Data Mining for Thesaurus Generation in Informal Design Information Retrieval

Maria C. Yang, William H. Wood, and Mark R. Cutkosky
Center for Design Research
Stanford University, Stanford, CA 94305

I. Abstract

Design is the evolutionary process of transforming informal requirements into a product. Whether this product is real or virtual, mass produced or made to order, the information development that produces this transformation is generic. The task we focus on here is that of making textual information more useful throughout the design process.

Design information can cover a range of different activities within the design process, from conceptual design to implementation. We focus on information captured in electronic notebooks. The informal information found in these notebooks is of particular interest because it is captured as it is generated, in effect, capturing the design process. In this paper, we examine techniques for improving access to such informal design information using hierarchical thesauri overlaid on generic information retrieval tools.

We examine several approaches to creating thesauri, using both manual and data mining techniques, and compare the tradeoffs of each strategy. Our main goal is to develop methods for search and retrieval that allow designers and engineers access to past information and encourage design information reuse.

II. Introduction

With the advent of the Internet and the incorporation of software tools into everyday work, more and more knowledge is being captured and stored electronically. This bounty of information presents an opportunity to provide users with not only access to previously generated ideas, but the potential to analyze, manipulate and leverage this data in unprecedented ways. While numerous methods exist for searching and retrieving archived information, many existing approaches may not be as efficient or resourceful as they could be when applied to specific domains, such as engineering design.

In this paper, we focus on information from the mechanical engineering domain and ways of encouraging its reuse. In particular, we look at the information that is generated during the design cycle, from concept and requirements to manufacturing. Knowledge of past design cycles can be of great value to future designers and engineers, providing insight on past work. An understanding of previous design explorations can prevent costly duplications in design effort, as well as suggest new paths to be taken. However, Liang, et al. [1998] report that simply archiving informal design information results in little reuse.

Furthermore, this paper will discuss the nature of design information, whether it is formal or informal. Formal documents include final reports which cover an entire project, while informal documents are items such as design logbooks and notes. Design logbooks and notes are of special interest because they capture information as it is created, covering design information comprehensively. In this paper, we examine electronic design notebooks (described in detail later), which offers electronic capture of design process knowledge. This is in comparison to final report style material, which is often a distillation of a final design, and may miss much process information [Wood et al 1998]. This paper concentrates on the latter type of information, focusing on information captured from the design process itself.

Our approach for searching for design process information is to use thesauri to both expand the search for design information, and limit the search to the design domain. Our strategies for developing thesauri: manually building thesauri using both informal, design process information and formal, post-process information, and generating thesauri using a data mining approach on document-vector representations of information. There are a number of trade-offs with each type of thesaurus, especially concerning the resources required to build them as well as their ability to enhance information retrieval performance.

The next section of this paper is a description of related work, including research in traditional information retrieval and more recent work in engineering information capture and retrieval. This is followed by a detailed explanation of our approach to engineering information retrieval, with particular focus on thesaurus creation. Finally, results of testing these thesauri on several electronic design notebooks are compared, presented and discussed.

III. Related Work

Engineering Information Capture

Information capture is increasingly becoming the focus of attention in computer-aided design. Major CAD vendors offer data 'safes' or 'warehouses' in an effort to store everything about a design. Unfortunately, most CAD applications capture geometric data without capturing information as to why something is shaped or arranged the way the designer has specified. Parametric CAD (e.g. Pro/Engineer, AutoCAD Mechanical Desktop, etc.) provides a way of capturing design relationships as mathematical constraints among geometrical entities. Fruchter et al. [1996a] take this idea further by integrating graphical views of a design with behavioral models and logical constructs. In addition, Fruchter et al. [1996b] extend this method to coordinate informal hypermedia information with CAD. Dong et al. [1995] describe 'smart drawings' in a similar vein, linking geometric and graphical objects to memos and notes.

In addition to methods which integrate information into a CAD environment, efforts have been made to model design information more generically. Based on the IBIS (Issue-Based Information System [Kunz and Rittel, 1970]) model for design information flow, McCall et al. focus on access to the information that is developed throughout the design process through virtually-structured hypertext [McCall et al., 1991]. The IBIS structure of issue-position-argument has been a fruitful basis for many other design information systems [Conklin and Begeman, 1988; Ullman, 1994; Qreshi et al. 1997]. Hong et al. [1995] describe PENS (Personal Electronic Notebook with Sharing), a design notebook promoting personal reflection that can also be shared among team members. The overriding tenor of all of this research activity is the ability to capture design nuances through much richer representations (most commonly free text) than can be easily accomplished through machine-understandable data coding. The next obvious step is then to consider how one can gain access to this information.

