The Synergistic Application of CBR to IR

by

Edwina L. Rissland
and
Jody J. Daniels

CMPSCI Technical Report 95-47

COMPUTER SCIENCE DEPARTMENT, LGRC
UNIVERSITY OF MASSACHUSETTS
BOX 34610
AMHERST MA 01003-4610

The Synergistic Application of CBR to IR*

Edwina L. Rissland and Jody J. Daniels
Department of Computer Science
University of Massachusetts
Amherst, MA 01003

rissland, daniels @cs.umass.edu

Abstract

In this paper we discuss a hybrid approach combining Case-Based Reasoning (CBR) and Information Retrieval (IR) for the retrieval of full-text documents. Our hybrid CBR-IR approach takes as input a standard symbolic representation of a problem case and retrieves texts of relevant cases from a document corpus dramatically larger than the case base available to the CBR system. Our system works by first performing a standard HYPO-style CBR analysis and then using texts associated with certain important classes of cases found in this analysis to "seed" a modified version of INQUERY's relevance feedback mechanism in order to generate a query composed of individual terms or pairs of terms. Our approach provides two benefits: it extends the reach of CBR (for retrieval purposes) to much larger corpora, and it enables the injection of knowledge-based techniques into traditional IR. We describe our CBR-IR approach and report on on-going experiments.

* *This research was supported by NSF Grant no. EEC-9209623, State/Industry/University Cooperative Research on Intelligent Information Retrieval, Digital Equipment Corporation, and the National Center for Automated Information Research.*

Table of Contents

1. Introduction

One strength of Case-Based Reasoning (CBR) systems is the ability to reason about a problem case and perform highly intelligent problem-solving, such as the generation of legal arguments or detailed operational plans [Kolodner, 1993]. In particular, CBR systems have at their core the ability to retrieve highly relevant cases. However, CBR systems are limited by the availability of cases actually represented in their case bases. Among current case-based reasoning systems few have large case bases (say, larger than 1000 cases). Those systems that have supported large case bases—containing thousands or even tens of thousands of cases—have employed simple case representations (e.g., MBRtalk [Stanfill & Waltz, 1986], PACE [Crecy et al., 1992], Anapron [Golding & Rosenbloom, 1991]). Our own CBR systems—HYPO [Rissland & Ashley, 1987; Ashley, 1990], CABARET [Rissland & Skalak, 1991], BankXX [Rissland et al., 1994a, 1994b]—perform in-depth reasoning to produce sophisticated precedent-based legal arguments, challenging hypothetical cases, interpretations of ill-defined legal concepts, etc. They use detailed case representations and have typically had case bases in the range of three to five dozen cases.

On the other hand, within the information retrieval (IR) world, there are many huge document collections, such as those commonly available in fields like law, business, or medicine, and individual cases are often very large (e.g., tens of pages of text). For instance, all the cases decided in the Supreme Court and other Federal courts since their beginnings (in 1789) and most state courts over at least the last 35 years are available through West Publishing Company's *WestLaw.*® However, the level of representation is shallow—the text itself. Thus, although full-text IR systems are not hampered by any lack of available cases (in textual form), they cannot reason about them and they cannot apply a highly articulated sense of relevance such as that found in CBR systems. Rather, text-based systems rely on broadly applicable methods, such as statistical measures, to define relevance [Salton, 1989]. Nonetheless, we would still like to be able to access these collections in a more intelligent, problem-based manner.

Such massive on-line corpora represent a tremendous resource and investment of capital. They are the stock-in-trade of many professionals, such as lawyers who use them extensively in legal research. Given their awesome scope and ready availability, it is simply not realistic to think about redesigning such text collections to suit the requirements of symbolic AI approaches, such as CBR. Thus, such collections, built up over the years, will most likely remain in their current textual form and be accessed pretty much as they are or not at all.

Of course, current text-based systems are no guarantee for intelligent retrieval. The user of such a system must know how to manipulate them to get truly relevant information back. Often users are not even aware of the difficulties in using such a system because nothing has appeared to go wrong. For instance, one study found that although the users felt that they had retrieved most of the right texts (i.e., that recall was high), in fact, they had only retrieved a mere 25% of the relevant texts [Blair & Maron, 1985].

The recurring opposite problem is retrieving too much information, only some of which is really relevant. Bringing in specifics of the case at hand is one way to deal with this sort of problem. This is what an experienced user does and what the vendors of such systems recommend. That sort of information is exactly the kind used by CBR systems. In addition to facts of the current case, information from known relevant precedents, past successful approaches to similar retrieval problems, particular knowledge of the domain, etc. can also be used. By being smart about query formation and other manipulations of the system, a user can drive a standard text-based retrieval engine to produce good results. We would like this to happen automatically without the currently assumed level of intervention or expertise on the part of the user.

Another problem that users encounter with traditional IR systems is that of composing effective queries. Even with natural language capabilities, users do not use IR systems effectively as they could. One recent study of users of the Library of Congress's THOMAS system, which uses the same INQUERY retrieval engine that we use, found that over 88% of all queries were composed of 3 or fewer terms [Croft et al., 1995].[1].

Thus we have two well-developed technologies, each with its own strengths and limitations. CBR is highly intelligent but limited in its reach and IR is easily applied but not able to reason in any depth. Consequently, a natural approach is to form a hybrid system to produce results or functionalities unachievable by either individually.

Our goal in this project is to take advantage of the highly articulated sense of relevance used in CBR and the broadly applicable retrieval techniques used in IR in order to

[1] Out of 25,321 queries 22,318 involved 3 or fewer terms. A mere 106 queries involved 6 or 7 terms; only 1 used 8 terms; none used more than 8. [Croft et al., 1995].

retrieve documents that are relevant to a problem case from commonly available large text bases. We would like to do this without sacrificing the high accuracy of CBR retrieval and without enlisting the aid of an army of knowledge engineers to re-tool available text collections. Therefore, a central question in our research is: *Can we automatically formulate good queries to an IR system based on information derived by a CBR system?*

In our hybrid CBR-IR approach, we first perform a standard HYPO-style CBR analysis [Ashley, 1990], and then use the results to cause the INQUERY IR system [Callan et al., 1992] to generate and act on a query. This is done by applying a modified version of INQUERY's relevance feedback (*RF*) mechanism to the documents associated with a subset of cases found during the CBR analysis, such as the most on-point cases. From this small set of "seed" documents—called the *relevance feedback case-knowledge-base* or *RF-CKB*—the RF mechanism selects and weights terms to form a query to the larger text corpus. This use of relevance feedback, in effect, tells the IR component that *the cases found through the CBR analysis are highly relevant and that INQUERY should retrieve more like them.*

The CBR analysis is performed with respect to the relatively small case base available to the CBR component. Relevance feedback is based on a subset of noteworthy CKB cases—the RF-CKB—selected from this analysis. Our RF-CKB's are smaller than sets usually used in relevance feedback. The IR can be performed on a text collection of arbitrary size. In one of our application domains, an area of tax law, the full-text collection is 500 times larger than the CBR module's case base; in the other, an area of bankruptcy law, it is about 20 times larger.

What the user gets back is a set of relevant texts retrieved from the document corpus. Any further analysis of these retrieved texts, for instance for the purpose of making a case-based argument, is up to the user.

Our approach works to the benefit of both CBR and IR by extending the reach of CBR and adding knowledge-based methods to traditional IR. It allows the results of the small-scaled CBR to be leveraged to collections dramatically larger than is usual in CBR. Since items in the larger document corpus are only "represented" in text form, they are not amenable to knowledge-based methods, in particular indexing and retrieval techniques used by CBR, and thus would ordinarily be beyond the reach of standard CBR. On the other hand, our approach injects knowledge-based reasoning—in particular, the highly articulated sense of relevance used in CBR—into traditional text-based IR. Knowledge-intensive indexing of the kind at the core of CBR is simply not possible in standard IR.

Our hypothesis is that the quality of documents retrieved via this hybrid approach is better than via IR methods alone. (The numbers of items that can be retrieved are certainly larger than with CBR alone.) This hypothesis has been borne out in our experiments, where our approach achieves a very fine level of performance, as measured by the standard measures of precision and recall.

In the next section, we discuss some past work on combining CBR and IR concerns. In Section 3, we give an overview of our approach and provide background on HYPO-style CBR and the INQUERY retrieval engine. In Section 4 we run through an example in detail. In Section 5, we present methodological details, such as how we built our test collections. In Section 6, we describe the various RF-CKB's that we use. In Section 7, we introduce our experiments, and in particular, various baselines for comparison to IR used alone without CBR. In Section 8, we present and discuss our results.

2. Background

There have been several approaches for enriching retrieval environments with knowledge-based methods. Recently, Hafner and Wise used expert systems technology to help users pose requests to standard IR environments [Hafner & Wise, 1993]. Earlier work in legal conceptual information retrieval (e.g., [Hafner, 1987a, 1987b], [Bing, 1987], [Dick, 1987]) relied on a graph of diverse legal entities and concepts where labeled links captured influences and taxonomic information.

Rose's SCALIR [Rose, 1994; Rose & Belew, 1991] is a hybrid symbolic/sub-symbolic system that uses a network of legal knowledge, including Shepard's links and West's key number taxonomy links, to perform retrieval. SCALIR uses spreading activation to perform the retrieval. Approximately 90% of the links in the SCALIR network are weighted connectionist links, with 75% of all the links between cases and terms.

A few other projects have tried to bridge the gap between CBR and IR. For instance, Aleven and Ashley performed some exploratory studies, as an off-shoot of their CATO project, on how knowledge of legal factors could be used in the formulation of natural language queries in Westlaw's WIN interface [Aleven & Ashley, 1993]. Specifically, they found that students taught to argue with factors can use them to produce good queries by expressing the factors in natural language (and then using WestLaw's WIN for retrieval). Their study did not involve the automatic generation of queries, but it would not be hard to do so. A potential problem, however, might be the limited number of terms in the query.

Goodman explored the opposite tack: enhance CBR with IR. This was done in the Prism system, a system for classifying bank telexes for further distribution and routing [Goodman, 1991]. Prism integrates IR methods for automatic index generation into the CBR paradigm. It uses a lexical pattern matcher to generate retrieval indices. Prism uses the retrieval indices to select cases from a case-base of over 9600 sample telexes. It then adapts the best matching cases to find classifications for the new telex.

In the FRANK project [Rissland et al., 1993], we explored how knowledge of the user's intended purpose for retrieving information—writing a one-sided pro-position advocacy brief, a balanced pro-con policy assessment memo, etc.—can be used to help configure CBR in order to retrieve useful cases. The high level purposes—the user's information needs—are used to specify what sort of cases to seek and which notions of similarity to use with the CBR.

In the BankXX project [Rissland et al., 1993, 1994a,b, 1995], we explored the use of heuristic search as a program architecture for legal information retrieval. We represented components of argument at various levels of abstraction, for instance in so-called *argument pieces* and *argument factors,* and in various core components of the system, such as its evaluation function. These cause BankXX to search for, peruse, and possibly harvest information of known utility for making precedent-based arguments, such as ordinary and best pro and con cases, legal theories. Such considerations also insure that the information BankXX retrieves is balanced in the sense that not all of it is cases, not all of the cases are for one side, etc.

