PREVENTING FAILURES BY MINING MAINTENANCE LOGS WITH CASE-BASED REASONING

Mark Devaney, Ashwin Ram
Enkia Corporation
[markd@enkia.com, ashwin@enkia.com]
Hai Qiu, Jay Lee
Center for Intelligent Maintenance Systems
University of Wisconsin-Milwaukee
[haiqiu@uwm.edu, jaylee@uwm.edu]

Abstract: *The project integrates work in natural language processing, machine learning, and the semantic web, bringing together these diverse disciplines in a novel way to address a real problem. The objective is to extract and categorize machine components and subsystems and their associated failures using a novel approach that combines text analysis, unsupervised text clustering, and domain models. Through industrial partnerships, this project will demonstrate effectiveness of the proposed approach with actual industry data*.

Key Words: Data Mining; Semantic Web; Maintenance log analysis; Natural Language Processing; Machine Learning, Case-Based Reasoning

1. Introduction:

This research addresses the problem of discovering knowledge required for critical business performance improvements by mining the equipment maintenance logs collected by the owners of complex machinery. These logs typically consist of terse free-text input manually entered by maintenance personnel. Implicit in these logs is information about equipment components and associated failures and repairs. If this information can be made explicit, it provides valuable knowledge resulting in improvements in operating and maintenance processes.

The approach described here is an inter-disciplinary combination of sophisticated text analytics, a novel bootstrapping algorithm for unsupervised text clustering, and an extensible semantic representational framework for the construction of domain models of equipment used in the manufacturing industry. This representational framework builds on the standards of OWL (Web Ontology Language) and the Semantic Web concepts of RDF (Resource Description Framework) in order to provide a powerful extensible representational component to the work.

Our approach extracts from the free text logs a canonical set of physical machine components and subsystems and their associated failures. The extraction of such information can provide the data necessary to characterize operating cycles, maintenance schedules, periodic breakdowns, and most importantly, to identify and address abnormal failure rates before critical problems arise. This ability has long been a goal of the owners of this expensive equipment, and some manufacturing companies have been retaining maintenance log records for over ten years with this objective in mind. However, due to the nature of the data, extracting useful data from these maintenance logs has proven to be a difficult endeavor. A data-mining tool to extract this valuable information can provide owners and maintainers of complex equipment with the knowledge necessary to improve their operating efficiencies, reduce critical downtimes, and determine optimal maintenance schedules.

2. Background:

Maintenance logs describe every problem, repair, adjustment, or alteration made to complex, expensive machinery over their entire lifespan. Each log record corresponds to some physical action taken on some physical component or subsystem of a particular machine. This data is collected for every machine in use and retain the historic logs for many years with the intent of extracting the operating information that is critical to improving efficiency and reducing downtimes and unexpected failures. However, because the log entries are made by humans and entered in free-text natural language without descriptive or other standards, the variation in the input is extremely large, exhibiting all of the expressive freedoms possible in natural language with the additional freedoms of not having to adhere to the usual constraints of spelling, grammar, or vocabulary.

The information that must be extracted from this multitude of free text representations is the canonical set of actions and the associated components that were made to a particular machine or class of machines. From this information it is possible to identify repair trends, spot deviations from expected failure rates, construct lifecycle models, and a number of other activities that increase efficiency and provide opportunities for process improvement. There are three specific objectives:

- To identify categories of components at a suitable "natural" level of description (e.g., "clamps")
- To identify categories of problems and/or repair actions associated with these components (e.g., "hydraulic oil leak", "adjust pressure")
- To learn distributions of the problems and/or repair actions (e.g., "hydraulic oil leak" accounts for 20% of "clamp" problems)

We formulate this task as a *clustering* problem (Hanson, 1990) in which the system must learn the natural categories of the data as well as learn a model for distributing new input into those categories. More specifically, it is a *text clustering* problem, because the maintenance data is represented solely as free-text. However, there are a number of characteristics of the maintenance log data that make the application of traditional approaches unviable and there are real-world constraints that require the innovative approach described here.

Due to the realities of the operating environment and personnel, maintenance logs are often recorded in fairly haphazard and non-standard ways. These logs may be written by hand and then transcribed or entered directly from the operating environment via limited input devices such as PDAs or handheld computers. The subsequent input data, while nominally natural language, exhibit many characteristics that make standard approaches to natural language processing and text classification difficult:

Vocabulary used in descriptions is inconsistent. There is typically no standard set of terms used for the names of mechanical parts or the activities performed on them. For example, one record might contain “change the oil” whereas another might contain “replace the fluid”, both referring to the same action. Additionally, unlike this example, many of the descriptions in the real-world data are not amenable to traditional dictionary or thesaurus-based approaches as the terms used may be highly specific to the particular machine and/or laced with jargon.