Information Retrieval

The task of mapping a query into a set of possibly relevant documents is referred to as *information retrieval* (IR). Eschewing natural language interpretations of the query and the documents, the common technical method of IR is to map textual language into symbol vectors which can be easily manipulated mathematically. The result set generated by IR is an ordered list of documents which likely contain information that the user has specified.

Supported in large part by the intelligence community, IR has focused on routing information from worldwide news sources into pre-defined channels, human [TREC] or computerized [MUC].

Because for this application queries are pre-defined, much of the research relates to methods for enhancing query representation. Exemplars of good results have become an important tool toward the application of machine learning methods to information retrieval. Given that routing and machine learning are such fertile ground for research, much work on the ad-hoc querying side has converged to some standard methods which are mostly fine-tuned versions of mature techniques.

Design Information Retrieval

Like the application of IR techniques in intelligence gathering, the context of design provides an opportunity to to customize the basic methods of information retrieval. Three main properties of design have seeded research: designers' questions tend to fall into distinct categories, designers tend to consider issues generic to their discipline, and design information tends to be centered around an artifact which can form a backbone for search.

To develop Dedal, Baudin et al. [1991] collected questions asked by design teams faced with a redesign task. Questions tended to fall into a general pattern of descriptor-subject pairs where generic design (and design process) concerns modify subjects drawn from the physical or functional aspects of the actual design. An indexing method was used in which design information was labeled by the intersection of two hierarchical representations, one for descriptor and one for subject. Queries then operated deterministically over this index to retrieve information, resulting in near-perfect IR performance. The cost for this is a both a large expense for hand-indexing design information and the static nature of the indexing model which does not easily adapt to an evolving design model.

Wood et al. [1998] describe a method based on typical IR techniques, extending them by mapping queries into a hierarchical thesaurus of generic life-cycle design issues. When searching within a single design context, this mapping demonstrates no loss of performance when the original query is replaced by design issue. When searching across contexts, the thesaurus improves performance two-fold. Yang and Cutkosky [1997] demonstrate results from applying generic thesauri in a Boolean term matching mode.

Dong and Agogino [1997] use machine learning techniques over IR representations of design documents to induce a directed graph of relationships among design concepts. The method used tags design information by part of speech, extracts noun phrases (generally considered to carry most of the information in the text), and finds coocurrences among them. These coocurrences are then fed into a Bayes-network learner to extract relationships among concepts.

In this paper, we extend design information retrieval by taking aspects of each of these three lines of research. From Dedal, we take the model of descriptor-subject querying and the notion that there are generic design descriptors. From Wood et al., we take the information retrieval focus and reinforce the idea that a generic thesaurus is of value across design contexts. Finally, from Dong and Agogino, we take the notion that we can extract meaningful design representations from design text. We will now describe our methods in detail.

IV. Experimental Methods

Information Retrieval Method

In the research described here, the SMART system [Salton 1988] is used to test out several strategies for gaining access to design information. The basics of the system include query/document representation, indexing, and similarity measurement:

Representation: Both user queries and the documents held within a corpus are each represented in the same way, through a vector of weights associated with the words they contain.

Indexing: Documents within a corpus are indexed as a group. As each document is read, each term that occurs is inspected. If it is a common word, it may be discarded. If it is kept, it is stemmed (common prefixes and suffixes are removed) and entered into the dictionary. The term count for the document (*tf*) is then incremented. If it is the first occurrence of the term in a document, the document count (*df*) for that term is incremented. Finally, all of this information is synthesized into a weight with which each term in the dictionary represents each document in the corpus. In SMART, these weights are calculated as follows:

$$c_{ij} = \log(1+tf_{ij})/df_j \qquad \text{Eq. 1}$$

where:

i is the document index
j is the term index
tf_{ij} is the count of term j in document i
df_j is the number of documents containing term j

Similarity Measurement: In mapping a query to documents within the corpus, a simple matrix multiplication is used. The document corpus matrix (**C**, *m* rows of *n*-dimensional document vectors) is multiplied by the query vector (**q**, *n* term weights in column form), resulting in a column vector of query-document similarities (**s**, *m* document similarities):

$$\begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{m1} & \cdots & c_{mn} \end{bmatrix} \begin{bmatrix} q_1 \\ \vdots \\ q_n \end{bmatrix} = \begin{bmatrix} s_1 \\ \vdots \\ s_m \end{bmatrix} \qquad \text{Eq. 2}$$

Technically, this form of IR is both simple and elegant. However, much additional work is directed toward circumventing the assumption that the *meaning* of a document can be determined solely from the profile of words that occur in it - its *expression*. It is in this area we direct the rest of this paper.