In the CABARET project [Rissland & Skalak, 1991], we created a theory of statutory interpretation to guide not only the argumentative tasks pursued but also the type of cases to be retrieved in support of them. Our three-tier model of statutory argument—consisting of argument *strategies, moves,* and *primitives*—specified the types of cases needed to carry out various aspects of argument, such as "broadening" a rule by finding cases that do not satisfy all rule prerequisites but still were allowed to reap the benefit of the rule's conclusion (e.g., an allowed tax deduction) [Skalak & Rissland, 1992].

3. System Overview

Our approach combines knowledge-based CBR with text-based IR. It works by first performing a CBR analysis of the input problem case and then using the results of the CBR analysis to drive text-based document retrieval. The CBR is performed by a HYPO-style module [Ashley, 1990] and IR is performed by the INQUERY retrieval engine [Callan et al., 1992].

In particular, our system (see Figure 1) first uses its HYPO-style CBR module to analyze the problem case with respect to the cases that are represented in its own *case knowledge base* (*CKB*). This produces a sorting—actually a partial ordering—of cases relevant to the problem case according to how *on-point* they are (based on the model of relevance and on-pointness used in HYPO-style systems). The result of this CBR analysis is represented in a so-called *claim lattice.* See Figure 2 for an example.

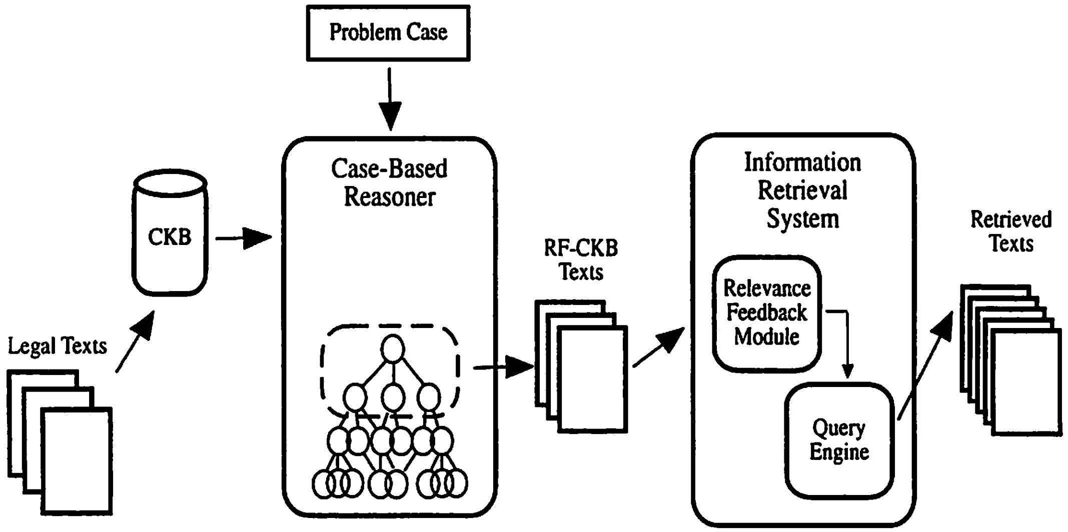

Figure 1. Overview of the hybrid CBR-IR architecture.

Next, our hybrid CBR-IR system selects a small number of certain special kinds of important cases from the claim lattice, for instance, the most on-point cases (i.e., maximal cases in the on-point ordering), to serve as examplars of the kind of documents that we would like the IR engine retrieve. Then the texts associated with these selected exemplar cases—called the *relevance feedback case-knowledge-base* or *RF-CKB*—are passed to a modified version of the relevance feedback mechanism of the INQUERY retrieval engine, which then generates a standard query consisting of the top *n* terms or top *n* pairs of terms from them. In the work reported here, for the texts we use the full texts of court opinions. We have experimented with a variety of RF-CKB's (see Section 6).

Note that ordinarily, INQUERY would not engage in relevance feedback until a retrieval, based on user input, had been made and the retrieved documents presented to and tagged by the user as to their relevance. Our system uses "feedback" in the form of the RF-CKB on a null query. This use of relevance feedback, in effect, tells the IR component that the RF-CKB cases are highly relevant and that INQUERY should retrieve more like them.

Once the query is generated, it is processed as usual and full-text documents are returned to the user. Of course, the system performs no analysis on retrieved texts. That would require natural language understanding of an unprecedented scale.

Although the CBR analysis is done with respect to the relatively small CKB available to the CBR component, and relevance feedback is done with respect to the even smaller set of RF-CKB cases, the actual retrieval can be performed on text collections of arbitrary size. In effect, our system leverages its own "in-house" analysis of the problem case to a full-blown retrieval from an "outside" document base. Instead of requiring the user to make up a query in order to initiate a retrieval, in our approach the user initiates the process by inputting facts of a case and the system carries on automatically from there.

For this project, we did not design new case representations (for problem cases and cases in the CKB). Rather, we used *as is* the representations developed in two past CBR projects from our lab: CABARET [Rissland & Skalak, 1991] and BankXX [Rissland et al., 1994a, 1994b]. Both projects used a standard frame-based representation for cases, in which specific facts fill designated slots. CABARET was a mixed paradigm system that used case-based and rule-based reasoning to analyze cases in an area of tax law dealing with the so-called *home office deduction*, as specified in Section 280A(c)(1) of the Internal Revenue Code. BankXX is a CBR system that uses an architecture based on heuristic search to guide retrieval of information important for case-based argument. BankXX is implemented in an area of bankruptcy law dealing with the *good faith* requirement for approval of personal (Chapter 13) debtor plans, as specified in Section 1325(a)(3) of the Bankruptcy Code.

3.1 Background on HYPO-style CBR

In the CBR portion of the system, we use a CBR engine of the HYPO-style, with which we have had extensive experience: in HYPO [Rissland et al., 1984; Rissland & Ashley, 1987; Ashley, 1990], in CABARET [Rissland & Skalak, 1991]; in FRANK [Rissland et al., 1993], and in BankXX [Rissland et al., 1994a, 1994b]. In brief, HYPO-style CBR engines work as follows.

First, a problem case is input and analyzed to see what *dimensions* [Rissland et al., 1984], sometimes also called *factors*, are applicable in the problem case. Dimensions address important legal aspects of cases and are used both to index and compare cases. They represent different argumentative approaches for dealing with an issue. For instance, the dimension called *relative-home-work-time* focuses on the percentage of total work time spent in the home office [Rissland & Skalak, 1991]. It represents one line of reasoning about the issue of whether the taxpayer's home office is his "principal place of business." It is an aspect of the so-called *focal point test*, used for many years by the tax courts.

Second, any case in the case-knowledge-base sharing at least one applicable dimension with the problem case is retrieved. These are considered the minimally *relevant* cases.

Third, relevant cases are sorted according to *how* on-point they are. In this sorting, which results in a partial order, *Case A* is considered *more on-point* than *Case B* if the set of applicable dimensions *Case A* shares with the problem case properly contains those shared by *Case B* and the problem case. Maximal cases in this ordering are called *most on-point cases* or *mopc's*. The result of sorting the cases can be shown in a so-called *claim lattice*. (See Figure 2 for an example.) Those cases on the top level of the lattice are the mopc's. The problem case is the root node.[2] Note, the claim lattice is usually the starting point for other aspects of CBR, such as the generation of arguments or creation of hypotheticals. However, in this project, we only use CBR to generate claim lattices.

[2] If there is a case with exactly the same applicable dimensions, it too would appear in the root.

3.2 Background on INQUERY

We use the INQUERY retrieval engine as our IR component. INQUERY uses a Bayesian probabilistic inference net model [Turtle & Croft, 1991]. It uses a directed acyclic graph with a query node at the root, document nodes at the leaves, and a layer of query concept nodes and a layer of content representation nodes in between. Nodes that represent complex query operators can be included between the query and query concept nodes. The INQUERY model allows for the combination of multiple sources of evidence (beliefs) to retrieve relevant documents.

INQUERY uses standard procedures for *stopping* and *stemming*. *Stop words* are high frequency words that do not represent content and add little value for discrimination between documents (e.g., *and, but, etc., the, a*). INQUERY removes all predefined stop words from documents and stems all the remaining words. In *stemming*, suffixes are removed to get at the root form of a word (e.g., *dwelling* becomes *dwell, nondeductible, nondeduct*). What remains in a document constitute the *terms* that are used as the (inverted) indices for the document. The same stopping and stemming procedures are used by the RF module to produce a list of terms to consider for inclusion in a query.

Full-text versions of the opinions for cases selected for inclusion in the RF-CKB are passed to a modified version of INQUERY's relevance feedback module. Relevance feedback is a widely-used method for improving retrieval. It has been found to improve precision significantly [Salton, 1989]. In relevance feedback, a user tags texts as to their relevance. In our system, we use the texts (i.e., the case opinions) associated with the cases in the RF-CKB as the set of tagged, relevant documents.

Using information derived from the texts tagged as relevant, an RF algorithm alters the weights of the terms used in the original query, and/or adds additional query terms, to produce a new, modified query. The new query is then submitted to the IR engine with the hope of retrieving more relevant documents. INQUERY's RF module uses a *selection metric* to extract a set of terms from the relevant texts. The top *n* terms or *n* pairs of terms are then weighted according to a *weighting metric*. A query consists of a set of weighted terms.

We use the selection and weighting metrics found to be best by Haines and Croft, who conducted a series of experiments using differing term selection and weighting schemes on two collections [Haines & Croft, 1993]. One of their collections, the *West* or *FSupp* collection, is very similar to the one used here since it contained full-text legal opinions. We use their recommended term selection and weighting formulas.

Ordinarily INQUERY would not engage in relevance feedback until a retrieval, based on user input, had been made and a set of documents retrieved, examined, and tagged by the user. However, since the CBR analysis already provides the system with a set of relevant documents—those associated with cases in the RF-CKB—there is no need for an initial user-provided query nor user-provided relevance judgments.

There are several variables usually manipulated in relevance feedback experiments:

1. the importance of the original query (re-weighting of the original terms),
2. the selection metric for finding terms to add from the relevant documents,
3. the weighting metric for weighting new terms,
4. the number of relevant documents to use,
5. types of terms to use: individual terms or pairs of terms, and for pairs, the window size used for finding pairs, and
6. the number of new terms (or pairs of terms) to add,

In our experiments, there is no "original query," *per se*. Instead, INQUERY's RF mechanism is given a null query and the RF-CKB as its set of relevant documents. Because there is no original query to modify, there is no need to re-weight original terms. The query consists strictly of terms or pairs of terms found within the RF-CKB. We did not vary the selection and weighting metrics.

For this paper we restrict pairs of terms to be found within a specified window or proximity. We vary the window sizes. We use the same model for pairs as used for proximity pairs in [Croft et al., 1991]. Small windows (e.g., 3) represent phrase-like proximity; large windows (e.g. 10) represent sentence-like proximity.

We did not experiment with queries that mixed terms and pairs. We did not use INQUERY's phrase operator (except for baseline queries). We did not mix pairs from windows of different sizes within a query. We also did not use INQUERY's passage operator which has also been shown to enhance performance significantly.

In summary, our experiments only vary the last three aspects of relevance feedback.

