CBR Textuality*

David C. Wilson
Computer Science Department
Lindley Hall, Indiana University
150 S. Woodlawn Ave
Bloomington, IN 47405
davwils@cs.indiana.edu

Shannon Bradshaw
Intelligent Information Laboratory
Department of Computer Science
Northwestern University
1890 Maple Ave.
Evanston, IL 60201
bradshaw@infolab.nwu.edu

Abstract

Current textual CBR research focuses on generating knowledge-rich representations for working with document-cases. However, there are many "weakly-textual" contexts in which textual information plays an important, but ancillary role in case composition and reasoning. For retrieving weakly-textual "semi-structured" cases that contain one or more textual features, it is desirable to measure the similarity of the textual components in a simple manner that is easily integrated with standard, non-textual similarity metrics, such as nearest-neighbor methods. This paper describes an investigation of the issues involved in weakly-textual CBR, which has been motivated by work on the DRAMA system for aerospace design support. It describes a characterization of textual CBR along a continuum of textual importance in case composition and reasoning, that places weakly-textual CBR in relation to current practice and motivates a focus on weakly-textual methods. The paper goes on to discuss methods, which are currently being incorporated into DRAMA, that integrate simple information retrieval techniques into standard CBR similarity metrics for weakly-textual contexts, and it concludes with a discussion of related work and future directions.

1 Introduction

Much of current textual CBR (TCBR) research focuses on transforming or augmenting knowledge-poor textual documents such as legal decisions (e.g, (Brüninghaus and Ashley 1997; Daniels and Rissland 1997; Weber *et al.* 1998)) or hotline support documentation (e.g., (Lenz 1998b)), in order to generate (hybrid-) structured representations that can be used by traditional knowledge-based CBR methods. Such support for textual CBR is especially important when the raw case information is composed entirely of free-form text. However, there are many situations in which textual information plays an important, but ancillary role in case composition and reasoning. For example, an architect might annotate structured design representations with text describing issues that arose in the course of creating a building design. When later retrieving

* The first author's research is supported in part by NASA under award No NCC 2-1035.

relevant designs for a new project, the architect could enter textual information about potential concerns in the new design context, and the set of retrieved cases could then be refined based on textual matching against previous design issues. In this scenario, the similarity measure is focused on the comparatively knowledge-rich design portion of the context, while knowledge-poor textual information lends additional support and refinement.

We are investigating how techniques from information retrieval, which have been widely used to support document-as-case TCBR, can be applied in situations where textual information represents only a small, but useful part of the overall reasoning context. This has led us to view textual CBR along a continuum from "weakly-textual," where textual information offers limited reasoning support but does not require sophisticated processing, to "strongly-textual," where textual information is the focus of reasoning but requires much more specialized treatment. For weakly-textual contexts, we propose that relatively simple and general information retrieval techniques can be used to provide a measure of textual distance that, although relatively weak in itself, is strong enough to enhance reasoning within a larger knowledge-based context. Such measures provide standard similarity metrics, such as nearest-neighbor methods, with support for "textual features," providing they are used with appropriate contextual support. This paper begins by describing our motivations for investigating weakly-textual methods in the DRAMA system for aerospace design. It characterizes textual CBR along a continuum of textual importance in case composition and reasoning, placing weakly-textual CBR in relation to current practice and motivating a focus on weakly-textual methods. The paper goes on to discuss methods currently being incorporated into DRAMA that integrate simple information retrieval techniques into standard CBR similarity metrics for weakly-textual contexts, and it concludes with a discussion of related work and future directions.

2 DRAMAtic Motivation

The DRAMA project provides an interactive support framework for aerospace design that combines case-based reasoning with "concept mapping" to capture and reuse expert design knowledge (Leake and Wilson 1999). DRAMA attempts to capture not only the important features of the aircraft designs themselves, but also the designer's conceptualizations of those designs—the relationships and rationale for making design choices. As part of rationale capture, particular design components (e.g., aircraft engines) are annotated with textual descriptions of the designer's rationale for choosing that component. The designer does not record rationale for the component as a whole (which could involve countless factors), but simply for why it is the *better* component in the current design context.