First, we must discuss metrics for assessing the quality of an information retrieval system's performance. Two metrics are usually applied to an IR system: recall, the proportion of relevant documents retrieved by the system; and precision, the proportion of retrieved documents that are relevant. Perfect performance would be to retrieve all of the documents that are relevant to a query with no irrelevant documents mixed in. This being unattainable, we must recognize that the two measures are coupled - an increase in precision is usually made at the expense of recall and vice versa. In large part, it is up to the user's preferences of recall vs. precision to determine which overall strategies are most useful.

Precision and recall both depend on knowing, for each query, which documents from the corpus are relevant. Thus, it is necessary to use human judges to determine the set of documents that a query *should* return. This set can then be used to calculate a retrieval system's precision and recall. Here, relevance is a Boolean decision and no specific ordering is given for the return set; however, as Eq. 2 shows, SMART gives a retrieval set that can be ordered by its similarity measure. Thus, precision and recall are calculated for various thresholds of similarity. In our case, we will show precision at distinct recall levels.

We now return to the issue of obtaining *meaning* from *expression.* Thesauri have been used to improve recall by expanding the query to include synonyms of terms found within it. The idea is that a query represents only one expression of its true meaning and that by appending more expression can improve query performance. This has shown to be the case both in design information retrieval [Wood et al. 1998] and in information retrieval in general [Salton 1988]. The best thesaurus to use is one that will not add terms to the query which might hurt performance. For this reason, it is important to limit the scope of a thesaurus to the domain of interest.

The method of thesaurus creation and its implication on query performance is the main focus of this paper. Human-developed thesauri (described below) will be compared to thesauri generated mathematically from the corpus representation used for information retrieval. Sets of terms which coocur in the corpus are extracted from the set of documents using a method called Latent Semantic Indexing (LSI) [Schutze and Silverstein]. The basic idea is that by performing a singular value decomposition (SVD) of the document-term matrix , term 'modes' can be extracted. Using SVD, the document term matrix (rows of individual document vectors) is decomposed into a product of three matrices :

$$\mathbf{C} = \mathbf{USV}^{\mathrm{T}} \qquad \text{Eq. 3}$$

where:

C is the mxn corpus matrix (rows are document vectors)
S is an mxn diagonal matrix (0 padded so dimensions agree)[1]
V is an nxn orthonormal matrix of term modes in the corpus
U is an mxm orthonormal matrix that reconstructs **C** from the term modes

Essentially, the term modes (**V**) represent the partitioning of the corpus into a set of 'themes', ordered by the strength of the mode in the corpus (given by its associated diagonal value from **S**). Clearly these themes are of interest toward identifying which corpora might be of interest in a given design situation; a corpus can be modeled by its main themes when doing a high-level, information source identification search. In the next section we hope to determine if corpus themes can be useful as a thesaurus. Because these themes represent the same level of granularity for both searching and retrieving, themes may not be specific enough to use as a thesaurus. For this reason, we split documents into smaller 'chunks' in which term cooccurence is limited to the paragraph or sentence level. The same theme extraction techniques are then used over these smaller, more closely related chunks.

V. Results and Discussion

Four retrieval experiments were run over three PENS design notebooks. Each notebook represents the primary documentation for a seven month, industry-sponsored project. For each of these experiments, the same set of queries was used: nine queries for each notebook - three sets of three queries each for <*alternative*>, <*construction*>, and <*performance*> Dedal descriptors. The set of relevant documents for each of the 27 queries was determined through an exhaustive search of each notebook.

The first experiment is designed to identify the best source of information from which to hand-construct a thesaurus and the generality of information which should be included in it. The second experiment compares means of automatically generating a thesaurus based on the best information

[1] **S** and **V** (without the 0 padding) are simply the eigenvalue decomposition of an affinity matrix ($\mathbf{CC}^{\mathrm{T}}$) which is calculated directly from term-term coocurrence within the corpus.