4. An Example

The following scenario illustrates our approach. Suppose a client consults with his lawyer about his attempt to take a tax deduction for an office in his home as allowed under Section 280A of the Internal Revenue Code which concerns "deductions of various expenses in connection with business use of a home, rental of vacation homes, etc." Even though the IRS has questioned it, the client believes that his deduction should be allowed under subsection 280A(c)(1).[3] He tells his lawyer various facts about his situation. The lawyer inputs the case facts into the CBR-IR system. (The facts of the problem case become slot fillers in a case frame representing a legal case.)

For example, suppose the client is Mr. Weissman of the home office deduction case *Weissman v. Comm.*, 751 F.2d 512 (2d Cir. 1984).[4] The facts of Mr. Weissman's situation are:

> David Weissman was a professor of philosophy at the City College in New York City. Although he was provided with a shared office at City College, it was not a safe place to leave teaching, writing, or research materials and equipment. So, in his 10-room apartment, Professor Weissman maintained a home office, consisting of two rooms and an adjoining bathroom. He estimated he worked between 64 and 75 hours per week but spent only 20% of that in his office at City College. The IRS challenged his deduction of $1540 of expenses related to his home office. The IRS claimed that his home office did not satisfy the statutory requirements of Section 280A(c)(1). In particular, the IRS said he failed to meet the requirements that his home office be his "principal place of business" and that its use be for the "convenience of his employer."

Suppose his lawyer is familiar with a set of home office cases from her own tax practice and that these make up the CKB used by the system. Assume the CKB contains cases from CABARET's original case base.

[3] [A deduction may be taken for] any item to the extent such item is allocable to a portion of the dwelling unit which is exclusively used on a regular basis—(A) [as] the principal place of business for any trade or business of the taxpayer, (B) as a place of business which is used by patients, clients, or customers in meeting or dealing with the taxpayer in the normal course of his trade of business, or (C) in the case of a separate structure which is not attached to the dwelling unit, in connection with the taxpayer's trade or business. In the case of an employee, the preceeding sentence shall apply only if the exclusive use referred to in the preceeding sentence is for the convenience of the employer. *I.R.C. Section 280A(c)(1).*

[4] *Weissman* is presented as an extended example in [Rissland & Skalak, 1991].

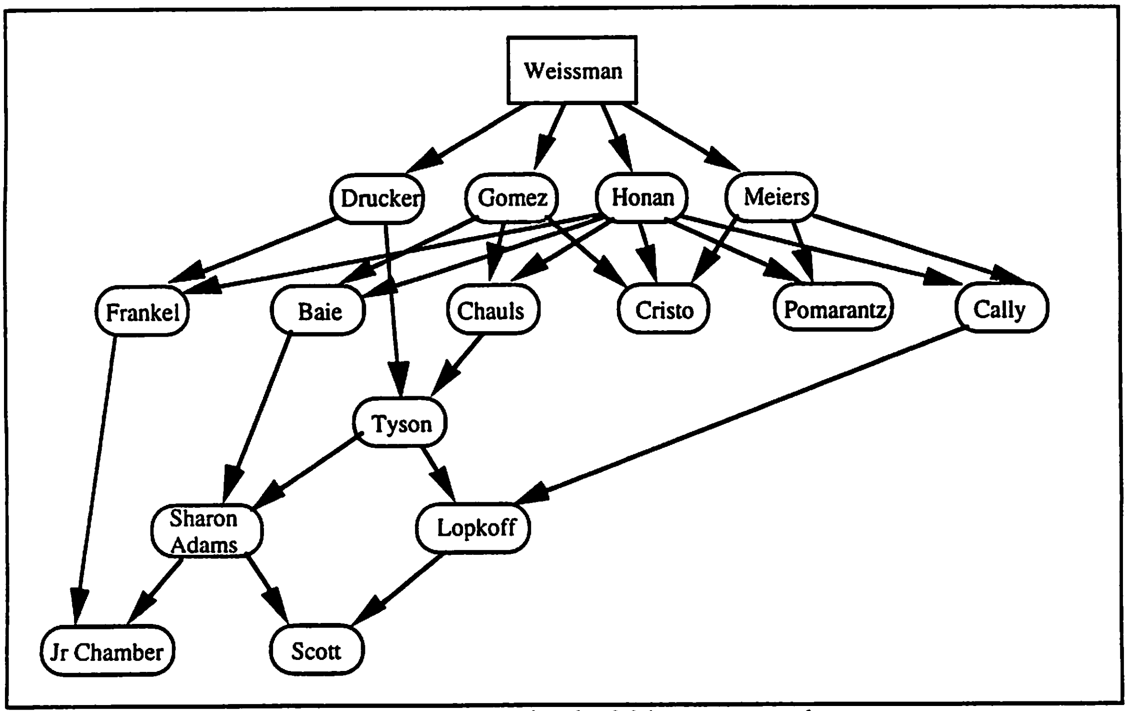

Figure 2. The claim lattice for the *Weissman* example.

Using the this CKB, the CBR module analyzes Mr. Weissman's case. Figure 2 shows the cases in the resulting claim lattice. *Drucker, Gomez, Honan,* and *Meiers* are the mopc's.[5]

The combined CBR-IR system now uses this analysis to search for additional relevant cases within a larger corpus of legal texts, say those available through the WestLaw Federal Taxation Case Law collection. To do this, the system formulates a query by employing relevance feedback on a small set of special texts—the RF-CKB—corresponding to cases selected from the claim lattice. For instance, the sets of cases in the top layer (i.e., the mopc's) or in the top two layers are good choices for the RF-CKB since they contain cases highly similar to the problem case.

The document identifiers for the texts associated with these cases are then passed to the RF module within INQUERY, which then selects and weights the top terms based on these RF-CKB texts, forms a query, and acts on it in the usual manner. Sample queries, generated using the mopc's (labeled RF-CKB1) and the top 2 layers of the claim lattice (RF-CKB6) for the Weissman example, are given in Figure 3.

[5] The claim lattice here is simpler than in that paper due to fewer cases used in the CKB. Here we also omit the list of the dimensions applicable to the cases in each lattice node as well as their dispositions.

Queries generated using the mopc's (RF-CKB1)

Query with 5 terms is:

```
#WSUM(1.000000  1.510823 baie  0.737447 c.b  2.779849  drucker
      0.696836 94-  0.766203 1976-)
```

Query with 10 terms is:

```
#WSUM(1.000000  0.696836 94-  0.737447 c.b  1.510823  baie 2.779849 drucker
      3.414727 musician  0.766203 1976-  0.383061  solo 0.459854 848-
      0.387462 moller  0.757890 k3355)
```

Query with 15 terms is:

```
#WSUM(1.000000  0.542400 rehears  1.510823  baie 0.387462  moller
      0.696836 94- 0. 766203 1976-  3.546573  focal 0.542400  opera
      3.414727 musician  2.779849 drucker  0.629428 sharon  0.737447 c.b
      0.383061 solo  0.757890 220k3355  0.459854  848- 0.757890 k3355)
```

Queries generated using the top 2 layers (RF-CKB6)

Query with 5 terms is:

```
#WSUM(1.000000  1.787807 1976- 4.196730 baie 2.438927 94-
      4.891825 focal 2.335248 c.b)
```

Query with 10 terms is:

```
#WSUM(1.000000  1.016186 938 2.438927 94- 4.196730 baie 1.610545 nondeduct
      1.787807 1976- 1.561463 rept 13.118103 280a 2.335248 c.b
      4.891825 focal 3.975086 dwell)
```

Query with 15 terms is:

```
#WSUM(1.000000  2.169601 opera  4.891825 focal  3.975086 dwell  2.438927  94-
      1.787807 1976-  1.561463 rept  1.560941  desk 13.118103 280a
      4.196730 baie 1. 368488  revd 1.016186  938 3.706465  drucker
      1.671446 curphey  2.335248 c.b  1.610545 nondeduct)
```

Figure 3. Queries generated on the *Weissman* example.

It is interesting to examine the terms in the queries and consider how they were found by the INQUERY RF mechanism and which might have been generated by our lawyer. For example, in the 15-term query generated from RF-CKB6, a term, like *280A*, is perfectly obvious. It is also not hard to see how others might have been found. For instance, *focal* is from the phrase *focal point test*, the name for a particular legal approach to the "principal place of business" requirement for home office deduction and *dwell* is the stem of *dwelling*, a term used frequently in the language of subsection 280A(c)(1), which is often quoted in its entirety in home office opinions.[6] Both *rept* and *revd* are abbreviations: the former is for *report*[7] and the latter is for *reversed* (as in "reversed on appeal," a common phrase in legal cases).

Other terms are not at all obvious, such as, *opera*, which no doubt comes from the *Drucker* case which concerned a musician in the Metropolitan Opera Orchestra. *Baie*, *Drucker*, and *Curphey* are case names.[8] Note, without even acting on the query, a new case—*Curphey* [9]—not known in the lawyer's CKB has been "discovered."[10] (Of course, in the usual scenario, the user would not ordinarily inspect the query that is generated.) A case, like *Baie* or *Drucker*, whose name is a highly valued term is likely to be among the retrieved documents that are highly rated by INQUERY. See Figure 4 for the list of the 20 top rated cases retrieved by INQUERY; the list includes *Baie* and *Drucker*.

Even an experienced user would be unlikely to use some of these terms, like *opera*, if she were to compose the query herself. Case names for cases that are not memorable or even known, like *Curphey*, would surely not be used since they are outside the ken of the user. (Presumably the user would include the most memorable cases in her CKB; *Curphey* is not in the CKB.). Of course, if a case is cited in an opinion of a CKB case, one could discover it by actually reading or "Shepardizing" the CKB case, but this would need to be done by some skilled human researcher.[11]

When the RF mechanism of INQUERY processes the terms within a case document, it is in a way "reading" the case and picking out what it considers to be high-valued terms. If a case name, like *Curphey*, occurs often enough but not so frequently as to be given a low rating, it may be selected as one of the terms used in a query. Of course, RF can also

[6] For instance, *dwelling* is used twice in 280A(c)(1), the home office subsection: "allocable to a portion of the dwelling" and "separate dwelling."
[7] As in the report of the Senate: "S. Rept. No 94-938, 1976-3 C.B. (Vol. 3) 49, 185" a citation found in *Gomez* and *Honan*, for instance. "C. B." stands for *Cumulative Bulletin*, a compendium of various tax-related reports, rulings, and memoranda.
[8] INQUERY uses lower case for all terms used in a query.
[9] *Curphey v. Commissioner*, 73 T.C. 766 (1980).
[10] Both *Baie* and *Drucker* are in the system's CKB. *Curphey* is not. *Curphey* is also not in the HOD test collection we used in our experiments.
[11] *Shepardizing* is a procedure that produces lists of all the cases cited by a given case and those that cite it. *Shepard's Citations* is published by McGraw-Hill, which continually updates the citation lists. It is available on-line.

cause some rather non-intuitive terms to be included, for instance, *C.B.*, *94-*, *938*, and *1976-*.[12] Such terms are due to vagaries in parsing and document statistics.

From our own observations, most users of INQUERY tend to use only one or two individual terms in their queries even though INQUERY allows ample natural language input. A typical user in our scenario would probably use the single term *280A* or perhaps the two terms *280A* and *dwelling* AND-ed or OR-ed together or the phrases *home office* or *home office deduction*.[13] Even expert users are somewhat unimaginative in their use of INQUERY. Naive users tend to pose queries that contain only a one or two keywords [Croft et al., 1995].