During design, DRAMA makes suggestions for appropriate aircraft components to use in the current design context. Distance-based retrievals are performed on the structured context of the current aircraft design in progress, and retrieved alternatives are listed in order of goodness of match according to a chosen design focus. We also wanted to allow the designer to enter additional criteria that could be matched against any textual annotations of rationale recorded by previous designers. For example, the designer could indicate in free-form text that fuel-efficient engines, if possible, should be considered more important in the current design. In order to enable this sort of relevance refinement, we used a very simple model of textual matching, based on information retrieval techniques (Salton 1983). This led us to consider how the metric might be improved in quality or extended for multiple types of textual annotations. In turn, this led to questions about tradeoffs in metric-engineering effort and required reasoning power, given that a great deal of design context was already at play. In addition, we were interested in how well information retrieval techniques would work with what would seem to be typically small pieces of text, and, given those considerations, we wanted to relate the work to current efforts in textual CBR.

To answer our questions, we propose in this paper (1) that weakly-textual CBR is an important area for investigation, (2) a way of looking at textual CBR as a whole that relates weakly-textual CBR to current methodology, (3) simple methods for supporting textual case features in weakly-textual contexts, and (4) an investigation of tradeoffs in case-textuality and required discrimination power for textual case-components in weakly-textual contexts.

3 A Textual CBR Continuum

In order to place our work in relation to prevailing textual CBR practice, we characterize TCBR in terms of textuality—here, the importance of text in the context of the goals of the reasoning process.

3.1 Defining Textual CBR

We start by proposing a working definition of textual CBR that will allow us to make finer distinctions about textuality. While the definition itself is fairly simple, it requires that some subtle distinctions hold.

First, the definition is made in terms of raw case context, that is the unmodified texts or textual components that will in some (often significantly modified or augmented) form be used for reasoning. These represent the raw experiences that a textual CBR system works with. For example, the definition would be applied to the texts of legal decisions prior to going through a specialized indexing process (manual or automatic) to support retrieval. If we did not make this distinction, some TCBR systems might not be considered textual at all, since their reasoning processes use fully-structured counterparts to actual documents.

Second, we presume that there is a sound reason for having the text as part of the experiential context: Either the text itself is an end (as in retrieving a legal decision) or the text represents useful knowledge that, if possible to do

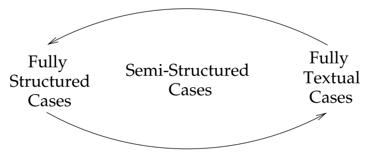

Figure 1: Semi-structured Case Knowledge

so, is either difficult or impractical to encode (as in many help-desk situations). Thus we presume that there is some value in the text itself, and that it is not simply a poor representational choice.

With these conditions in mind, we define textual CBR:

> *Textual CBR is CBR that will make use of text to enable or to enhance its reasoning process.*

3.2 Semi-Structured Cases

In making finer distinctions about textual CBR, it is important to understand the roles that text can play in case composition. Many textual CBR systems work with cases that correspond in their entirety to full-text documents. However, a case may also contain both textual and knowledge-rich components. For example, in a help-desk situation for printer support, the operator may represent structured information about the user's computer system (e.g., machine type, operating system) while filling in a textual description of the printer problem. Here, the textual and knowledge-based components together make up the raw case experience. Thus our definition of textual CBR does not presume that the text involved in reasoning comprises the whole of a case, in order to allow for such "semi-structured" cases.

It is useful to define what we mean by "semi-structured cases." *Semi-structured cases* are simple or complex cases that are composed in part of well-defined, knowledge-rich (having a rigorous semantic interpretation in the system context) structured features, as well as of ill-defined, knowledge-poor unstructured features such as free-form text. We view this distinction on the knowledge-level, instead of the representation or implementation levels (Dietterich 1986), so the fact that a document may be represented as a term-vector for measuring textual similarity does not change our view of the knowledge it affords as a document. Though our focus and use of "semi-structured cases" is on text, we note that other knowledge-poor components (e.g., audio or video) could comprise semi-structured cases.

