Mehregany for valuable inputs and suggestions on many aspects of the project.

References

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, *Proc. 20th Int. Conf. Very Large Data Bases*, pages 487–499. Morgan Kaufmann, 12–15 1994.
2. M. Bergman. The Deep Web: surfacing hidden value.
<http://www.beta.brightplanet.com/deepcontent/tutorials/DeepWeb/index.asp>
3. S. Bloehdorn, P. Cimiano, and A. Hotho. Learning ontologies to improve text clustering and classification. In M. Spiliopoulou, R. Kruse, A. Nürnberger, C. B., and W. Gaul, editors, *From Data and Information Analysis to Knowledge Engineering: Proceedings of the 29th Annual Conference of the German Classification Society*, Springer, 2006.
4. S. Bloehdorn and A. Hotho. Boosting for text classification with semantic features. In *Proceedings of the Workshop on Mining for and from the Semantic Web at the 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 70–87, 2004.
5. K. Bollacker, S. Lawrence and C. Giles. CiteSeer: An autonomous web agent for automatic retrieval and identification of interesting publications. In *Agents '98*. 1998.
6. J. P. Bordat. Calcul pratique du treillis de galois d'une correspondance. *Math. Sci. Hum.*, 96:31–47, 1986.
7. E. Brill. A simple rule-based part-of-speech tagger. In *Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing*, pages 152–155, 1992.
8. J.P.H. Burden. Stemming algorithms and their use. Technical report, University of Wolverhampton, School of Computing and Information Technology, Search Engine Evaluation and Development Research Group, November 2000.
9. J. Cigarrán, A. Peñas, J. Gonzalo, and F. Verdejo. Automatic selection of noun phrases as document descriptors in an FCA-based information retrieval system. In B. Ganter and R. Godin (Eds.): *ICFCA 2005*, LNCS Vol. 3403, pp. 49 - 63, 2005.
10. P. Cimiano, A. Hotho and S. Staab. Learning concept hierarchies from text corpora using formal concept analysis. *J. Artificial Intelligence Research* Vol. 24, pp. 305 - 339, 2005.
11. B. Ganter. Two basic algorithms in concept analysis, fb4-preprint no. 831. Technical report, TH Darmstadt, 1984.
12. B. Ganter and R. Wille. *Formal Concept Analysis*. Springer-Verlag, 1999.
13. N. Guarino and R. Poli (eds.) *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer Academic Publishers, Pargue, Boston, 1993.
14. N. Guarino. Understanding, Building, and Using Ontologies. *Int. J. Human and Computer Studies*, 46, pp. 293 - 310, 1997.
15. C. Herrmann, S. Hölldobler and A. Strohmaier. Fuzzy conceptual knowledge processing. *Proceedings of the 1996 ACM symposium on Applied Computing*, Philadelphia, pp. 628 - 632, 1996
16. A. Hotho, S. Staab and G. Stumme. Text clustering based on background knowledge. AIFB Technical Report No. 425, University of Karlsruhe, 2003.

17. A. Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, 216–223, 2003.
18. A. Kontostathis, Leon M. Galitsky, William M. Pottenger, Soma Roy, and Daniel J. Phelps. A survey of emerging trend detection in textual data mining. In Michael Berry, editor, *A Comprehensive Survey of Text Mining*. Springer-Verlag, 2003.
19. S. O. Kuznetsov and S. A. Obiedkov. Comparing performance of algorithms for generating concept lattices. *J. Exper. & Theo. Artificial Intelligence*, Vol 14, 189–216, 2002.
20. C. Lindig. *Working with Conceptual Structures - Contributions to ICCS 2000*, chapter Fast Concept Analysis. Shaker Verlag, 2000.
21. H. Liu. Montylingua: An end-to-end natural language processor with common sense. Available at: <http://web.media.mit.edu/~hugo/montylingua/>, August 2004.
22. R. Mandala, T. Tokunaga, and H. Tanaka. Complementing WordNet with Roger’s and corpus-based Thesauri for Information Retrieval. *EACL’99*, 1999.
23. Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *Int. J. AI Tools*, Vol. 13, 157–169, 2004.
24. G. Miller. WordNet: A lexical database for English. *CACM*, Vol 38, 39 - 41, 1995.
25. NASA ADS stopword list. http://adsabs.harvard.edu/abs_doc/stopwords.html.
26. M. Olson, K. Bostic, M. Seltzer. Berkeley DB. In *Proc. 1999 Summer Usenix Technical Conference*, Monterey, 1999.
27. M. Ortu no, P. Carpena, P. Bernaola-Galván, E. Mu noz, and A. M. Somoza. Keyword detection in natural languages and dna. *Europhysics Letters*, Vol 57, 759–764, 2002.
28. <http://www.w3.org/2004/OWL/>
29. U. Priss. Formal concept analysis in information science. *Annual Review of Information Science and Technology (ARIST)*, Vol. 40., 1996.
30. T. Quan, S. Hui, A. Fong, and T. Cao. Automatic generation of ontology for scholarly semantic web. *Proceedings of ISWC*, LNCS Vol 3298, 726–740, 2004.
31. S. E. Robertson, S. Walker, M. Beaulieu. Okapi at TREC-7: automatic ad hoc, filtering, VLC and interactive track. *Proc. 7th Text REtrieval Conference (TREC-7)*, 253–264, 1999.
32. G. Salton. *Automatic Information Organization and Retrieval*. McGraw-Hill, 1968.
33. G. Salton, J. Allan, and A. Singhal. Automatic text decomposition and structuring. *Information Processing and Management*, Vol 32, 127–138, 1996.
34. A. D. Troy, GQ Zhang, and M. Mehregany. Evolution of the Hilton Head Workshop research community. In *Education Digest of the 2006 Solid-State Sensor and Actuator Workshop*, June 2006.
35. S. A. Yeviusenko. System of data analysis “concept explorer”. In *Proceedings of the 7th national conference on Artificial Intelligence KII-2000*, 127–134, 2000.
36. GQ Zhang, G. Shen, Y. Tian, and J. Sun. Concept Analysis as a Formal Method for Web-Menu Design, *The 12th International Workshop on Design, Specification and Verification of Interactive Systems (DSVIS’05)*, Springer Lecture Notes in Computer Science, Vol 3941, to appear.
37. GQ Zhang and Y. Tian. ACOSys: an experimental system for automated content organization. Common Semantics for Sharing Knowledge: Contributions to the 13th International Conference on Conceptual Structures, Kassel University Press, 186 - 198, 2005.
38. GQ Zhang and G. Shen. Approximating concepts, Chu spaces, and information systems. In de Paiva and Pratt (Guest Editors), *Special Issue on Chu Spaces and Applications, Theory and Applications of Categories*, 2005. In Press.
39. GQ Zhang. Chu spaces, formal concepts, and domains. *Electronic Notes in Theoretical Computer Science*, Vol. 83, 16 pages, 2003.