Descriptor + Subject

<Alternative> *Part of an object model*
<Assumption> *Ex:*
<Comparison>
<Construction>
<Location>
<Operation>
<Performance>
<Rationale>
<Relation>
<Requirement>

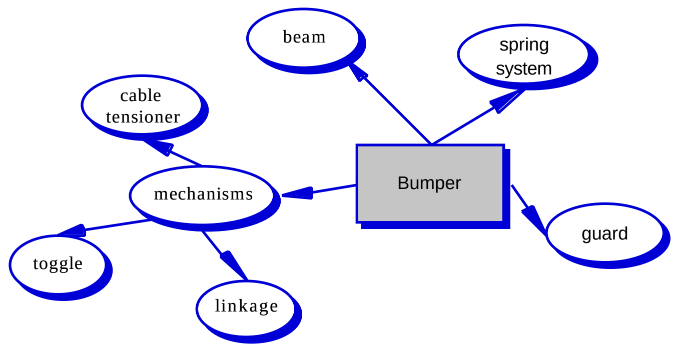

Figure 1: The Dedal Framework

source found from the first experiment. Finally, we compare the performance differences between hand and machine generated thesauri.

Hand-Built Thesauri

Dedal's descriptor-subject pair, shown in Figure 1, presents two distinct points of introduction for thesaurus terms which might improve information retrieval performance. Because Dedal descriptors are generic, we will use a generic thesaurus based on dictionary/thesaurus searches for each descriptor base. For example, <*construction*> might be augmented by a set of terms like (assemble, attach, build, machine, manufacture, tool, etc.). This thesaurus is generic to the mechanical design process, so is the same for all <*construction*> queries. More challenging is the task of determining valid Dedal subjects. In the original research, a final model of the design was constructed and the documentation linked by hand into this network of design subjects. Because of this retrospective deterministic indexing Dedal could retrieve information about a subject even if the

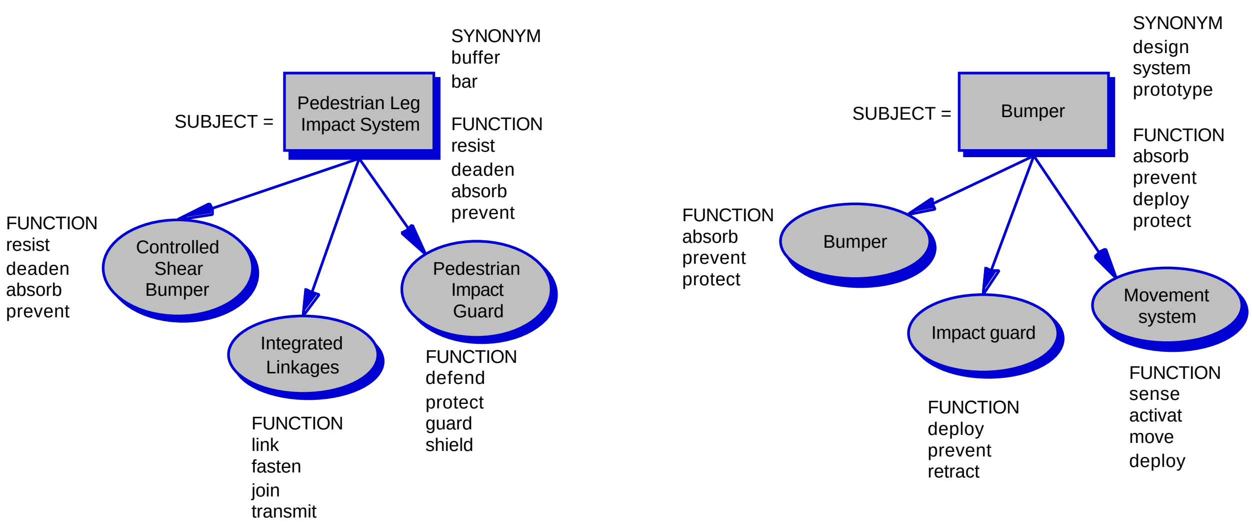

Figure 2: Difference in part naming in CAD and DP models

name used in the model was never mentioned explicitly in the design documents. In order to use information retrieval methods, we needed to create a thesaurus for each design that could mimic the original Dedal subject index while covering the actual terminology used within the design notebook.