A fully-textual document-case, then, can be viewed as a case with a single "document" feature that contains the full text of the document. This enables us to view case composition as a continuum of case types from fully-textual (single textual/document component) to semi-structured (some textual and some knowledge-rich components) to fully-structured (set of knowledge-rich components), as shown in Figure 1.

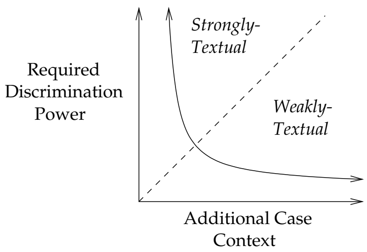

Figure 2: Feature-level Textuality

3.3 Strongly- & Weakly-Textual CBR

Given our working definition of TCBR and a view of semi-structured cases, we make the following finer-grained distinctions. In textual CBR, when the importance of text (in the raw experiential context) as a basis for reasoning exceeds the importance of the available knowledge-rich reasoning context, it is considered *strongly-textual CBR* (ST-CBR). When the importance of text as a basis for reasoning is overshadowed by the available knowledge-rich reasoning context, it is considered *weakly-textual CBR* (WT-CBR). Figure 2 shows the continuum between weakly- and strongly-textual CBR from the standpoint of an individual textual component. When little additional case context is available, it is very important to make distinctions based on the textual component in question, which will require greater discrimination power. As additional case context becomes available, it is less important to make fine distinctions based on the particular feature in question.

4 Textuality & Semi-structured Cases

In order to round out our discussion of textual CBR before focusing on methods for addressing weakly-textual situations, we consider some of the issues involved in working with semi-structured cases.

The textuality of a CBR system is closely related both to the type of case structuring and to the composition and textual processing power of the system's similarity metric. For purposes of indexing and retrieval, textual CBR systems can follow two related routes to varying degrees. First, cases can be transformed from more textual to more structured representations, enabling the use of traditional knowledge-based methods, as in (Brüninghaus and Ashley 1997). Second, similarity metrics can be made more complicated, perhaps involving multiple stages, as in (Daniels and Rissland 1997).

Figure 1 also illustrates a complementary relationship in moving between the extremes of case composition. A great deal of textual CBR can be viewed as enabling existing knowledge-based CBR methods by transforming fully-textual, fully-unstructured document-cases into more structured, knowledge-rich cases. From our point of view, the complementary goal is also important; that is, enabling the augmentation of fully-structured, knowledge-rich cases with fully-textual, knowledge-poor information by extending the functionality of standard similarity metrics to deal with textual components. It is to this that we turn our attention in later sections.

The idea of moving from a single-featured document-case to a set of more knowledge-based features is compelling. We can view the process of transforming a textual case into a more structured case as happening in two possible ways. First, text can be transformed into or augmented with knowledge-rich features, either manually or automatically as with information extraction techniques (e.g., (Lenz and Glintschert 1999)). Second, components with larger amounts of text could be transformed into multiple textual components, each containing a portion of the original text and providing a more narrowly defined contextual interpretation of text therein. For example, if some textual information about a plaintiff in a legal case can be identified, the "plaintiff" textual component could be matched against plaintiff textual components of other cases. Both help to fill-in otherwise empty contextual portions of case knowledge and can strengthen reasoning. The former enables more powerful reasoning, but the latter may require less processing.

This gives rise to two interesting and related issues. The first is whether it would be possible for a set of many contextualized textual features to perform as well as some knowledge-rich representations. Though at a different level (we are prompting for empirical investigation), this question is similar to questions about the the relative roles of deep and surface features in analogical reminding (e.g., (Gentner *et al.* 1993; Hammond *et al.* 1991)). A related question is whether it is possible to reduce the amount of effort required to transform a case or to apply a complex similarity measure by judging when enough new contextual knowledge has been gained to make reasoning practicable, and discontinuing the knowledge-building effort at that point.