Weissman-Top 20

	Cases retrieved using mopc's (RF-CKB1)	**Cases retrieved using the top 2 layers (RF-CKB6)**
1	*Meiers* (mopc)	*Cristo* (layer 2)
2	*Drucker* (mopc)	*Meiers* (mopc)
3	*Honan* (mopc)	*Baie* (layer 2)
4	***Weissman-the real case***	***Hamacher***
5	***Dudley***	*Lopkoff* (layer 4)
6	***Soliman*-T.C.**	***Dudley***
7	***Cadwallader***	*Pomarantz* (layer 2)
8	***Soliman*-S. Ct**	*Honan* (mopc)
9	***Soliman*-F2d**	***Weissman-the real case***
10	*Pomarantz* (layer 2)	***Soliman*-T.C.**
11	*Lopkoff* (layer 4)	***Weightman***
12	***Pomarantz* -T. C. Memo**	*Drucker* (mopc)
13	***Hamacher***	***Soliman*-S.Ct.**
14	***Kisicki***	*Frankel* (layer 2)
15	***Weightman***	***Cadwallader***
16	*Baie* (layer 2)	*Cally* (layer 2)
17	*Cristo* (layer 2)	***Soliman*-F2d**
18	***Crawford***	***Williams***
19	***Williams***	***Crawford***
20	***Murphy***	***Pomarantz*-T.C. Memo**

Figure 4. The 20 most highly ranked documents returned for the *Weissman* problem case using the mopc's (RF-CKB1) and top 2 layers (RF-CKB6) to generate queries of 150 terms. Cases given in **boldface** are not present in the CBR module's CKB. Parenthetical remarks on non-bolded cases indicate their position in the claim lattice.

Finally, our system returns those texts retrieved with the system-generated query. Figure 4 shows the 20 most highly rated cases returned by INQUERY on the *Weissman*

[12] See footnote 7 above.

[13] As it turns out, these queries produce results with fairly high average precision: 81.1% for *280A* alone and 77.1% for *280A* and *dwelling* when *and*-ed or *or*-ed together. Baselines are discussed in Section 7.3.

problem for queries with 150 terms. All are home office deduction cases. Thus, in so far as "precision" in the top 20 is concerned, our approach achieves 100%.

The cases returned include a variety of home office deduction cases, including cases, like *Drucker* and *Baie*, from the top 2 layers, which the lawyer already knew about, and new cases like *Dudley*, which she didn't. Note that the real *Weissman* case, *Weissman v. Comm.*, 751 F.2d 512 (2d Cir. 1984), is retrieved—a reassuring sign—and that it is highly ranked.[14] Some case like *Pomarantz* and *Soliman* appear more than once since they have opinions from different forums, such as the Tax Court (cited with "T.C.") or a court of appeals (cited with "F2d"). The mopc *Gomez* just missed the top 20: it ranked 22 with RF-CKB1 and 31 with RF-CKB6.

In particular, the *Dudley* case—*Dudley v. C.I.R.*, T.C. Memo 1987-607[15]—is the highest ranked case, after the real *Weissman* case, on the list generated with RF-CKB1 that is not known in the CKB. It is also highly rated when RF-CKB6 is used. It is, as lawyers say, "on all fours" with Mr. Weissman's problem:

> Mr. and Mrs. Dudley are both college professors. Neville Dudley was a full-time business professor at the downtown campus of the Wayne County Community College. Gloria Dudley was a part-time professor. Most of the expenses claimed concerned Mr. Dudley. Mr. Dudley taught a variety of business courses. In addition to classroom instruction, his duties included writing syllabi for his various courses, evaluating textbooks, reading professional publications, preparing notes for class, preparing, correcting, and grading examinations. The College did not require him to do research. Nor did it require him to maintain a home office. Even though his contract with the College specified that the College was to provide each full-time faculty member with a private office, it did not provide Mr. Dudley, or any other faculty member, with one. Rather it provided an on-campus room that had 20 work places, each with a writing shelf but no locked storage space. Mr. Dudley used a room in his home for his work-related activities. There, he typically spent the morning hours from 8 a.m to 11 a.m. reading and preparing for class. In the evenings when he did not teach, he spent an additional 2 hours reading, writing examinations. He also used the home office for work-related tasks on weekends.

From our Mr. Weissman's point of view, *Dudley* is a contrary case since the Tax Court found that the focal point of Mr. Dudley's business activities, that is, his principal place of business, was where he taught. It also held that Mr. Dudley failed to show that his

[14] In our example, the real *Weissman* case was deleted from our CKB so it could be run *de novo* as a problem case.

[15] There is also an appeals version *Dudley v. C.I.R.*, 860 F.2d 1078 that affirmed the lower Tax Court result without issuing an opinion. The absence of an opinion means that it could not be included in the HOD document collection that we created. (See below, Section 5.3) Cases without opinion need to be handled in some way that makes them amenable to full-text retrieval methods. This is related to the problem of using the title of a document in retrieval [Croft et al, 1995].

use of his home office was for the convenience of his employer. The tax court opinion cited the *Curphey* case for the proposition that a taxpayer can have only one principal place of business. It distinguished *Drucker*, and *Meiers*, and the real *Weissman* case, an appeals case, on which the Dudleys relied for support of their position. It distinguished *Dudley* from *Weissman* by saying that Professor Dudley did not show that research and writing were required by his college, whereas they were for Professor Weissman, and that Weissman spent time in his home office doing research and writing, and Dudley didn't. The tax court thus cut a fine line between the professorial duties of Weissman and Dudley.

Of course, in our example, Weissman is being treated *de novo*. This means that we can think of our Mr. Weissman's problem as being either the real (1984) *Weissman* case, in which case, we should filter out any cases, like the 1987 *Dudley* case, which occurs after Weissman, from our example (i.e., in the document collection and system output), or as a new case isomorphic to the real *Weissman* case, and thus, coming in time after both *Dudley* and the real *Weissman* case. However one sets up the time line of our example, *Dudley* and *Weissman* are intimately related. And in the later interpretation, Mr. Weissman's lawyer would need to think carefully about *Dudley*.

The *Cadwallader* case—*Cadwallader v. C.I.R.*, T. C. Memo 1989-356—is also right on point:

> The taxpayer, a professor, used his home office for research and for storage of research materials that didn't fit in the campus office or storage areas that his employer, Indiana State University, provided for him. Research was an important part of his job and these materials were important to his research. These materials, which represented a great wealth of information, had been accumulated by Cadwallader over the years; they were not available in his university's library.

In this 1989 case, the tax court denied Cadwallader's deduction on the grounds that his situation also failed to meet the focal point test, "This Court consistently has held that the focal point of a college professor's activities is the campus, campus office, or classroom, rather than the home office." It cited a variety of cases including *Baie*, *Williams*, and *Cristo*, which are included on our list of highly ranked retrieved cases.

Note, such cases concerning professors were retrieved coincidentally. None of the cases used in the RF-CKB's to generate queries were about professors. For instance, none of the terms in the queries shown in Figure 3 are *professor*.

Perhaps the most significant cases on the returned list are the *Soliman* cases. There are three versions of *Soliman*, each representing the case at a different step on the appellate ladder: Tax Court in 1990, Court of Appeals for the Fourth Circuit in 1991, and Supreme Court in 1993.[16] *Soliman* is the only home office deduction case that has been decided in the Supreme Court. It speaks to the use of the focal point test as part of two-prong facts-and-circumstances approach. Both for its pedigree[17] and holding, this case is exceedingly important to home office deduction cases, in general, and Mr. Weissman's case, in particular. Its importance is not diminished by the fact that Soliman is not a professor but an anesthesiologist, who used his home office for maintaining records and billing clients, since the hospitals where he worked provided him no office space. It presents Mr. Weissman with a high precedential hurdle to clear. Weissman will have a hard time passing the focal point prong, although he is probably in a pretty good stance vis-a-vis the relative time prong since he spent 80% of his effort in his home office. His lawyer will need to study *Soliman* closely as well as the appellate contrary cases (*Dudley* and *Cadwallader*) and supportive cases (*Drucker, Meiers, Pomarantz* and real *Weissman*).

It is important to note that *Soliman,* in all its variations, was found automatically without performing legal research (e.g., Shepardizing). It shows how our approach can retrieve cases decided after the system's CKB was last updated.[18] Thus, our approach successfully copes with what could be called the "staleness" problem for case bases.

The system returns to the lawyer a ranked list of documents, which she can then download for her research on Mr. Weissman's problem. The CBR-IR approach has:

1. found cases that are factually highly similar (e.g., *Dudley, Cadwallader*),
2. located new cases unknown to the CBR module (e.g., *Dudley*), and
3. found important cases decided since the CKB was built (e.g., *Soliman*).

Of course, she, herself, has to read and analyze them. However, without any need for formulating queries or cleverly manipulating the retrieval engine directly, she has been

[16] *Soliman v. C.I.R.*, 94 T.C. No. 3 (1990), *Soliman v. C.I.R.*, 935 F.2d 52 (4th Cir. 1991), *C.I.R. v. Soliman*, 113 S. Ct. 701 (1993). In an audit, the Commissioner of the Internal Revenue (C.I.R.) denied Soliman's home office deduction, and Soliman appealed to the Tax Court, which overruled the Commissioner, who then appealed to the Court of Appeals for the 4th Circuit, who affirmed the decision of the Tax Court. The Commissioner then appealed that outcome and won a reversal in the Supreme Court. However, the Supreme Court held that the focal point test was not the sole means to determine 280A(c)(1) deductibility issues. Although the opinion written by J. Kennedy employed a "facts and circumstances" approach, the focal point test remained one of two key considerations; the other was the relative time spent in the home office.

[17] INQUERY takes no notice of court pedigree in ranking retrieved cases. A system specialized for legal applications probably should.

[18] When the CABARET CKB was built in 1989, the *Soliman* case had not even begun is judicial journey.

able to access a massive on-line document collection in a problem-based manner and discover relevant cases she might have missed otherwise.

5. Domains, CKB's, Problem Cases, and Text Collections

In this section we provide background on the experiment domains, selection of the RF-CKB's, building of the collections, and the creation of the relevance files or "answer" keys.

5.1 Problem Domains and CKB's

We have experimented with our approach in two domains thus far:

1. the *home office deduction (HOD) domain*, the domain used by CABARET [Rissland & Skalak, 1991],

2. the *good faith bankruptcy domain*, used by BankXX [Rissland et al., 1994a, b].

CABARET's original case base consisted of 36 real and hypothetical cases concerning the home office deduction, whose requirements are given in Section 280A(c)(1) of the Internal Revenue Code. For this project, we used 25 real cases from the CABARET case base.

BankXX's original case base consisted of 55 cases concerning the *good faith* issue for the approval of debtor plans under Chapter 13 of the Bankruptcy Code, specifically in Section 1325(a)(3). For this project, we used the 45 decided after 1981.

In each domain, we have run a series of experiments by submitting problem cases, chosen from one of these two case bases. When a case is used as a problem case, the system treats it in a *de novo* manner by temporarily deleting it from the CKB and analyzing it as though never before seen by the system. The rest of the cases in the CKB become the cases against which it is analyzed. For instance, when we run a problem case from the home office deduction domain, it is run against a CKB containing the 24 remaining cases.