Two scenarios surface when charged with this task: creating the thesaurus based on final, formal design documentation or building it in-process as the design evolves and names for subsystems and subfunctions change. Figure 2 shows partial models for one of the design notebooks created using both sources, including both structural and functional aspects of the design. The hierarchical nature of the models presents another avenue for study beyond just evaluating the knowledge source; we can also vary the breadth of term introduction from just the described subject to its functional and structural relatives.

Figure 3 shows results averaged over the three design notebooks for: Dedal <descriptor> <subject> query augmented with the descriptor thesaurus terms (*DedaldT*); *DedaldT* augmented with the design process thesaurus terms for subject and function (*DP DedaldTsTfT*); *DP DedaldTsTfT* augmented with the parent and children from the design process model (*DedaldTsPCTfPCT*); *DedaldT* augmented with the final design document-derived thesaurus for subject and function (*CAD DedaldTsTfT*); and *CAD DedaldTsTfT* augmented with its parent and children terms (*CAD DedaldTsPCTfPCT*). The main feature of the graph is the poor performance of the traces for the final document-derived thesaurus (labeled CAD *) compared to those for the in-process thesaurus (labeled DP *). Using a thesaurus based on the final design documents only slightly improves performance over *DedaldT*, the baseline for comparison. Overall, the difference between the in-process and final design thesaurus performance is near 40%. The difference in

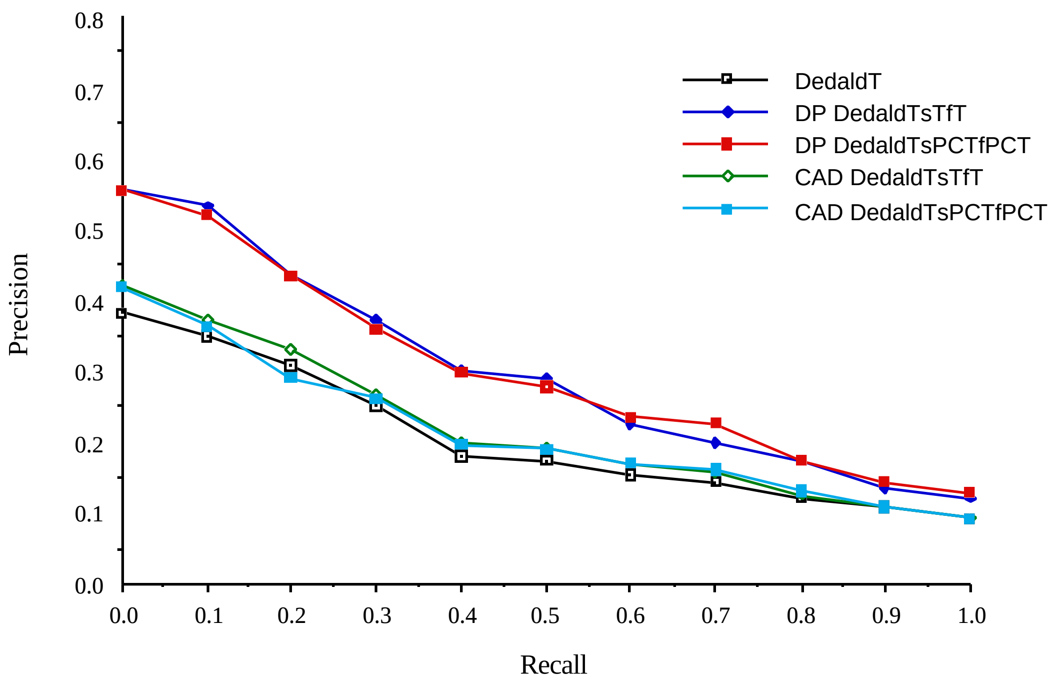

Figure 3: In-design-process thesauri vs. post-process thesauri

terminology between in-process and final documentation is significant; a thesaurus used to access actual in-process documentation must be derived from it. This result is not surprising but does cost a great deal of effort. It is relatively easy to construct a model from the final design, but it requires a great deal more resources to construct one that represents the evolution of ideas throughout the design process.