Having described weakly-textual CBR, semi-structured cases, and surrounding issues, we turn our attention to the practical problem of augmenting similarity metrics to enable the addition of textual components in fully-structured case representations.

5 Representing Textual Features in WT-CBR

We are currently updating DRAMA's textual relevance adjustment mechanism with methods that have been developed and tested in Rosetta, an information retrieval system for digital libraries of scientific literature (Bradshaw and Hammond 1999). Rosetta parses the text surrounding citations to a particular research article within other documents and extracts noun phrases that have been used to describe the cited work. These noun phrases are used to index the articles, because the information descriptions they provide match queries for the information the documents contain for a large percentage of queries. The small-text processing in Rosetta seems to be a good match for DRAMA's weakly-textual requirements, and the following sections describe the details of the implementation methodology as it is being applied in DRAMA.

Our model of textual feature similarity is based in a straightforward way on simple standard IR techniques (Salton 1983) that can be easily integrated with nearest-neighbor methods. In weakly-textual CBR, cases will typically contain a small number of textual features, each containing a small body of unstructured text. Since the textual components will not be the predominant influence on overall similarity measurement, an approach to computing the similarity between such features should be easy to implement and require little or no knowledge engineering effort. More generally, the amount of effort required should be proportional to the value added.

Textual features are treated in a manner similar to standard feature types, such as interval-scaled, ordinal, or nominal. Similarity distance between textual features will be measured as a continuous value between 0 and 1 that can be combined with other distance measures in the nearest-neighbor metric. Sometimes individual textual features will take on greater relative importance, so we have designed the metric to allow for additional processing power by adding layers, such as described in (Lenz 1998a), from domain-independent to domain-specific and from low-power to high-power, depending on the user's needs and task context.

5.1 The Vector-Space Model

Our approach is based on the well-established vector-space model (VSM) (Salton *et al.* 1971). Though the general VSM has been presented in other textual CBR contexts (e.g., (Brüninghaus and Ashley 1997)), we present it here to motivate some of our own modifications for use in weakly-textual CBR. Since we will be using some information retrieval parlance, we take "document" in these sections to refer to any textual case feature. In the VSM, each document is represented by a vector (list) of *terms* with associated weights that describe a term's value as a label for a document, and a term is defined as a word or phrase used in a document. Representing documents as vectors of terms makes it possible to compare two documents using vector algebra, in particular, the cosine measure of similarity (Salton and McGill 1983). Using this technique, the degree to which two documents are similar is determined by the cosine of the angle between the vectors in an n dimensional space, where n is the number of unique terms used in the entire collection, and the magnitude of a document vector for a given term is the weight of that term. The cosine measure has been used in a variety of applications (Kulyukin 1998; Pazzani *et al.* 1996; Deerwester *et al.* 1990) and has been established as an effective means of comparing two documents (Salton 1983).

In our similarity model, there are as many vector-spaces as there are textual features for a TCBR system, because each textual feature (across all cases) may have a different set of textual semantics. In the same way that each ordinal feature has a set of possible ordinal values, each textual feature has a set of possible meanings defined by the information presented in each case that has a value for that feature. Therefore, if we are to represent textual features with term vectors, evaluation of the differences and similarities among cases for a particular textual feature must be performed in a unique vector space, separate from the vector spaces for other textual features.

5.2 Term Selection

In the VSM, term vectors are constructed using a term selection metric, which measures the descriptive value of terms used in a document. Most applications of the VSM use a term selection metric based on $tf \cdot idf$ (Salton and Buckley 1988). $tf \cdot idf$ is actually a metric that both selects terms to use in term vectors and ranks those terms in order of importance. With $tf \cdot idf$ the weight of a term is the product of the frequency with which a term is used in a document (tf) and a value that varies inversely as the number of documents in the collection which use the term (idf). The top k terms are then used as the term vector, where k is a system-dependent value for the size of term vectors. For large bodies of text $tf \cdot idf$ works well; however, for small pieces of text there may often not be enough data for a statistical evaluation of the descriptive value of terms. This is especially important if we are to use phrases as terms, since statistical approaches to phrase detection are based on the frequency with which words co-occur. Many textual features are composed of short text segments as well as long ones; therefore, it is important that our model deal effectively with both types. Consequently, our approach to term selection does not rely on a measure of the frequency with which a term is used.