Vocabulary may not correspond directly to systems or components of interest. Repair personnel may refer to a component or part of a system without explicitly mentioning the large system that the company is interested in monitoring. For example, “broken hose” may actually be “broken coolant hose” which needs to be categorized as a coolant system fault and not a hose fault *per se*.

Input is not well-formed. Because of character length limitations and probable treatment of the log entries as secondary task, the text descriptions are limited to short phrases or sentence fragments. Additionally, attention is not typically paid to spelling and other language rules, and the data therefore exhibits very high degrees of grammatical and spelling errors. For example, “add oil to leakey[sic] fixture” contains a typo whereas “add oil to fixture” neglects to mention that the fixture is leaky.

Jargon and extremely terse abbreviations are common. Due to the limited input space and time pressures of maintenance engineers, log entries are usually laced with creative abbreviations and contain large amounts of jargon. For example, “HD-7” might be a kind of “hydraulic oil”. The jargon may be range in use from generally-known to very local terms for particular machines or tools.

Large amounts of data are not available. We work with some of the largest manufacturing companies in the United States. These companies, some of whom have been collecting data for almost ten years, have maintenance log databases of at most 5-10,000 records for a single class of machines. Thus, a new type of machine put into service will have on the order of 50-100 maintenance actions performed per year. While this is too large for manual analysis, it is too sparse for purely data-intensive approaches.

The following are examples of actual maintenance log records that exhibit some of the properties described above:

128272	"HIGH PRESSURE COOLANT COMES ON AT WRONG
83618	"HYD LEAK ON CLAMPS, POSSIBLE BLOWN GUAGE"

414181	"Hyd. oil leak, B-Axis table."
89666	"HYDRAULIC PUMP MAKING NOISE, LOCATED IN
181353	adjust clamping pressure
241772	adjust clamps
409507	ADD COOLANT HOSE TO 4264
674894	ADJUST CLAMPS
476269	ADD OIL TO LEAKEY FIXTURE
594549	add HD-7
398880	add hydraulic oil
124594	adjust clamps from hitting heads
644999	ADJUST CLAMPS OVER DATUM POINTS
199925	BROKEN HOSE

The very limited amount of data available and its sparseness of content make purely data-intensive statistical text clustering approaches impractical. Conversely, knowledge-intensive approaches, such as supervised learning models, require a degree of knowledge engineering (e.g., to construct a labeled training set) that is impractical for a real-world application, as companies typically do not have the personnel or other resources to devote to this task.

3. Solution:

Our solution is based on four key ideas:

- Develop a domain ontology through a knowledge engineering effort
- Represent the domain in OWL/RDF
- Use text analytics to sanitize the data (spelling, stemming, stop words) and reduce problem dimensionality
- Apply a conceptual clustering approach to learn categories from ontology-based data
- Employ case-based reasoning (Kolodner, 1993) to identify patterns of maintenance activities or component malfunctions that lead to systemic failure.

This approach has its roots in Clerkin, Cunningham & Hayes (2001), who propose a method based on the concept formation system COBWEB (Fisher, 1987) to learn ontologies represented in RDF schema (Brickley & Guha, 2000). Their goal was to cluster documents in order to generate class hierarchies for the semantic web. Although our data set shares some of the characteristics of theirs, our problem is somewhat different. Instead of learning ontologies, our goal is to use existing ontologies to learn natural categories of objects that occur in the data set.

In order to accomplish this, we need to develop a method of evaluating similarity of ontological objects that might be described at different levels in the ontological hierarchy. Maedche & Zacharias (2002) discuss methods for similarity assessment for clustering ontology-based meta-data, again for the semantic web. Our approach also requires clustering of ontology-based data, but our data is somewhat different from the meta-data in their approach.