5.2 Problem cases

So far we have run experiments with 4 home office deduction cases and 3 bankruptcy cases as problem cases. They are:

A. from the home office deduction domain:
1. *Weissman v. Comm.,* 751 F.2d 512 (2d Cir. 1984)
2. *Honan v. Comm.,* T.C. Memo. 1984-253
3. *Meiers v. Comm.,* 782 F.2d 75 (7th Cir. 1986)
4. *Soliman v. Comm.,* 935 F.2d 52 (4th Cir. 1991)

B. from the bankruptcy domain:
1. *In re Easley,* 72 B. R. 948 (Bkrtcy. M. D. Tenn. 1987)
2. *In re Makarchuk,* 76 B. R. 919 (Bkrtcy. N. D. N. Y. 1987)
3. *In re Rasmussen,* 888 F.2d 703 (10th Cir. 1989)

5.3 Building the Document Collections

To test our approach, we constructed two document collections:

1. in the home office deduction domain, the test corpus, called the *HOD-corpus,* consists of over 12,000 legal case texts from a variety of legal areas;

2. in the bankruptcy domain, the test corpus, called the *Bankruptcy-corpus,* consists of over 950 legal texts addressing the issue of approval of a debtor's plan, as specified in Section 1325(a).

Both text collections were built by downloading the full-text of opinions for the cases in the CKB's and then preparing them for use by INQUERY. For instance, we had to assign document id's and add SGML tags.

The HOD-corpus contains cases addressing a great many legal questions. It was built by adding approximately 200 cases to another already existing, nearly 12,000 document collection, called the *West* or *FSupp* collection [Haines & Croft, 1993; Turtle, 1994]. The additional texts came from the cases found in the CABARET CKB and those found when the natural language query *home office* was posed to West's WIN system against the WestLaw Federal Taxation Case Law database. We restricted the query cases to be between January 1986[19] and November 1993 and removed all redundant cases. After

[19] A major revision was made to the tax law in 1986. It included 280A(c)(1) as a new provision. The scope of this new section, in particular the meaning of ingredient terms, such as *principal place of business,* were

accounting for the 25 cases from CABARET's CKB, this resulted in adding in 103 new HOD cases.

The new collection contains 12,172 texts, of which, 128 cases discuss taking the home office deduction.[20] Therefore, only about 1% of the cases in the HOD-corpus address the home office deduction (280A(c)(1)) issue we are interested in. The HOD collection is fairly heterogeneous. Using the query *280A* with INQUERY 1.5.6, we achieve an average precision of 81.1%. Other possible queries resulted in similar baselines. (See Table 3 in Section 7.3)

The Bankruptcy-corpus contains cases dealing only with the specific issue of debtor plan approval, as given in Section 1325(a). We built this corpus by downloading all the cases between 1982 and 1990 that were found with the WIN query *1325(a)* posed to the WestLaw Federal Bankruptcy Case Law database. It contained all but the 10 earliest cases from the original 55-case BankXX CKB. In the Bankruptcy-corpus about 40% (385 cases) make specific reference to the narrower "good faith" issue of subsection 1325(a)(3).[21] Thus, this corpus is very focused. Using INQUERY 2.0 on the simple one phrase query *good faith* against this corpus results in an average precision of 89.3%; this extremely high value indicates that a high proportion of good faith cases actually use that phrase and that cases on other issues do not. This corpus is exceedingly homogenous.

Even though the bankruptcy corpus is much more homogeneous than collections ordinarily used in IR experiments, we felt it was realistic for a statutory problem domain. We believe that a typical lawyer would first retrieve such a collection, using the statutory citation as a query, and then within this collection look for truly relevant cases. As we shall discuss, it is hard surpass the results achieved with obvious baseline queries in this corpus when the results are judged by a very forgiving definition of relevance (i.e., a case is relevant if it is about the 1325(a)(3) issue). A tightly defined text collection really requires a more attenuated sense of relevance, for instance, that based on what cases are relevant to a particular problem situation. (See discussion below.)

undefined in 1986 since they were not defined elsewhere in the statute and there were no cases addressing them. The CABARET CKB contains several pre-1986 cases.

[20] We examined the more than 200 new cases by hand to determine this.

[21] We determined this by checking the cases retrieved by the WIN query *1325(a)(3)*.

Home office deduction cases often discuss more than just the home office deduction (280A(c)(1)) issue. In fact we found that as many as seven or more other issues might be covered within such a case. Such cases are *impure* in the sense that they discuss the home office deduction and one or more other issues. By contrast, a *pure* case addresses no other issues. Of the 25 cases in the CBR module's CKB case base, 18 cases are pure. Within the other 103 home office deduction cases from the entire HOD-corpus, fewer than 10 were pure. On average, pure texts are much smaller than typical texts in the *FSupp* collection.

On the other hand, we found that most of our bankruptcy cases were pure, that is, they only addressed the one "good faith" (1325(a)(3)) issue. Not surprisingly, the home office deduction cases vary significantly in length—anywhere from one to 20 or more pages in length—whereas the bankruptcy cases tend to be on the shorter side, running generally less than 10 pages.

Not surprisingly, pure cases provide the least number of unique terms for the RF mechanism to select among and they tended to be shorter documents. Impure documents had much larger numbers of unique terms and were significantly longer. The average *FSupp* document was as long as the average impure document, yet had significantly fewer unique terms. (See Table 1 below.)

5.4 Answer Keys

For each problem case, we constructed an answer key that specified the documents to be considered as relevant. There are two sense of relevance that we have used:

1. a general correct answer, in which a case must simply address the issue, that is, the 280A(c)(1) home office deduction issue or the 1325(a)(3) good faith issue.

2. a problem-specific answer:
   (a) cases actually cited in the case opinion for the problem case
   (b) cases judged relevant (by us) to the problem case.

The first sense is a very broad sense of relevance. For instance, any of the 128 cases from the HOD-corpus that actually concerns a taxpayer trying to take the deduction is considered relevant to a problem case. Furthermore, all problem cases are assigned the same set of texts as the correct answer if this sense of relevance is used. The answer includes those which CABARET would have considered on-point.

The problem-specific senses of relevance are narrower. To apply sense 2(a), we look at the court opinions to see what cases were actually cited—and thus considered relevant—by the court. For 2(b), we examine the output of our system (e.g., the top 20 cases) across a given level of queries (e.g., 150 terms) to determine which retrieved cases are relevant to the problem case. In effect, this addresses precision among a subset of the documents; it does not address recall. Note, we do not try to construct an answer key from scratch, for instance, by performing legal research *de novo* on a problem case.

Note, it is by examination of retrieved cases that we were able to see that *Dudley*, *Cadwallader*, and *Soliman* were retrieved in our example, based on *Weissman*. This sort of analysis is very time-consuming, but it provides insights not available through the usual aggregate statistics of average precision.

Sense 2(a) is highly objective: it is defined by a court. In the bankruptcy domain, we are able to use this definition of relevance because we already have hand-coded answers for individual problems available from our empirical evaluation of the BankXX system [Rissland et al., 1995] in which we compared the sets of items (e.g., cases, legal theories) retrieved by BankXX against those actually mentioned in a case. Creation of this set of answers had been a laborious task. We do not have a complete set of such answer keys, at this time, for the HOD domain.

In summary, Sense 2(a) is highly focused but expensive to apply, and Sense 1 is very broad, and easy to apply. Sense 2(b) is somewhat intermediate. The narrower sense is particularly important in evaluating results using our very homogeneous bankruptcy corpus. In this paper, we concentrate on the first, broad sense of relevance, and results using the HOD collection. Problem-specific relevance is the subject of on-going work.

6. RF-CKB's—Cases for Seeding Relevance Feedback

On the *Weissman* case, which was the first with which we experimented, we used 6 different RF-CKB's:

1. **RF-CKB1** consists solely of the set of mopc's. For the *Weissman* fact situation, there are 4 such cases. Coincidentally, this set of 4 mopc's happens to be pure. RF-CKB1 can also be thought of as comprised of cases from solely the top layer.

2. **RF-CKB2** consists of only impure cases; a random selection of 5 chosen from the *Weissman* claim lattice. RF-CKB2 tests the ability of relevance feedback to discriminate important terms from non-relevant ones within noisy texts.

3. **RF-CKB3** is the union of RF-CKB1 and RF-CKB2 and so has both pure and impure texts. RF-CKB3 has the advantage of having a large number terms from which to select the important ones.

4. **RF-CKB4** contains all the pure texts from the top two layers of the *Weissman* claim lattice. It contains 8 texts.

5. **RF-CKB5** includes all the impure texts in the CBR module's CKB of 25 cases. There are 7 such cases.

6. **RF-CKB6** uses all the cases in the top 2 layers of the claim lattice. It contains 11 cases, of which 8 are pure (i.e., RF-CKB4) and 3 impure. Since it includes the top two layers, it contains RF-CKB1.

After conducting experiments with these RF-CKB's and the *Weissman* case, we narrowed our focus. For further experiments in both domains, we only used RF-CKB1 and RF-CKB6 as they related to the new problem case.

	West	RF-CKB1	RF-CKB2	RF-CKB3	RF-CKB4	RF-CKB5	RF-CKB6
Documents	11953	4 pure	5 impure	9 mixed	8 pure	7 impure	11 mixed
Unique Terms	142749	1242	2430	2885	1952	2941	2767
Unique Terms perText (avg)	530	477	842	680	516	834	589
Total Terms per Text (avg)	3250	1254	3321	2402	1533	3353	2031

Table 1. RF-CKB statistics for the Home Office Deduction experiments with the *Weissman* case.

7. Overview of Experiments

In our experiments, we varied:

1. the problem case,
2. the RF-CKB's used to seed the query,
3. the types of terms (individual terms, pairs) composing a query,
4. the number of terms to use in the query,
5. for pairs, the window size used to delineate pairs.

In general, we performed the most variations of these parameters on the *Weissman* case. Then based on the results from *Weissman,* we narrowed our choices for the other half-dozen problem cases (listed in Section 5.2).

We initially tested queries composed of 5–400 individual terms generated from each of the six RF-CKB's described above (Section 6) with the *Weissman* case as the problem case. Results from the term experiments on the *Weissman* case are shown below in Table 5 in Section 8.1. For each of the original six RF-CKB's and the *Weissman* problem case we also experimented with 5–40 pairs of co-occurring terms. Results from pair experiments on *Weissman* are given below in Tables 7 and 8 in Section 8.2.

7.1 RF-CKB's

Based on the results from this initial set of experiments, we tried other problem cases. However, we only used RF-CKB1 and RF-CKB6 to select our texts for the relevance feedback module. In addition, we also used the problem case, by itself, as a RF-CKB. This provides another comparison point but obviously is not what would usually be done in practice since at the time a user is searching for relevant documents for a problem case, the case presumably has not been decided and an opinion written.

Thus, the RF-CKB's used in all problem cases were:

1. **RF-CKB1**—the top layer of the claim lattice for the problem case (i.e., the mopc's).
2. **RF-CKB6**—the top two layers of that claim lattice for the problem case.
3. **RF-CKB-cfs**—the problem case, itself.

RF-CKB-cfs provides a baseline against which to assess the performance of RF-CKB1 and RF-CKB6. (See Table 4 below.)