Machine-Generated Thesauri

Having determined a design subject thesaurus should come from the actual in-process design documents, we now look for methods which might reduce the extra work required to create such a thesaurus. Using the SVD of each design notebook, we extract term coocurrence modes which represent prevalent themes. This is done for the original corpus as well as for a second corpus created from the first set of documents by breaking each into overlapping fifty word ‘chunks’. The motivation for doing this is to try to create extract themes from within document sections which we presume to be contextually ‘close’[2]. The process that is used is relatively automatic: the SVD of each corpus yields a set of orthonormal ‘modes’ whose peaks are extracted and used to populate a thesaurus database. The original Dedal <descriptor> <subject> query is applied to a SMART index of the database to identify themes in which both terms appear with the same numerical sign[3]. These

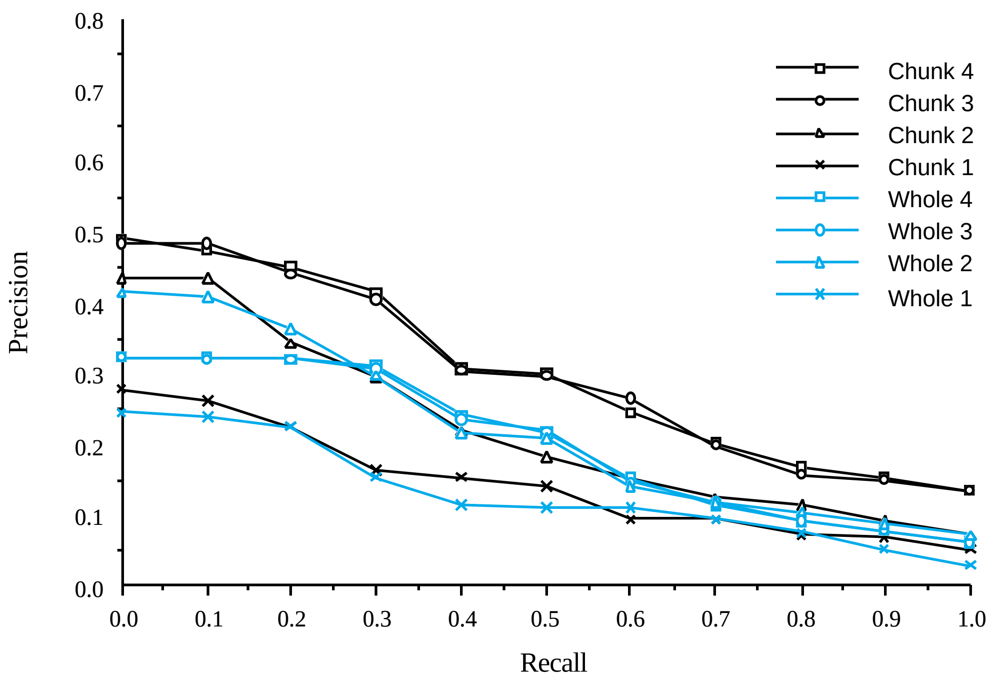

Figure 4: Chunk indexing vs. Whole document indexing for generating thesauri

[2] Here, physical proximity is assumed to be the closest easily accessible measure of contextual proximity. In the presence of media other than text, chunking on a spatial basis might be possible.

[3] Although both the corpus matrix and the affinity matrix derived from it have only positive entries, matrix modes contain both positive and negative terms. For this reason, modes contain information both about which terms should appear together but also that they should not appear along with terms of opposite sign.

themes are ordered with respect to SMART similarity scores and divided into quartiles for selective augmentation of the original Dedal query.

Results from this experiment are shown in Figure 4. A root of *Chunk* denotes that the thesaurus generated from the chunked corpus was used, *Whole* denotes that the whole document corpus was used. Appended to each of these is the lowest quartile of thesaurus terms added into the query. There are two definitive trends shown here: 1) the chunked thesaurus performs much better (~70%) than the whole document thesaurus demonstrating that contextual proximity improves theme extraction; and 2) that nearly all of the thesaurus terms retrieved by SMART are useful (the first two quartiles produce inferior performance to the last two which produce nearly identical results). Now we compare results between the best performers from the first two experiments.
Hand vs. Machine Generated Thesauri

At this point we have established that the actual in-process design documentation is the best starting point for a thesaurus and that if a machine-generated thesaurus is to be mined from this corpus, the corpus should be broken into smaller chunks. Figure 5 directly compares the two thesaurus methods; the conclusion here is that performance by the hand created thesauri is superior to the machine generated ones. This is not surprising in the sense that leveraging domain specific knowledge would intuitively improve recall and precision. However, the clear trade-off between the two approaches to building thesauri is the effort and resources required to create them. Hand constructed thesauri may provide better performance, but at the same time need knowledge and time to create. In addition, the machine-generated thesauri perform better than the baseline *DedaldT* method in all but the lowest recall, highest precision setting. This appears to be due to a chunking factor that is too large; further study of chunking method is warranted toward closing the gap between automatic and hand-generated thesauri.