In selecting the terms that accurately represent a textual case feature, it is important to build term vectors that make good descriptors for the topics addressed. To find such terms in the absence of useful frequency information we use a low-overhead Natural Language Processing (NLP) approach in which we leverage freely-available tools to extract noun phrases and other content-bearing terms as descriptors of content.

Noun phrases typically contain the bulk of the content-bearing terms in a piece of text and have been used as the basis for representing documents in a variety of IR applications (Bradshaw and Hammond 2000; Franklin and Bradshaw 1999; Schatz *et al.* 1999; Thompson 1992). To extract noun phrases from a piece of text we use a simple algorithm that employs the Moby lexicon (Ward 1994) as a means of determining parts of speech. This lexicon contains 230,000 English words and lists the part of speech sorted by the frequency with which a word is used as that part of speech. Using this lexicon the part-of-speech tagger needs no training, a feature that makes this approach simple to implement in a CBR system. The noun phrases selected are those in which a noun is modified by pre-noun modifiers such as adjectives and other nouns or by post-noun modifiers in the form of prepositional phrases. Examples of these kinds of phrases are: "top air speed" and "testing of navigation equipment". We consider as a noun any word that can be used as a noun according to the lexicon, and we consider as an adjective any word that can be used as an adjective. We treat a word that can be used as both a noun and an adjective as a noun. Furthermore, even though domain-specific words are rarely found in the lexicon they are often content bearing (Chung *et al.* 1999). Therefore, we treat such words as nouns and extract them as such when they occur in a sequence of words consistent with the syntax of a noun phrase.

The system extracts both noun phrases and “noun phrases” containing unrecognized words by evaluating word sequences ranging in length from one to k, where k is a application-specific value. Phrases matching the criteria described above are extracted and added to the term vector for the textual feature being evaluated. For example, the following piece of text might occur in the associated remarks of an aircraft engine component:

> We found that the Allison 150+-AD engine has a thrust capability of 580 lbs. which is slightly above the thrust rating of 575 lbs. advertised by Allison. The 150+-AD is also relatively fuel-efficient.

For this feature, terms such as *"Allison 150+-AD engine"*, *"150+-AD engine"*, *engine*, *"thrust capability"*, *thrust*, and *fuel-efficient* are extracted and used as elements of the term vector representation.

5.3 Term Weighting

After a term vector has been constructed for each textual feature, a weight is assigned to each term in those vectors. Term weights are calculated using the following metric,

$$w_{tc} = \frac{n_{tc}}{1 + \log N_t} \tag{1}$$

where w_{tc} is the weight of term t as a label for textual feature c, n_{tc} is the number of times term t was used in c, and N_t is the number of cases which contain a textual feature that uses the term t. In this term weighting scheme we make the assumption that if a term is used repeatedly it is an important indicator of content. So in the example passage above, assuming all the terms extracted were distributed equally in the case library, the terms, *Allison*, *thrust*, and *150+-AD* are given greater weight and therefore considered more important than other terms such as *fuel-efficient*, because they were used more frequently. Alternatively, for the following textual component:

> We compared the 150-AD to a variety of other engines from Allison and found that the 150-AD is the most fuel-efficient for the thrust it provides. In addition, we performed fuel-efficiency comparisons between the 150-AD and a variety of engines from other manufacturers and again found it to be the most fuel-efficient. Thrust matched the advertised value of 485 lb.

fuel-efficient and *150-AD* are used multiple times and thus weighted more heavily than *thrust* which is only used once.