7.2 Terms and Pairs

For the term experiments in each RF-CKB that we experimented with, the relevance feedback module selected, weighted, and formed a query composed of the top 5, 10, 15, 20, 25, 50, 100, 150, 200, 250, 300, 350, and 400 terms found in the RF-CKB. The maximum length query was 400 terms due to a limitation within the relevance feedback module. Therefore, longer queries, such as all of the terms from within a RF-CKB, were not tested.

For the pair experiments, we ran queries with 5-40 pairs of terms within window sizes of 3 to 10. In *Weissman*, we tried larger window sizes but they yielded worse results and were computationally much more expensive. Pairs with window size 3 represent a phrase-level separation. Pairs with window size 10 represent a sentence-level separation.

7.3 Baselines

In these experiments, we compared our CBR-IR results against various baselines. Baselines are expressed in average precision scores that are calculated from recall and precision statistics:

- ***Recall*** measures the percent of those items that should have been retrieved by the query that actually were. It measures coverage. It is the ratio of the number of relevant retrieved items (i.e., items in the intersection of the answer key and the retrieved items) to the total number of relevant items.

- ***Precision*** measures the percent of retrieved items that are relevant. It measures accuracy. It is the ratio of the number of relevant retrieved items to the total number of retrieved items.

- ***Average precision*** is the average of the precision scores achieved at 11 levels of recall: 0%, 10%, 20%, ... 100%.

Since we know what the correct answer is, we can determine when a given level of recall is achieved by the system and then calculate the precision at this level. For instance, we can determine when 10% of the relevant texts have been retrieved by the system, and use the retrieved texts to calculate precision at the 10% level. When we use 11 levels of recall, it is called *11-point* average precision. Table 2 shows typical data. Notice that there is a big drop-off in precision at 80% recall; it is hard to retrieve the last 10% or 20% of the relevant documents.

level of recall	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%	**AvgPrecision**
precision	100%	99.0%	99.0%	99.0%	99.0%	99.0%	99.0%	99.0%	96.4%	1.1%	1.1%	**81.1%**

Table 2. Data for 11-point average precision calculation for the query *280A*. Average precision is 81.1%.

Baselines represent results that can be achieved with the use of IR alone. See Table 3. We established some obvious baselines:

1. problem-independent hand-crafted queries of simple terms or a pair of terms:
   a. for the HOD-domain, we used queries composed of:
      i. *280A*
      ii. *280A* AND *dwelling*
      iii. *280A* OR *dwelling*
      iv. weighted sum of *280A* and *dwelling* with 3 relative weightings
      v. the phrase: *home office*
   b. for the Bankruptcy domain, we used:
      i. the phrase: *good faith*
2. the entire full-text opinion of the problem case as the query.
3. a 1-3 paragraph summary of the problem case as the query.

Table 3a. problem-independent baselines

Query	Avg Precision
280A	81.1%
280A AND dwell	77.1%
280A OR dwell	77.1%
WSUM(1.0 280A 1.0 dwell)	77.1%
WSUM(5.0 280A 1.0 dwell)	81.1%
WSUM(10.0 280A 1.0 dwell)	81.1%
phrase(home office)	71.6%
phrase(good faith)	89.3%

Table 3b. problem-dependent baselines

	Average Precision			
Query	*Weissman*	*Honan*	*Meiers*	*Soliman*
case summary	36.1%	41.3%	53.8%	33.0%
case opinion	60.2%	69.6%	79.1%	60.3%

Table 3. Baseline results (11-point average precision) for various queries on the HOD-collection using INQUERY 1.5.6. Only the case summary and case opinion are problem-specific queries.

It should be noted that INQUERY usually achieves better results with phrases than terms. This is shown quite strongly in our baselines in Table 3a. For instance, 89.3% achieved by the phrase *good faith* is exceedingly high. Note, the maximum of the average precision scores for the HOD baselines is 81.1%. Many of the scores are significantly lower. An RF-CKB composed the problem case, itself, also serves as a baseline (Table 4.)

RF-CKB-cfs-baselines

Number of Terms	*Weissman*	*Honan*	*Meiers*	*Soliman*
5	10.1	10.1	10.1	10.1
10	10.1	10.1	14.7	10.1
15	10.1	10.1	16.0	10.1
20	13.3	10.1	20.8	13.8
25	13.1	10.1	24.0	13.2
50	28.1	36.8	35.1	15.8
100	40.9	46.0	39.6	42.5
150	38.2	69.6	62.5	47.5
200	55.4	69.5	66.9	47.8
250	42.7	63.2	65.0	43.8
300	43.4	62.2	64.2	**81.9**
350	42.1	60.2	62.4	78.0
400	42.5	61.2	68.1	73.9

Table 4. Baseline results on RF-CKB-cfs, the RF-CKB containing only the problem case, with 11-point average precision using INQUERY 1.5.6.

8. Results

8.1 Terms

We generated 11-point precision and recall tables for each of the queries associated with the *Weissman* problem case. Table 5 gives results for the six RF-CKB's on the *Weissman* case with different numbers of terms used to form a query.

Terms-Weissman

Number	RF-CKB1	RF-CKB2	RF-CKB3	RF-CKB4	RF-CKB5	RF-CKB6
of Terms	Mopc/Pure	5 Impure	9 Mixed	8 Pure	7 Impure	Top 2 Layers
5	40.6	55.2	**83.8**	39.5	53.1	39.9
10	38.6	54.0	**86.7**	42.5	63.8	**83.8**
15	36.3	**88.1**	**86.5**	**83.0**	66.8	**83.7**
20	79.3	**90.7**	**86.3**	**83.1**	68.4	**85.3**
25	79.0	**87.6**	**88.8**	**83.8**	68.1	**89.0**
50	78.9	**87.5**	**89.3**	**88.1**	**85.7**	**89.0**
100	**81.2**	**87.5**	**88.5**	**88.5**	**83.5**	**90.3**
150	**85.9**	**87.5**	**88.4**	**89.0**	**83.5**	**90.2**
200	**86.6**	**88.2**	**88.4**	**88.9**	**83.5**	**90.2**
250	**87.4**	**86.5**	**88.3**	**89.2**	**83.6**	**90.5**
300	**87.6**	**86.5**	**89.2**	**89.2**	**82.0**	**90.2**
350	**86.4**	**86.0**	**89.1**	**88.5**	80.7	**89.8**
400	**85.4**	**85.4**	**88.8**	**88.8**	**81.9**	**89.3**

Table 5. For the top *n* terms, 11-point average precision achieved by various RF-CKB's on *Weissman*. **Boldface** indicates that the query exceeded the baseline of 81.1%. Shaded cells are maxima within the column. All results are based on using INQUERY 1.5.6, the HOD-CKB, and HOD-collection.

RF-CKB1, composed of mopc/pure texts, takes the longest to find a set of terms and weights. It is not until there are between 51 and 100 terms that a query achieves an average precision exceeding the baseline of 81.1%. RF-CKB3 and RF-CKB6 achieve this quickly, with 10 or fewer terms. RF-CKB2 and RF-CKB4 take 15. Overall, RF-CKB6 achieves the best set of average precisions, and RF-CKB5 the worst.

In *Weissman,* every RF-CKB results in improvement over all the baselines—including the highest 81.1%—by the time 100 or fewer terms have been included. Significant improvements are achieved in most cases and in many cases the relative improvement is nearly 10%. Thus, the hybrid CBR-IR method significantly out-scores straight IR alone on *Weissman.*

There is a large jump in the average precisions for most of the RF-CKB's. For example, within RF-CKB6, a jump from 39.9% to 83.8% occurs between 5 and 10 terms. This may be explained by examining the set of terms that are added to the longer queries. It turns out that whenever the jump occurs, both *280A* and *dwell* are new terms. No such large jump is apparent with RF-CKB3 since it starts out high; both terms are used in queries with 5 or more terms. Note that a query composed of just the terms *280A* and *dwell* only achieves 81.1%. (See Table 3a.)

We had expected that RF-CKB1, composed of mopc's, would perform the best, and were somewhat surprised at the very strong performance of other RF-CKB's, particularly RF-CKB6. This may be because RF-CKB1 (1) has a limited number of smaller documents (see Table 1) available from which to draw terms and judge importance, and (2) is pure. By contrast, RF-CKB6 is larger (nearly three times so), has larger documents (on the order of twice as large), and contains a mix of pure and impure cases.

In *Weissman,* RF-CKB1 may be especially handicapped by the purity of its texts. Since these texts discuss only one issue, they do contain many terms descriptive of the home office deduction. Yet, because so many of high-value terms occur across all four documents, they are hard to discriminate and are undervalued by the RF mechanism.

Discriminating high-value terms might be more easily done in larger and/or non-pure RF-CKB's, such as RF-CKB2 and RF-CKB3, since the terms descriptive of the home office deduction comprise a smaller proportion of each text within an impure RF-CKB because additional issues are represented. This may aid the selection metric in finding the terms descriptive of the home office deduction. Within a mixed RF-CKB, the impure documents may provide the "noise" necessary for high-value terms to be more recognizable. This means, in fact, that the query to the IR system is: *find me cases that look like this* where for INQUERY this means *find me cases that have high value terms with respect to the given RF-CKB.*

It is also noteworthy that many of RF-CKB's have more than one peak in their curve. For example, RF-CKB2 has peaks at 20 and 200 terms. (See Table 5.) However, all the top scores are closely bunched in range. For four the RF-CKB's, the maximum scores are achieved between 50 or 250 terms.

It is unexpected that there should be multiple peaks; if the selection metric finds the most descriptive terms, in order, and these terms are appropriately weighted in the resulting query, then there would be a single peak when there were sufficient terms to adequately describe the concepts involved. Expanding the query with additional terms would just produce noise and one would expect the average precision to begin declining as more noise were added. Therefore, multiple peaks might indicate that some of the more descriptive terms are not being as highly ranked by RF in the set of all terms as they could be. Further, although they might not be selected until later, weights might compensate for the addition of these terms, as well as other, less descriptive terms.

Results for the other HOD-cases *Honan, Meiers,* and *Soliman* are similar to those for *Weissman* case. See Table 6.

Terms-HOD cases

	Weissman			*Honan*			*Meiers*			*Soliman*		
Terms	RF-CKB1 N=4	RF-CKB6 N=11		RF-CKB1 N=4	RF-CKB6 N=12		RF-CKB1 N=4	RF-CKB6 N=11		Root node N=1	RF-CKB1 N=6	RF-CKB6 N=13
5	40.6	39.9		30.9	5.2		31.5	36.9		10.1	38.6	33.9
10	38.6	**83.8**		30.9	**81.3**		31.5	**83.7**		10.1	35.2	45.7
15	36.3	**83.7**		30.9	75.5		31.5	**83.9**		10.1	38.0	**85.6**
20	79.3	**85.3**		30.9	76.9		37.7	**85.8**		13.3	38.0	**85.0**
25	79.0	**89.0**		30.2	79.1		37.7	**85.8**		13.1	77.0	**85.0**
50	78.9	**89.0**		29.8	**86.5**		**84.8**	**87.3**		28.1	78.2	**88.1**
100	**81.2**	**90.3**		30.1	**87.6**		**84.8**	**88.2**		40.9	**83.8**	**89.4**
150	**85.9**	**90.2**		71.7	**89.2**		**85.7**	**88.2**		38.2	**83.7**	**89.8**
200	**86.6**	**90.2**		69.5	**89.0**		74.3	**89.6**		55.4	**82.5**	**89.9**
250	**87.4**	**90.5**		35.6	**89.0**		77.6	**89.3**		42.7	**83.2**	**90.5**
300	**87.6**	**90.2**		40.7	**88.7**		76.6	**89.1**		43.4	**82.0**	**90.1**
350	**86.4**	**89.8**		43.5	**88.3**		73.4	**89.2**		42.1	79.7	**89.7**
400	**85.4**	**89.3**		49.6	**88.2**		71.7	**88.7**		42.5	78.5	**88.6**

Table 6. Results with 11-point average precision using RF-CKB1 and RF-CKB6 with INQUERY version 1.5.6 on home office deduction problem cases. Boldface indicates the score exceeds the baseline of 81.1%. Shading indicates the cell is a maximum within the column. The N gives the number of cases in the RF-CKB.