Building on these approaches, our unsupervised conceptual clustering approach uses the OWL ontologies to classify and assess similarities between objects, problems, and repair actions in the data set. The output is a set of clusters representing families of <object, repair> pairs that occur in the data. Our approach uses automated techniques to augment the manual creation of ontologies. Specifically, we extend Blum & Mitchell's (1988) co-training approach, which allows a system to learn over a large set of unlabeled data using a small subset that is manually labeled. Finally, the learned categories are used to construct a *case library* database of maintenance patterns leading to systemic failure. These patterns are used by a Case-based Reasoning engine (CBR) to predict future failures based on observed patterns of maintenance activity as well as allow more efficient troubleshooting and diagnosis in the event of a failure.

The component overview of this architecture is shown in Figure 1 below.

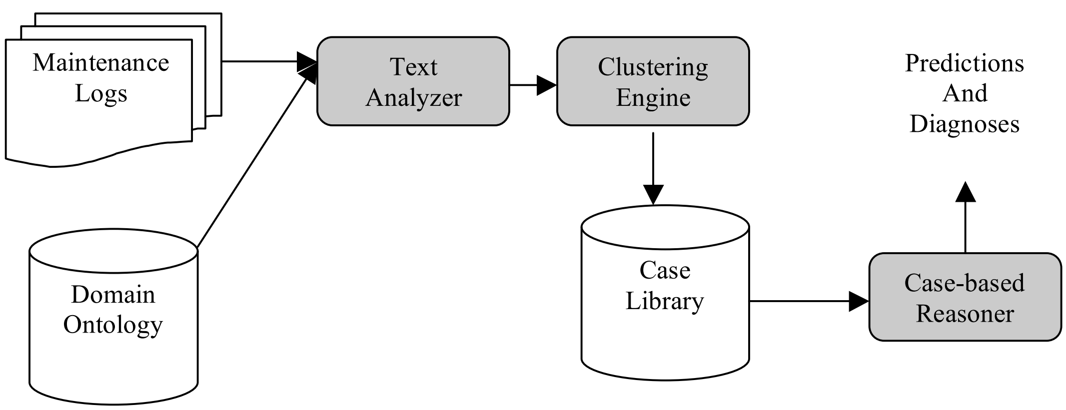

Figure 1: System Architecture

3.1. Generic Representations:

There is a great deal of effort underway to construct ontologies and other meta-data representations in order to help make use of the vast number of information repositories available. The "Semantic Web" (Berners-Lee, 2001) is the notion that in addition to low-level information, there will be machine-readable databases of meta-data and other background knowledge that will enable computers to more effectively fuse different sources of data and make much more powerful inferences from that data. Similarly, in the manufacturing industry (among others), standards organizations (e.g., ISO) have been creating taxonomies, ontologies, and vocabularies to describe the equipment and processes used in those businesses. For example, The International Electrotechnical Commission (IEC) standard 61499-1 describes architectures for industrial-process measurement and control. However, these standards are more typically focused on processes rather than physical objects and are extremely complicated and detailed.

Our approach is to fuse the ideas of the Semantic Web and industrial standards organizations and develop ontologies for the domain of machine tool maintenance. These ontologies are represented using OWL (Web Ontology Language), an XML-based extension of RDF (Resource Description Framework) that is an official W3C standard (World Wide Web Consortium). Our ontologies are developed to be generally-applicable across a number of application domains and types of machinery. For example, all tools share high-level systems and subsystems such as hydraulics, electronics, pneumatics, etc. and each of these systems has certain characteristics that are independent of the particular machine in which they reside. All hydraulic systems have hoses, pumps, and fluid, all of which have specific functions and interconnections. And so on.

OWL is a hierarchical representational framework, so that in addition to a generic representation, more specific ontologies or models can be constructed within the OWL framework. Thus, for example, an ontology of a specific machine such as an "*HPC 500/630 TS Multi-Spindle CNC*" can be written to define components and systems specific to that machine while inheriting general properties from a higher representational class. By using an open and widely accepted representational framework we can leverage the contributions of others outside of this particular effort. For example, more general ontologies and models of physical systems that have been created elsewhere may be incorporated into our application to increase its capabilities.

3.2. Text Analytics:

Advanced text analytics algorithms including the use of standard English dictionaries and thesauri are used for parsing and analyzing records in the maintenance database in order to disambiguate jargon, misspellings, and abbreviations to the degree possible. Higher-level terms are identified through the use of part of speech tagging, *n*-gramming, and correlation techniques such as Latent Semantic Analysis. These techniques are used to reduce the representational dimensionality of the problem, i.e., to increase the number of apparent similarities between records. For example, in the sample data above, the two records "add HD-7" and "add hydraulic oil" actually refer to the same thing because HD-7 is the local name for the hydraulic fluid used. A thesaurus-based analysis allows the system to correctly equate those two records.