6 Integrating Textual Features with Measures of Case Similarity

To use textual features in nearest-neighbor methods for measuring similarity between cases we need a gauge by which to judge the similarity between such features. For this we use the standard cosine correlation (Salton and McGill 1983) defined by the following equation:

$$cos(c_i, c_j) = \frac{\sum_{k=1}^{T}(t_{ik} \times t_{jk})}{\sqrt{\sum_{k=1}^{T}(t_{ik})^2 \times \sum_{k=1}^{T}(t_{jk})^2}} \tag{2}$$

Where c_i and c_j are the two features to be compared and t_i and t_j are the weights of terms as descriptors in a term space of dimension T. Therefore, the distance between the corresponding textual features of two cases is determined by the cosine measure, and is used as any other normalized feature distance would be in computing a weighted Euclidean measure.

Our implementation uses three distinct modules that may be replaced or supplemented as needed. These modules are the *term-selector*, the *term-weighter*, and the *component-matcher*. We chose this model to make it easily adaptable to many types of reasoning requirements, allowing for the application of different knowledge layers, such as described in (Lenz 1998a). For example, in a system where textual components are fairly large documents, it might be desirable to fine-tune the term-weighting module to better represent the frequency with which terms occur. Moreover, with this model it is straightforward to add stronger methods of term-selection and term-weighting. For example, to capture different uses of the same root word and other relationships such as synomony one might implement the use of a thesaurus such as WordNet (Miller 1995). Note that using a thesaurus simply broadens the definition individual terms; the basic framework of selection, weighting, and matching remains the same.

7 Brief Example

To illustrate the value of integrating textual features with case similarity measures we look at an example using two simplified examples of DRAMA engine components, as shown in figure 3 (the text fragments correspond to earlier example passages).

These two examples describe distinct aircraft engines maintained as design components by DRAMA. For a query in which the feature requirements are a diameter of 11 inches, a weight of 40 pounds, a thrust rating ranging from 400 lb. to 600 lb., and the remark "Need an engine that is very fuel-efficient" the system would have a difficult time ranking the two engines relative to one another, since the remark does not come into play. With our integrated model of textual similarity, however, the system determines that the 150-AD is a closer match, because the remarks feature for the 150-AD engine is represented by a term vector in which the term *fuel-efficiency* figures prominently.

name	Allison 150+-AD	Allison 150-AD
type	military	military
thrust	550 lb.	485 lb.
diameter	11.6 in.	11.6 in.
weight	41.0 lb.	41.0 lb.
remarks	We found that the Allison 150+-AD engine...	We compared the 150-AD to a variety...

Figure 3: Simplified Examples of DRAMA Aircraft Engine Cases

8 Related Work

Our methods are closely related to other work in textual CBR for legal reasoning (Brüninghaus and Ashley 1997; Daniels and Rissland 1997; Weber *et al.* 1998), for help-desk systems (Kunze and Hübner 1998; Racine and Yang 1997), for question answering (Burke *et al.* 1997), and for corporate experience-sharing (Baudin and Waterman 1998) in leveraging information retrieval and natural language processing techniques to support reasoning with texts. Our focus, however, is on enabling the use of textual reasoning to support ends that are not necessarily, though could be, document-centric. Although we make use of the same types of techniques, we take a different view of the level at which the techniques operate. In WT-CBR, the texts are important but ancillary, and the processing effort should reflect that focus. We presume, but do not require, that textual case components contain only small passages of text. While they do not make explicit use of textual features, a number of systems have made use of reduced textual passages for identifying document passages (Daniels and Rissland 1997; Burke *et al.* 1997) and for focusing document indexing (Brüninghaus and Ashley 1999). Our view of the possibilities for strengthening, when necessary, the textuality of a system rely on increasingly sophisticated domain-independent and domain-dependent knowledge, such as outlined in (Lenz 1998a). Additionally, the idea of developing techniques for semi-structured cases is related to the notion of a "middle-ground" between high- and low- cost retrieval performance (Brown *et al.* 1998), but our is focused on the relative strength required by the overall reasoning task, rather than the specificity of a query in a strongly-textual context.