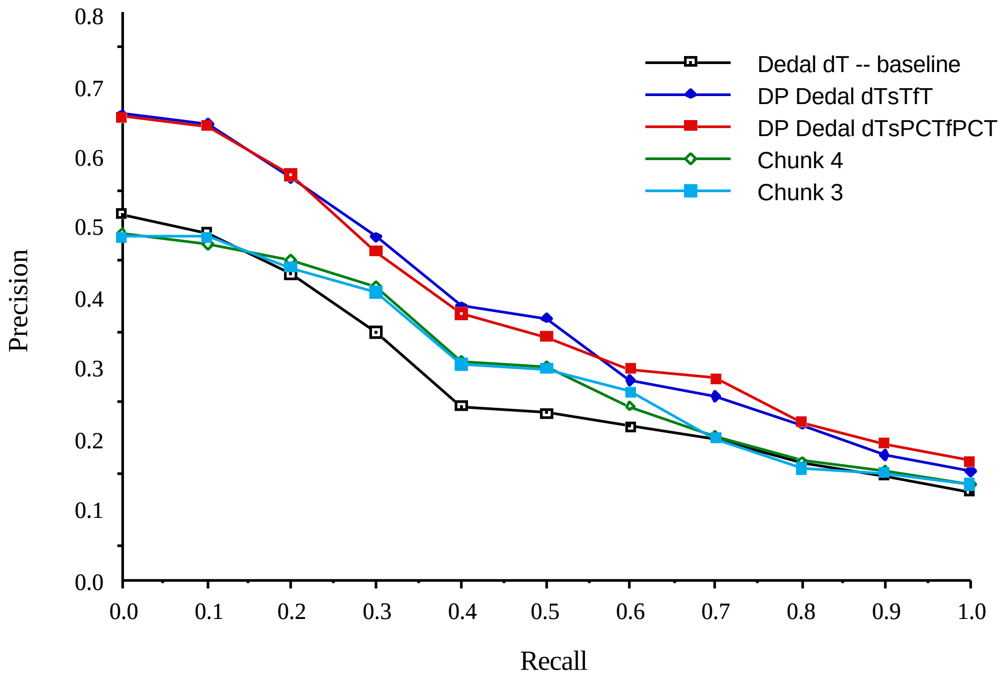

Figure 5: Hand created thesauri vs. Machine generated

VI. Conclusions and Future Work

This paper explores the use of a thesaurus to augment search for design information in design notebooks. We examine several methods of generating thesauri: hand building and machine generation. For both types of thesauri, we experimented to see what types of source material would be most effective for developing thesauri.

In the case of hand built thesauri, we compared the results of using in-process information from design logbooks to create thesauri with after-process information found in final reports. It was believed that creating thesauri from after-process material is more straightforward than from in-process material, and would make a hand generation of thesauri a more palatable option for information retrieval. However, it was demonstrated that using in-process information produced better recall and precision than after-process information. This result was not surprising, given that the material being searched is also in-process information.

An alternative to hand building of thesauri is machine generation using term-document vectors. This paper shows a comparison between using chunked, or sections of documents, and whole documents as sources for thesauri. It was hypothesized that using chunked documents would produce better results because of the way chunking limits the proximity of terms. Our experiments bore this guess out, showing chunking to be the better method.

Finally, we directly compared the best of each type of thesauri, hand and machine generated. Machine thesauri require little human overhead to generate, making it an attractive alternative to hand built thesauri. Testing demonstrated that manually created thesauri from in-process information produce clearly better precision and recall than machine generated thesauri from chunked material.

These results illustrate a classic intelligence trade-off. That is, using more domain knowledge produces better results than using non-domain knowledge. In the work presented in this paper, domain knowledge is equated with increased human effort to generate thesauri.

In view of these results, future work will concentrate on filling in the gap between manually built and machine generated thesauri. Because of the superior performance of chunked source material over whole documents for machine generated thesauri, we will explore the effectiveness of further decreasing chunk size to boost performance. We will also take other approaches to machine generation of thesauri, using different schemes for creating the modal matrix, such as covariance matrices. For hand generated thesauri, we will test the utility of using machine techniques for creating the Dedal descriptor thesaurus. It is felt that the descriptor thesaurus is a generic and stable set of terms, ideal for machine approaches. Finally, in an effort to reduce the human effort required to produce manual thesauri, we will continue to look for more structured methods for creating these thesauri.