While text analytics represent an important component of this project, our primary focus is on the definition and construction of a powerful representational framework as described above and the development of a clustering algorithm to produce natural categories from the raw log data as described below.

3.3. Bootstrapping Clustering:

Even after applying text analytics to reduce discrepancies between records that describe the same physical action on the same underlying component (e.g., replacing a coolant hose), there is still a great deal of representational variety—that is, the number of unique log records is much larger than the number of unique <component, repair> pairs. The heart of the solution to this problem of discovery natural categories that can be used for business-process analysis is the use of *clustering* techniques to group the disparate log records into their functional groupings.

Domain models are used in a bootstrapping clustering algorithm by providing the *categories* represented by the data as a whole but no category labels for any individual records as needed by supervised classification methods. This approach provides the opportunity for improved performance over strictly unsupervised methods and presents new technical challenges. An additional component of our approach is the inclusion of advanced text analytics algorithms that also take advantage of the generic domain models in order to disambiguate text as much as possible, reducing the dimensionality of the input to the greatest degree.

Building on the work of Clerkin, Cunningham & Hayes (2001) and Maedche & Zacharias (2002) discussed earlier, our unsupervised conceptual clustering system uses the bootstrapping approach to discover natural groupings of components and repair actions in the text-sanitized repair logs. Domain ontologies from text analytics are used to assess similarities between objects, problems, and repair actions in the data set, and to classify these items into appropriate categories.

Taken together, these techniques enable an AI system to identify types of components, types of problems and repair actions, and their distribution in an automated manner.

3.4. Case-based Reasoning Engine

In case-based reasoning (CBR), solutions to new problems are obtained by finding similar past problems and analyzing the solutions to those problems for the answer to the current one. CBR is a mature technology and has been used successfully in a number of commercial and industrial applications. It has proven successful in a number of diagnostic and maintenance applications, for example, 3Com Corporation's on-line troubleshooting system (http://knowledgebase.3com.com/) and Iomega's "i-man" online support system (http://iomega.com). The case-based reasoning process consists of a number of steps as shown in Figure 2 below.

Retrieval : In the first stage of CBR, a new maintenance log (or subset) is provided to the system. The indexer module retrieves a set of related maintenance histories ("cases") from the case library. For a case to be retrieved, a mathematical combination of its similarity to the new maintenance history and its historical behavior must meet a performance threshold relative to the other cases in the library – e.g., each case must "compete" against all others in order to be chosen for retrieval. Similarity is computed by the co-sine between the two feature vectors:

$$s^{(\mathrm{C})}(\mathbf{x}_a, \mathbf{x}_b) = \frac{\mathbf{x}_a^{\dagger}\mathbf{x}_b}{\|\mathbf{x}_a\|_2 \cdot \|\mathbf{x}_b\|_2}$$