Focusing on a set of weakly-textual case components is very much in the spirit of boosting (Schapire 1999), where a number of weak, moderately inaccurate "rules of thumb" are learned and combined to form a strong overall classification rule. Weakly-textual features can be seen as the lazy-learning analogues of the eagerly-derived weak component rules in boosting. Accurately retrieving instances comprised completely of textual features has been investigated from a database perspective in the WHIRL approach (Cohen 1998), which is closely related to the WT-CBR methods proposed here. Both techniques show strong results based on weak components and support the theory and practice of using weakly-textual components in CBR.

9 Conclusion & Future Directions

We have introduced a new way of viewing textual CBR that illuminates the differences between strongly- and weakly-textual CBR, motivated by work in the DRAMA system for aerospace design. We have also presented a practical model for augmenting standard similarity metrics that enables the use of textual components in weakly-textual contexts. Weakly-textual CBR has interesting and practical applications, and our investigation has identified interesting issues for further research. We are working on methods for evaluating WT-CBR techniques that will test the benefits of WT-CBR in contexts of differing textuality. We anticipate that more powerful techniques will not greatly increase the benefits for our methods in weakly-textual CBR, but that as the available knowledge-rich context decreases, more powerful methods will need to be applied in order to maintain the same level of performance. As textual CBR evolves, we expect that weakly-textual methods will play an increasingly important role in developing CBR systems that make use of texts at various levels.

References

Catherine Baudin and Scott Waterman. From text to cases: Machine aided text categorization for capturing business reengineering cases. In Mario Lenz and Kevin Ashley, editors, *Proceedings of the AAAI-98 Workshop on Textual Case-Based Reasoning*, pages 51–57, Menlo Park, CA, 1998. AAAI Press.

Shannon Bradshaw and Kristian Hammond. Constructing indices from citations in collections of research papers. In *Proceedings of the 62nd Annual Meeting of the American Society for Information Science*, Washington, D.C., Oct. 31 - Nov. 4 1999. To Appear.

Shannon Bradshaw and Kristian Hammond. Guiding people to information: Providing an interface to a digital library using reference as a basis for indexing. In *Proceedings of the 2000 International Conference on Intelligent User Interfaces*, New Orleans, LA, January 2000. Submitted.

Mike Brown, Christiane Förtsch, and Dieter Wissmann. Combining information retrieval and case-based reasoning for “middle ground” text retrieval problems. In Mario Lenz and Kevin Ashley, editors, *Proceedings of the AAAI-98 Workshop on Textual Case-Based Reasoning*, pages 3–7, Menlo Park, CA, 1998. AAAI Press.

Stefanie Brüninghaus and Kevin D. Ashley. Using machine learning to assign indices to legal cases. In *Proceedings of the Second International Conference on Case-Based Reasoning*, pages 303–314, Berlin, 1997. Springer-Verlag.

Stefanie Brüninghaus and Kevin D. Ashley. Bootstrapping case base development with annotated case summaries. In *Proceedings of the Third International Conference on Case-Based Reasoning*, 1999. In Press.

Robin D. Burke, Kristian J. Hammond, Vladimir Kulyukin, Steven L. Lytinen, Noriko Tomuro, and Scott Schoenberg. Question answering from frequently asked question files: Experiences with the FAQ finder system. *AI Magazine*, 18(2):57–66, 1997.

Yi-Ming Chung, Qin He, Kevin Powell, and Bruce Schatz. Semantic indexing for a complete subject discipline. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, Berkeley, CA, Aug. 1999.

William W. Cohen. The WHIRL approach to integration: An overview. In *Proceedings of the AAAI-98 Workshop on AI and Information Integration*. AAAI Press, 1998.

Jody J. Daniels and Edwina Rissland. What you saw is what you want: Using cases to seed information retrieval. In *Proceedings of the Second International Conference on Case-Based Reasoning*, Berlin, 1997. Springer-Verlag.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, September 1990.