VII. Bibliography

Baudin, C., Underwood, J. G., Baya, V., Mabogunje, A., 1991, “Dedal: Using Domain Concepts to Index Engineering Design Information”, *Workshop Notes from the Ninth National Conference on Artificial Intelligence*, Intelligent Multimedia Interfaces, Anaheim, California.

Baudin, C., Underwood, J. G., Baya, V., 1993, “Using Device Models to Facilitate the Retrieval of Multimedia Design Information”, *Proceedings of International Joint Conference on Artificial Intelligence* .

Conklin, J., and Begeman, M., 1988, “gIBIS: A Hypertext Tool for Exploratory Policy Discussion”, *Proceedings of the Conference on Computer-supported Cooperative Work*, ACM, New York, pp. 140-152.

Dong, A., Moore, F., and Woods, C., 1995, “Managing Design Knowledge in Enterprise-Wide CAD”, *IFIP WG5.2 Second Workshop on Formal Design Methods for CAD*, IFIP.

Dong, A. and Agogino, A.M., 1997, “Text Analysis for Constructing Design Representations”, *Artificial Intelligence in Engineering*, 11(2):65-75.

Fruchter, R., Clayton, M., Krawinkler, H., and Kunz, J., 1996a, “Interdisciplinary Communication Medium for Collaborative Conceptual Building Design”, *Advances in Engineering Software*, 25(2-3):89-101.

Fruchter, R., Reiner, K. A., Toye, G., and Leifer, L. J., 1996b “Collaborative Mechatronic System Design”, Concurrent Engineering: Research and Applications. Vol. 4, no. 4, Dec. 1996, p. 401-412.

Hong, J., Toye, G., Leifer, L., , 1994, “Using the WWW for a Team-Based Engineering Design Class”, *Electronic Proceedings of the Second WWW Conference*, Chicago, Illinois, October 1994.

Kunz, W. and H. Rittel, 1970, “Issues as Elements of Information Systems,” Studiengruppe Fuer Systemforschung, Heidelberg, Working Paper #131, 1970. (Also described in D. Grant, “How to Use the IBIS as a Procedure for Deliberation and Argument in Environmental Design Planning,” *Design Methods and Theories*, vol. 11, no. 4, pp. 185-220, 1977.)

Liang, T., Cannon, D. M., and Leifer, L. J., “Augmenting the Effectiveness of a Design Information Capture and Reuse System Based on Direct Observations of Usage”, Submitted to the 1998 ASME DETC Design Theory and Methodology Conference.

MUC http://www.muc.saic.com/

McCall, R., Ostwald, J., and Shipman, F., 1991, “Supporting Designers’ Access to Information through Virtually Structured Hypertext”, *IntCAD ‘91 Preprints*, IFIP WG 5.2 Working Conference on Intelligent CAD.

Qreshi, S.M., Shah, J.J., Urban, S., Harter, E., Bluhm, T., 1997, “Integration Model to Support Archival of Design History in Databases”, *Proceedings of DETC ‘97:1997 ASME Design Engineering Technical Conferences*, DETC97/DTM-3876

Salton, G., 1988, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer,* Addison-Wesley, Reading, MA.

Schutze, H. and Silverstein, C., 1997, “Projections for Efficient Document Clustering”, *SIGIR 1997*, pp 74-81 (ftp://parcftp.xerox.com/pub/qca/papers)

TREC, (http://trec.nist.gov/)

Ullman, D.G., 1994, “Issues Critical to the Development of Design History, Design Rationale, and Design Intent Systems”, *Proceedings of Design Theory and Methodology - DTM ‘94*, DE-Vol. 68, pp. 249-258, ASME.

Wood, W. H. and Agogino, A. M., 1996, “A Case-based Conceptual Design Information Server for Concurrent Engineering”, *Journal of Computer Aided Design*, 28(5), pp. 361-369.

Wood, W. H., Yang, M. C., and Cutkosky, M. R., 1998, “Design Information Retrieval: Improving Access to the Informal Side of Design”, Submitted to the 1998 ASME DETC Design Theory and Methodology Conference.

Yang, M. C. and Cutkosky, M. R., 1997, “Automated Indexing of Design Concepts for Information Management”, *Proceedings of the Eleventh Annual Conference on Engineering Design*, Tampere, Finland, August 1997.