For *Soliman*, we also ran an RF-CKB consisting of the one case, *Weissman*, that was in the root node (since its set of applicable dimensions exactly matched *Soliman's*). It is exactly the same as the RF-CKB-cfs for *Weissman* (see Table 4 above). RF-CKB1 thus contains *Weissman* and the rest of the mopc's; RF-CKB6 includes *Weissman* and the rest of the top two layers.

As with *Weissman*, with *Meiers* and *Soliman*, the system exceeded the baseline with the mopc RF-CKB's, although not in a sustained manner. In *Weissman*, *Meiers*, and *Soliman* the scores for RF-CKB6 are all similar: high scores are achieved early and are sustained. RF-CKB1 in *Honan* does not exceed the baseline, and RF-CKB6 is a little slow.

Thus, in all four HOD problems, RF-CKB6 is always better overall than RF-CKB1. With the top-2-layer RF-CKB6's, the baseline is exceeded quickly—with 15 or fewer terms. The high scores are sustained over a large range of queries. The absolute maxima for all the RF-CKB's occurs in the 100-300 term range. Generally the top scores are closely bunched.

We feel that RF-CKB6 does so well because the top two layers in claim lattices combine several important considerations: (1) they contain the most and next-most highly relevant cases; (2) they usually contain a mix of both pure and impure cases; (3) they usually contain about twice as many cases as RF-CKB1.

In summary, for all four HOD problem cases, using the top-2-layer RF-CKB6, the system exceeds the baseline within 15 or fewer terms and achieves better overall results than with the mopc RF-CKB1. Using RF-CKB6, the system exceeds the baseline by as much as 11.7% (relatively). With the top-2-layer RF-CKB6's, the maximum scores were achieved in the 100-300 term range, and all scores for queries using 50 or more terms were very close together. While the RF-CKB1 did outscore the baseline, it was not sustained.

In the HOD term experiments, RF-CKB1 and RF-CKB6 results in improvement over all the baselines—including the highest 81.1%—usually by the time 50 or fewer terms have been included. With RF-CKB6, improvements are achieved early and significant improvements (10% or better) are sustained. Thus, the hybrid CBR-IR method significantly out-scores straight IR alone in the HOD domain.

Thus, a preliminary recommendation for practice is that using a top-2-layer RF-CKB with a good number of terms (e.g., 150) is very effective. Note using this many terms requires no added effort on the part of the user and little added cost for INQUERY.

Within the bankruptcy domain we also experimented with three problem cases. At this point, the bankruptcy term results do not appear to be as strong. The CBR-IR system achieved average precisions ranging in the high 50-low 60%'s, which is below the baseline. Of course, the only baseline we use currently is a phrase, and phrases do better in INQUERY than individual terms.

However, two of our general conclusions still stand:

- Better average precision occurs with higher numbers of terms (150 to 400)
- Better results are achieved with RF-CKB6 than with RF-CKB1.

It should be noted that the total number of documents used by the relevance feedback module is very small; the largest RF-CKB only contained 9 documents. (In the HOD domain, the maximum was 13.) Furthermore, the sense of relevance is very broad, and both the CKB and the text collection are very homogeneous. These latter characteristics present extreme conditions, indeed, for relevance feedback. What is needed in this domain—given the homogeneity of the CKB and the document collection—is a more restricted sense of relevance in the answer keys. This is the focus of current work.

8.2 Pairs

In a second set of experiments, we investigated generating queries composed of pairs of terms. The pairs selection algorithm was initially designed for use with large sets of relevant documents. Because of the large numbers of pairs found in each document, it became memory intensive. Therefore, the code was rewritten to only keep track of a pair after it had been found at least four times within a single text. If a pair did not exceed this threshold, it was discarded. Thus, the algorithm is sensitive to the ordering of the documents. For our application, this restriction severely hampers our ability to find good pairs, since the relevance feedback module only uses a small number of texts. The algorithm will be altered in future experiments to remove this ordering sensitivity.

For both domains and the vast majority of the RF-CKB's, the queries composed of pairs of terms scored higher than queries composed of single terms, regardless of the number of pairs used or the window size. Within the home office deduction domain, where single terms achieved average precision results in the mid to high 80's, pairs were in the mid to high 90's. These queries greatly exceeded our expectations and surpassed the 81.1% baseline by 15–20%. See Tables 7–10.

Pairs-Weissman

Number	RF-CKB1	RF-CKB2	RF-CKB3	RF-CKB4	RF-CKB5	RF-CKB6
of Pairs	Mopc/Pure	5 Impure	9 Mixed	8 Pure	7 Impure	Top 2 Layers
5	**93.5**	**88.2**	**93.5**	**92.6**	74.1	**91.5**
10	**95.4**	**94.6**	**96.3**	**94.7**	77.5	**95.4**
15	**95.5**	**94.2**	**96.7**	**95.8**	81.1	**96.2**
20	**95.7**	**93.0**	**96.2**	**95.9**	**82.6**	**96.5**
25	**95.1**	**92.2**	**96.3**	**96.8**	**85.0**	**97.0**
30	**96.1**	**93.2**	**96.1**	**96.9**	**91.7**	**97.0**
35	**96.1**	**93.0**	**95.9**	**97.0**	**90.7**	**96.9**
40	**96.1**	**92.8**	**95.8**	**97.3**	**91.1**	**97.1**

Table 7. For the top n pairs, window size 3, average precision scores achieved with *Weissman* using various RF-CKB's and INQUERY 1.5.6. Boldface indicates that a scores is above the 81.1% baseline. Shading indicates the maximum within a column.

Pairs-Weissman-RF-CKB6

Number of Pairs	size 3	size 4	size 5	size 6	size 7	size 8	size 9	size10
5	**91.5**	**88.2**	**85.1**	79.1	80.0	**84.7**	**88.2**	**87.8**
10	**95.4**	**92.8**	**90.0**	**88.2**	**88.1**	**89.2**	**88.4**	**91.7**
15	**96.2**	**96.6**	**95.4**	**96.0**	**96.2**	**91.6**	**90.4**	**92.2**
20	**96.5**	**97.0**	**95.1**	**95.5**	**96.1**	**95.3**	**92.9**	**92.6**
25	**97.0**	**97.2**	**96.5**	**95.7**	**96.0**	**95.6**	**96.3**	**96.6**
30	**97.0**	**97.2**	**96.8**	**96.6**	**95.8**	**95.6**	**96.5**	**96.6**
35	**96.9**	**97.3**	**96.6**	**97.0**	**96.9**	**95.6**	**96.4**	**96.6**
40	**97.1**	**97.3**	**96.8**	**97.0**	**97.0**	**96.8**	**96.3**	**96.7**

Table 8. For the top n pairs, window sizes 3-10, average precision scores achieved with the *Weissman* case using RF-CKB6 and INQUERY 1.5.6. Boldface indicates that a scores is above the 81.1% baseline. Shading indicates the maximum within a column.

Pairs-size 3-HOD cases

	Weissman			*Honan*			*Meiers*			*Soliman*		
Pairs	RF-CKB1 N=4	RF-CKB6 N=11		RF-CKB1 N=4	RF-CKB6 N=12		RF-CKB1 N=4	RF-CKB6 N=11		Root node N=1	RF-CKB1 N=6	RF-CKB6 N=13
5	**93.5**	**91.5**		**83.5**	**87.2**		**94.2**	**93.5**		10.1	**93.3**	**81.1**
10	**95.4**	**95.4**		**92.7**	**94.9**		**95.7**	**96.3**		10.1	**95.2**	**92.0**
15	**95.5**	**96.2**		**94.2**	**96.0**		**95.7**	**96.2**		17.8	**95.0**	**95.5**
20	**95.7**	**96.5**		**94.6**	**96.4**		**94.7**	**96.3**		19.6	**95.1**	**96.3**
25	**95.1**	**97.0**		**94.6**	**97.1**		**94.9**	**96.5**		33.0	**95.5**	**96.4**
30	**96.1**	**97.0**		**94.6**	**97.1**		**94.9**	**97.0**		43.2	**96.2**	**96.9**
35	**96.1**	**96.9**		**94.6**	**97.5**		**94.9**	**97.1**		56.6	**96.3**	**97.0**
40	**96.1**	**97.1**		**94.6**	**97.5**		**94.9**	**96.8**		72.0	**96.0**	**97.1**

Table 9. For the top n pairs of terms, window size 3, average precision scores achieved using RF-CKB1 and RF-CKB6 with INQUERY version 1.5.6 on home office deduction problem cases. Boldface indicates that a scores is above the 81.1% baseline. Shading indicates the maximum within a column.

Pairs-size 10-HOD cases

	Weissman			*Honan*			*Meiers*			*Soliman*		
Pairs	RF-CKB1 N=4	RF-CKB6 N=11		RF-CKB1 N=4	RF-CKB6 N=12		RF-CKB1 N=4	RF-CKB6 N=11		Root node N=1	RF-CKB1 N=6	RF-CKB6 N=13
5	80.6	**87.8**		62.7	**87.5**		70.2	81.0		10.1	80.5	**96.4**
10	**96.2**	**91.7**		**81.1**	**90.3**		**85.5**	**97.7**		10.1	**96.9**	**87.9**
15	**97.6**	**92.2**		**89.6**	**90.0**		**96.9**	**97.9**		10.1	**96.7**	**93.0**
20	**96.6**	**92.6**		**94.1**	**90.6**		**97.1**	**97.3**		10.1	**97.1**	**92.3**
25	**96.0**	**96.6**		**96.0**	**96.5**		**96.6**	**96.7**		10.1	**96.5**	**93.2**
30	**96.2**	**96.6**		**96.1**	**96.9**		**95.5**	**96.7**		10.1	**96.5**	**95.1**
35	**96.1**	**96.6**		**95.7**	**97.1**		**96.0**	**96.6**		10.1	**97.4**	**96.6**
40	**96.1**	**96.7**		**95.6**	**96.8**		**96.0**	**96.5**		10.1	**97.4**	**96.7**

Table 10. For the top n pairs of terms, window size 10, average precision scores achieved using RF-CKB1 and RF-CKB6 with INQUERY version 1.5.6 on home office deduction problem cases.

In the bankruptcy experiments, queries using pairs also exceeded their single term counterparts and did so by an average of approximately 20 percentage points. Of our three bankruptcy problem cases, queries using RF-CKB6 exceeded those using RF-CKB1 in two cases. Oddly, in one problem case (*Rasmussen*) when we added in the second layer from the claim lattice, average precision declined slightly across all window sizes.