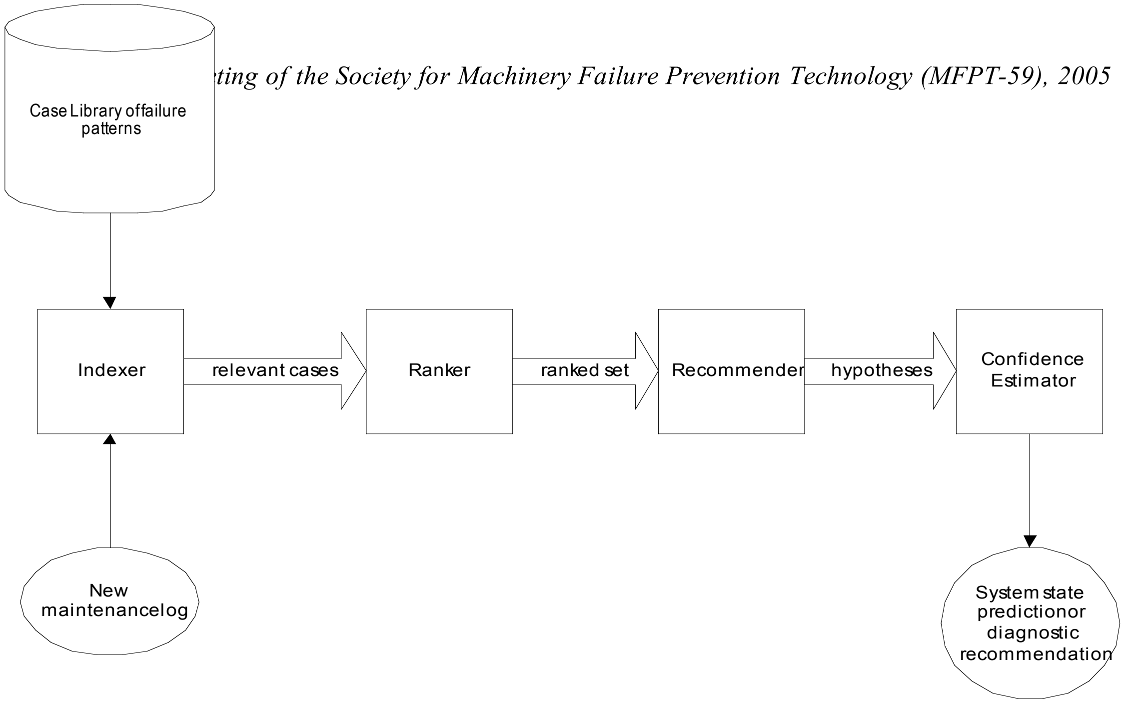

Figure 1: Case-Based Reasoning Process

Ranking: The set of related prior cases from the indexer is then ordered or prioritized by the ranker module. This ordering is based on similarity vs. quality metric computed during the retrieval phase and results in a creating a distribution of the items in the retrieval set against that smaller population.

Recommendation: In the recommendation stage of CBR, the ranked set of most similar past cases are combined to produce a set of hypothesized corrective actions for the new discrepancy. This is done by computing a weighted normalized histogram of cluster patterns based on all of the cases in the ranked set.

Confidence estimation: In the final stage of the CBR process, each hypothesis generated by the recommendation process is given a confidence value (0..1), which helps a human operator to decide whether to pursue the recommendation of the automated system. Confidence values are derived from internal properties of the system such as the relative cohesiveness of the ranked set of more similar past log histories and the degree to which one or more of the predictions "stands out" from the others.

4. Evaluation Plan

A comprehensive evaluation of the approach described above is in progress and we are anticipating significant results based on our related experience in applying text analytics techniques to complex maintenance problems. For example, the chart in Figure 3 below shows the application of text analytics and user feedback to identifying appropriate troubleshooting recommendations based on free-text problem descriptions (Ram & Devaney, 2005).

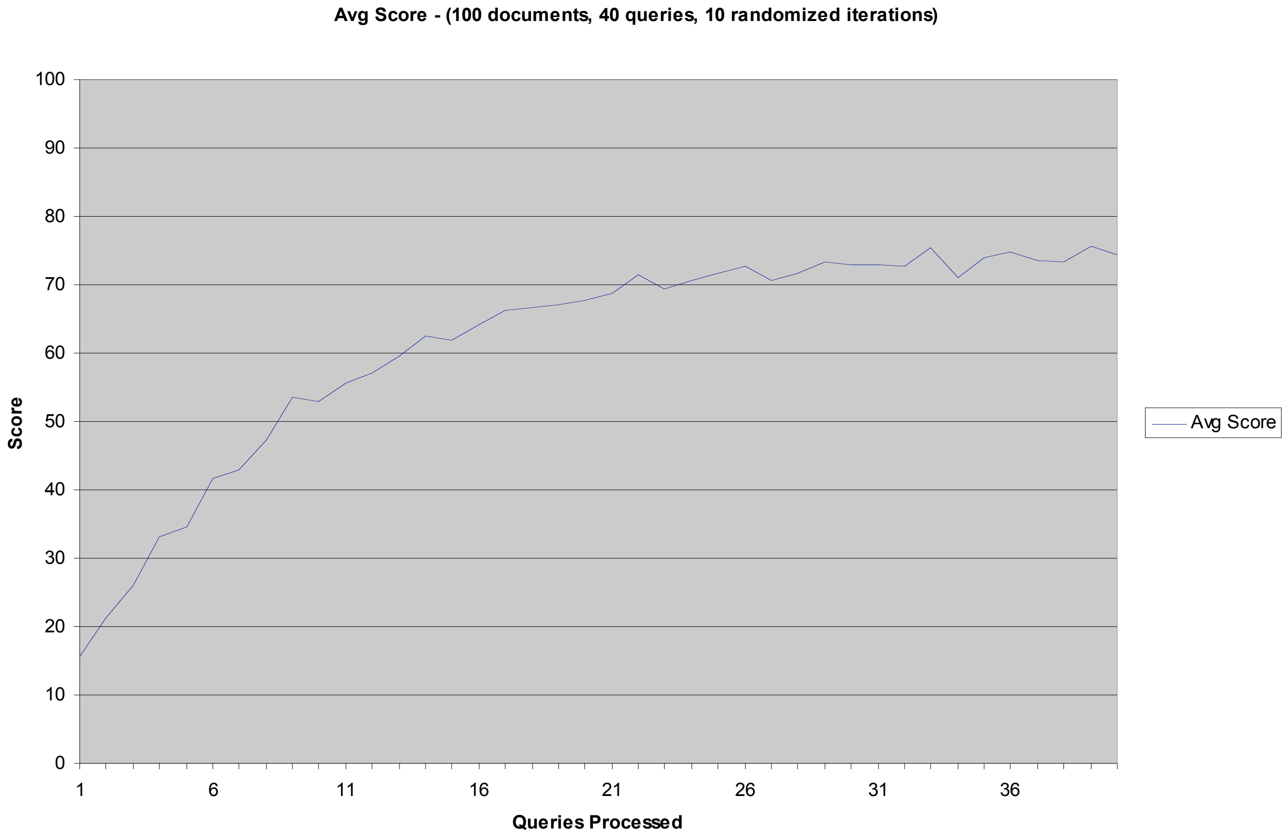

Figure 2: Troubleshooting problem identification with user feedback

Our industry partners have committed to providing us with real-world data with which to evaluate the effectiveness of the proposed approach. Our evaluation plan will focus on quantitatively measuring the quality of the classification of log records into natural categories through the use of domain experts as well as our industry partner. These evaluations will be primarily useful for benchmarking against the results obtained using naïve clustering algorithms as described in section 2.1 as well as measuring the improvements that are made as the domain ontology and model evolves over time. Additionally, evaluations will be conducted to measure the business impact of the approach by running the system and generating a taxonomy and frequency of repair actions. These results will be provided to experienced maintenance engineers and managers who will assess the utility of the results and provide rough characterizations of the usefulness of the systems performance. Ideally, we also hope to obtain quantitative estimates of business metrics that would be improved through use of the system, such as cost and time savings, failures avoided, etc.

5. Conclusions

Maintenance logs contain a potential wealth of information that can be used to improve the maintenance of complicated machinery, reduce downtimes, and prevent failures. Advanced text analytics techniques show great promise in extracting this useful information from the complicated, diverse, and inconsistent entries typically found in these maintenance logs. This paper has described our approach in employing these techniques along with advanced knowledge representations in order to construct a powerful Data Mining software architecture. Our next step is to conduct extensive evaluations of the system using real-world data and collecting both quantitative and qualitative metrics.

6. References

Berners-Lee, T. (2001). Semantic Web. http://www.w3.org/2001/sw/

Blum, A., & Mitchell, T. (1988). Combining Labeled and Unlabeled Data with Co-Training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, 92-100.

Brickley, D., & Guha, R.V. (2000)., 2000] Dan Brickley und R.V. Guha. *Resource Description Framework (RDF) - Schema Specification 1.0*. http://www.w3.org/TR/2000/CR-rdf-schema-20000327.

Clerkin, P., Cunningham, P., & Hayes, C. (2001). Ontology Discovery For The Semantic Web Using Hierarchical Clustering. In *Proceedings of the Workshop in Semantic Web Mining, Twelfth European Conference on Machine Learning*, Freiberg, Germany.

Fisher, D.H. (1987). Knowledge Acquisition with Incremental Conceptual Clustering. *Machine Learning*, 2:139-172.

Hanson, S.J. (1990). Conceptual clustering and categorization: Bridging the gap between induction and causal models. In R. S. Michalski and Y. Kodratoff, editors, *Machine Learning: An Artificial Ingelligence Appproach (Volume III),* chapter 9, pages 235-268. Morgan Kauffmann, San Mateo, CA, 1990.

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

Maedche, A., & Zacharias, V. (2002). Clustering Ontology-Based Metadata in the Semantic Web. *Principles of Data Mining and Knowledge Discovery*, 348-360.

Ram, A., & Devaney, M. (2005 in progress). Interactive CBR for Precise Information Retrieval. In *Case-Based Reasoning in Knowledge Discovery and Data Mining,* S. Pal & D. Aha (Eds.). To be published.