T. Dietterich. Learning at the knowledge level. *Machine Learning*, 1:287–316, 1986.

David Franklin and Shannon Bradshaw. Beyond "Next slide, please": The use of context and speech in multi-modal control. In *In the Working Notes of the AAAI-99 Workshop on Intelligent Information Systems*, 1999.

D. Gentner, M. Rattermann, and K. Forbus. The roles of similarity in transfer: Separating retrievability from inferential soundness. *Cognitive Psychology*, 25:524–575, 1993.

Kristian J. Hammond, Colleen M. Seifert, and Kenneth C. Gray. Functionality in analogical transfer: A hard match is good to find. *Journal of the Learning Sciences*, 1(2):111–152, 1991.

Vladimir Kulyukin. *Question Driven Information Retrieval Systems*. PhD thesis, University of Chicago, 1998.

Mirjam Kunze and André Hübner. Textual CBR case studies of projects performed. In Mario Lenz and Kevin Ashley, editors, *Proceedings of the AAAI-98 Workshop on Textual Case-Based Reasoning*, pages 58–61, Menlo Park, CA, 1998. AAAI Press.

D. Leake and D. Wilson. Combining CBR with interactive knowledge acquisition, manipulation, and reuse. In *Proceedings of the Third International Conference on Case-Based Reasoning*, Berlin, 1999. Springer-Verlag. In press.

Mario Lenz and Alexander Glintschert. On texts, cases, and concepts. In *Proceedings of XPS-99*, Berlin, 1999. Springer-Verlag.

Mario Lenz. Defining knowledge layers for textual case-based reasoning. In P. Cunningham, B. Smyth, and M. Keane, editors, *Advances in Case-Based Reasoning: Proceedings of EWCBR-98*, number 1488 in Lecture Notes in Artificial Intelligence, Berlin, 1998. Springer-Verlag.

Mario Lenz. Knowledge sources for textual CBR applications. In Mario Lenz and Kevin Ashley, editors, *Proceedings of the AAAI-98 Workshop on Textual Case-Based Reasoning*, pages 24–29, Menlo Park, CA, 1998. AAAI Press.

Mario Lenz. Textual CBR and information retrieval - a comparison. In Lothar Gierl and Mario Lenz, editors, *Proceedings of the Sixth German Workshop on Case-Based Reasoning*. University of Rostock, 1998.

George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, November 1995.

M. Pazzani, J. Muramatsu, and D. Billsus. Syskill & webert: Identifying interesting web sites. In *Proceedings of the National Conference on Artificial Intelligence*, 1996.

K. Racine and Q. Yang. Maintaining unstructured case bases. In *Proceedings of the Second International Conference on Case-Based Reasoning*, pages 553–564, Berlin, 1997. Springer Verlag.

Gerard Salton and Chris Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

Gerard Salton and M. J. McGill. *The SMART and SIRE Experimental Retrieval Systems*. McGraw-Hill, New York, 1983.

G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1971.

Gerard Salton. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

Robert E. Schapire. A brief introduction to boosting. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1401–1406, San Francisco, 1999. Morgan Kaufmann.

Bruce Schatz, William Mischo, Timothy Cole, Ann Bishop, Susan Harum, Eric Johnson, Laura Neumann, Hsinchun Chen, and Dorbin Ng. Federated search of scientific literature. *IEEE Computer*, 32:51–59, February 1999.

Roger Thompson. OCLC online computer library center, inc. In *Proceedings of the First Text Retrieval Conference*, Gaithersburg, MD, Nov. 4-6 1992.

Grady Ward. A set of lexical resources compiled by grady ward. http://www.dcs.shef.ac.uk/research/ilash/Moby/, 1994.

Rosina Weber, Alejandro Martins, and Ricardo M. Barcia. On legal texts and cases. In Mario Lenz and Kevin Ashley, editors, *Proceedings of the AAAI-98 Workshop on Textual Case-Based Reasoning*, pages 40–50, Menlo Park, CA, 1998. AAAI Press.