In the bankruptcy domain, the pair scores with a given RF-CKB were not as consistently close across the different problem cases as in the home office deduction domain. For example, in the home office domain, pair scores with RF-CKB6 with a window size of 3 for all four problem cases were mostly in the 93–96% range. Within the bankruptcy domain, pair scores with RF-CKB1 were in the high 60's, mid 70's, and low 80's, on the three problem cases.

Overall, queries generated from pairs of terms exceeded queries generated from single terms, sometimes by 20 percentage points. Additionally, as before with terms, co-occurring pairs found in the RF-CKB6 texts, those texts in the top two layers of the claim lattice, out-scored the pairs found within the mopc texts, RF-CKB1.

9. Conclusions

The goal of this project is to create a system that provides access to more cases than usually afforded by a CBR system and with a more precise sense of relevance than provided by traditional IR systems. In our hybrid CBR-IR approach, knowledge-intensive reasoning is performed on a (small) corpus of cases represented in a CBR system, and the important cases selected from this analysis are used to drive a traditional text-based IR system to retrieve more like them. We use the CBR system to locate good examples of the kind of cases we want, and the IR system to retrieve more of the same.

Our approach integrates CBR with IR to:

- extend the range of retrievals to materials outside the scope of the CBR system;
- leverage the strengths of each
- achieve robust, decent results with minimal effort
- require no human in the loop, other than case entry
- be reproducible across a variety of problem cases.

In our experiments we have investigated whether, in the absence of other knowledge, a limited number of relevant full-text documents could be used to retrieve, with a high level of both recall and precision, additional relevant legal case texts from a large corpus. We have shown that using a modified version of relevance feedback, in which we have no initial query to modify, and a small number of well-chosen full-text documents, we can automatically and easily produce a query that achieves good results.

For single-term queries, the results are generally best when we use 100 or more terms. Note that since the sets of terms are generated automatically (and efficiently) by the relevance feedback module, the only added cost is that of INQUERY's evaluation of the query (which is linear in the number of terms). This is in contrast to the situation where the user must input terms or even natural language. Even if we are restricted to small sets of short texts that all discuss the same issue, we achieve good results. Within the home office deduction domain, the majority of mopc RF-CKB's exceeded the baseline and all of the RF-CKB's from the top two layers did, generally by nearly 10% and in a sustained manner. Using a large number of terms (300-400) does not degrade the query as much as might be expected, and, in fact, in most instances achieved results as good as or better than queries with fewer terms. Thus, not only is there limited cost associated with using this many terms, there is no real detrimental effect.

These results stand in contrast to those of Croft and Das [Croft & Das, 1990], who found that relevance feedback may not be beneficial when using a small set of relevant documents. We found this not to be the case. Their belief is due to the potential lack of concept coverage by a small set of documents. However, their documents were relatively short; they used abstracts whereas we used full-length legal cases. Also, in our collection, particularly in the mopc RF-CKB1, the terms (or pairs of terms) should be the most descriptive of the important relevant concepts, because these texts describe many, if not all, of the pertinent concepts relative to our problem case.

Overall, queries with pairs surpassed single-term queries, often by as much as 20 percentage points. Queries derived from the top two layer RF-CKB's generally surpassed their mopc counterparts, with both single terms and pairs. While we were unable to exceed the baseline within the bankruptcy domain with either terms or pairs of terms (exceeding an almost 90% average precision is a daunting task), we still achieved some very high average precisions for those queries. The home office deduction queries, of either type, almost always were able to surpass the baseline of 81.1%—even though it too was very high—and often by very wide margins.

In on-going work, we are re-examining our results using more restricted, problem-specific, senses of relevance. This is particularly important for our experiments with the bankruptcy domain, where both the document collection and the CKB used by the CBR module are exceedingly homogeneous. We are also investigating more problem-specific senses of relevancy in the home office deduction domain. We feel that end-users judge retrieval results according to specifics of the case and task at hand. Thus, problem-specific senses of relevancy are ultimately needed to demonstrate our approach. Nonetheless, we feel that the results presented here emphatically demonstrate the better performance of our hybrid CBR-IR approach over IR (or CBR) alone. We believe that the problem-specific analyses will make these conclusions unassailable.

Both case-based reasoning and information retrieval have their strengths and weaknesses. We should seek to exploit the strengths from one process by integrating it into the other where reasonable and if it remedies a weakness. CBR and IR lend themselves to many such cross fertilizations.

10. References

Ashley, K. D. (1990). *Modeling Legal Argument: Reasoning with Cases and Hypotheticals.* M.I.T. Press, Cambridge, MA. 1990.

Aleven, V., & Ashley, K. D. (1993). What Law Students Need to Know to WIN. *Proceedings of the Fourth International Conference on Artificial Intelligence and Law (ICAIL-93),* 152-161. Amsterdam. ACM Press.

Bing, J. (1987). Designing Text Retrieval Systems for "Conceptual Searching". *Proceedings of the First International Conference on Artificial Intelligence and Law (ICAIL-87)*, 43-51. Boston. ACM Press.

Blair, D. C., & Maron, M. E. (1985). An Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System. *Communications of the ACM*, 28(3): 289-299. March 1985.

Callan, J. P., Croft, W., B., & Harding, S. M. (1992). The INQUERY Retrieval System. In AM Tjoa and I. Ramos (Eds.), *Database and Expert Systems Applications: Proceedings of the International Conference*, 78-83. Valencia, Spain. Springer Verlag.

Creecy, R. H., Masand, B. M., Smith, S. J., & Waltz, D. L. (1992). Trading MIPs and Memory for Knowledge Engineering. *Communications of the ACM*, 34(8), 48-64, August 1992.

Croft, W. B., Cook, R., Wilder, D., Becker, H. (1995). "Providing Government Information on the Internet: Experiences with THOMAS." To appear in *Proceedings Second International Conference on the Theory and Practice of Digital Libraries*. June, Austin, TX.

Croft, W. B., & Das, R. (1990). Experiments with Query Acquisition and Use in Document Retrieval Systems. In *13th International Conference on Research and Development in Information Retrieval*, 349-365.

Croft, W.B. Turtle, H.R. & Lewis, D.D. (1991). The Use of Phrases and Structured Queries in Information Retrieval. In *Proceedings of the 14th Annual ACM/SIGIR Conference on Research and Development in Information Retrieval*, 32-45, Chicago, IL. October 1991. ACM.

Dick, J. (1987). Conceptual Retrieval and Case Law. *Proceedings, First International Conference on Artificial Intelligence and Law (ICAIL-87)*, 106-115. Boston. ACM Press.

Golding, A. R. & Rosenbloom, P. S. (1991). Improving Rule-based Systems Through Case-Based Reasoning. *Proceedings, Ninth International Conference on Artificial Intelligence (AAAI-91)*, 22-27. Anaheim, July 1991.

Goodman, M. (1991). Prism: A Case-based Telex Classifier, In Alain Rappaport and Reid Smith, (Eds.) *Innovative Applications of Artificial Intelligence—2*, 25-37. AAAI Press, Menlo Park, CA.

Hafner, C. D. (1987a). Conceptual Organization of Case Law Knowledge Bases. *Proceedings, First International Conference on Artificial Intelligence and Law (ICAIL-87)*, 35-42. Boston. ACM Press.

Hafner, C. D. (1987b). *An Information Retrieval System Based on a Computer Model of Legal Knowledge*. Ph.D. thesis, University of Michigan. Republished by UMI Research Press, Ann Arbor, MI (1981).

Hafner, C. D., & Wise, V. J. (1993). SmartLaw: Adapting "Classic" Expert System Techniques for the Legal Research Domain. *Proceedings of the Fourth International Conference on Artificial Intelligence and Law (ICAIL-93)*, 133-142. Amsterdam. ACM Press.

Kolodner, J. L. (1993). *Case-Based Reasoning*. Morgan Kaufmann.

Rissland, E. L., & Ashley, K. D. (1987). "A Case-Based System for Trade Secrets Law." *Proceedings First International Conference on AI and Law, (ICAIL-87)*, 60-66. Boston. ACM Press.

Rissland, E. L., Daniels, J. J., Rubinstein, Z., & Skalak, D. B. (1993). "Case-based Diagnostic Analysis in a Blackboard Architecture." *Proceedings of the Eleventh National Conference for Artificial Intelligence (AAAI-93)*, 66-72. Washington, D.C., July 1993.

Rissland, E. L., & Skalak, D. B. (1991). "CABARET: Statutory Interpretation in a Hybrid Architecture." *International Journal of Man-Machine Studies (IJMMS)*, June, 1991, (34): 839-887.

Rissland, E. L., Skalak, D. B. & Friedman, M. T. (1993). BankXX: A Program to Generate Argument through Case-Based Search. *Proceedings Fourth International Conference on Artificial Intelligence and Law (ICAIL-93)*, 117-124. Amsterdam. ACM Press.

Rissland, E. L., Skalak, D. B. & Friedman, M. T. (1994a). "Heuristic Harvesting of Information for Case-Based Argument." *Proceedings of the Twelfth National Conference for Artificial Intelligence (AAAI-94)*, 36-43, Seattle.

Rissland, E. L., Skalak, D. B. & Friedman, M. T. (1994b). *BankXX: Supporting Legal Arguments through Heuristic Retrieval.* TR 94-76, Dept. of Computer Science, University of Massachusetts, Amherst. Submitted for publication.

Rissland, E. L., Skalak, D. B. & Friedman, M. T. (1995). *Evaluating a Legal Argument Program: The BankXX Experiments.* TR 95-30, Dept. of Computer Science, University of Massachusetts, Amherst. Submitted for publication.

Rissland, E. L., Valcare, E. M, & Ashley, K. D. (1984). "Explaining and Arguing with Examples. *Proceedings of the Fourth National Conference on Artificial Intelligence (AAAI-84)*, 288-294, Austin.

Rose, D. E. & Belew, R. K. (1991). A Connectionist and Symbolic Hybrid for Improving Legal Research. *International Journal of Man-Machine Studies*, 35, 1-33.

Rose, D. E. (1994). *A Symbolic and Connectionist Approach to Legal Information Retrieval.* Lawrence Erlbaum.

Salton, G. (1989). *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.* Addison-Wesley.

Shimazu, H., Kitano, H., & Shibata, A. Retrieving Cases from Relational Data-Bases: Another Stride Toward Corporate-Wide Case-Base Systems. *Proceedings 13th International Joint Conference on Artificial Intelligence (IJCAI-93)*, 909-914. Chambery, France. Morgan-Kaufmann.

Stanfill, C., & Waltz, D. L. (1986). Toward Memory-Based Reasoning. *Communications of the ACM*, 29(12):1213-1228. December 1986.

Turtle, H. (1994). Natural Language vs. Boolean Query Evaluation: A Comparison of Retrieval Performance. In *Proceedings of the 17th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, 212-220, Dublin, Ireland, July 1994. ACM.

Turtle, H. R., & Croft, W. B. (1991). Evaluation of an Inference Network-Based Retrieval Model. *ACM Transactions on Information Systems*, 9(3): 187-222. March 1991.

Veloso, M. M. (1992). *Learning by Analogical Reasoning in General Problem Solving*. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA.