A Case-Based Approach to Intelligent Information Retrieval [*]

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

Phone: (413) 545-3639
Email: {Daniels, Rissland}@cs.umass.edu

Abstract

We have built a hybrid Case-Based Reasoning (CBR) and Information Retrieval (IR) system that generates a query to the IR system by using information derived from CBR analysis of a problem situation. The query is automatically formed by submitting in text form a set of highly relevant cases, based on a CBR analysis, to a modified version of INQUERY's relevance feedback module. This approach extends the reach of CBR, for retrieval purposes, to much larger corpora and injects knowledge-based techniques into traditional IR.

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 [9]. 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 [17], PACE [4], Anapron [7]). Our own CBR systems–HYPO [1] [11], CABARET [15], BankXX [12] [13]–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 they 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*®system. 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 [16]. 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. Given their awesome scope and ready availability, they are the stock-in-trade of many professionals, such as lawyers who use them extensively in legal research. 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 gone 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 [2].

A recurring 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.

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 broadly applicable 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 strengths of both CBR and IR in order to retrieve documents that are highly relevant to a problem case from a standard IR collection without the need for creating symbolic case representations for documents

[*] 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.

Appears in: Proceedings of the SIGIR '95 Conference
SIGIR '95 Seattle WA USA 1995 ACM
0-89791-714-6/95/07.$3.50

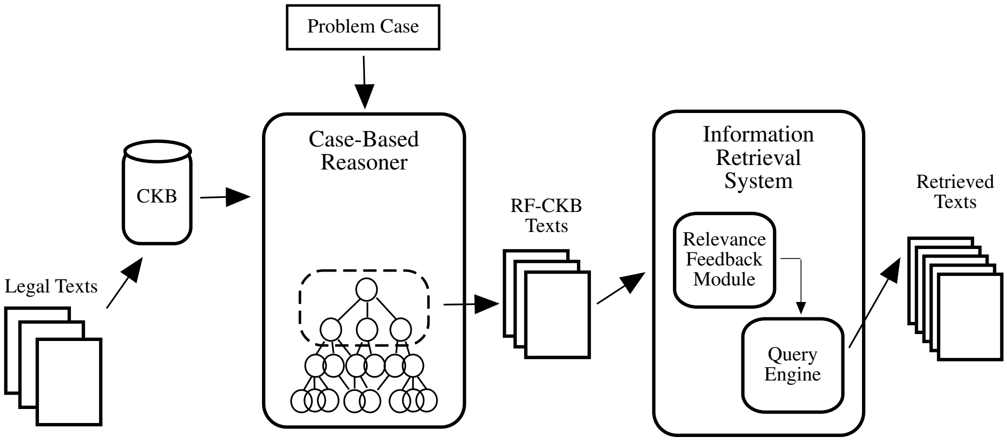

Figure 1: Overview of hybrid CBR-IR Architecture

in the collection. In particular, we address the issue of *how to automatically formulate good queries* based on a problem situation in order to perform retrieval from large text corpora. We would like to extend our case-based retrieval to the IR context 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.

Our hybrid CBR-IR approach combines knowledge-based CBR with text-based IR. It allows the results of the small-scaled CBR to be leveraged to dramatically larger text collections. 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.

In the next section, we give an overview of our approach and in Section 3 run through an example. Section 4 describes our use of relevance feedback. Other methodological details are given in Section 5. Section 6 discusses the experiments, Section 7 analyzes the results, and Section 8 summarizes this work.

2 System Overview

Our hybrid CBR-IR system works by first performing a standard CBR analysis of the input problem case and then using the results of the CBR analysis to drive text-based document retrieval. In particular, our system first uses its HYPO-style CBR module ([1] [11]) to analyze the problem case with respect to the cases that are represented in its *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*.

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). Then texts associated with these selected cases are passed to a modified version of the relevance feedback (*RF*) mechanism of the INQUERY IR system [3], which then generates a standard query consisting of the top n terms or top n pairs of terms generated from these texts. In the work reported here, for the texts we use the full texts of the court opinions. We call the set of cases selected from the CBR module's CKB and whose texts are given to the RF mechanism the *RF-CKB*. In this work we have experimented with a variety of RF-CKB's. They are discussed below in Section 5.2.

The query is then submitted as usual and full-text documents are returned to the user. Statistics on the results of such queries are given in Section 7. Note, the system performs no analysis on these texts. That would require natural language understanding of an unprecedented scale.

Ordinarily, INQUERY would not engage in relevance feedback until a retrieval, based on user input, had been made and a set of documents retrieved and presented to the user. In effect, our system uses "feedback" in the form of the RF-CKB on a null query. Our system's 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.

Note that while 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 special cases in the RF-CKB, the IR can be performed with respect to a text collection of arbitrary size. Instead of the user initiating the retrieval by making up a query, in our approach the user begins by inputting facts of a case. In effect, our system leverages its own "in-house" analysis of the problem case to a full-blown retrieval from an outside document base.

The IR document corpus may be many times larger than the case base available to the CBR system. In one of our application domains, an area of tax law, the full-text collection is 500 times larger; in the other it is about 20 times larger. 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 not ordinarily be usable by standard CBR. On the other hand, any form of knowledge-intensive reasoning of the kind at the core of CBR is not possible in the text collection by IR.

Of course what the user gets back is a set of documents, not a nicely polished CBR analysis or argument; this is up to the user. However, the user has been able to perform an intelligent, problem-based retrieval from a large collection ordinarily outside the reach of the CBR system.

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 [1], [10], [11], [12], [13], [15].

In brief, HYPO-style CBR systems work as follows. First, a problem case is input and analyzed to see what *dimensions*, 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.

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, these relevant cases are sorted according to a model of on-pointness. 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 *A* shares with the problem case properly contains those shared by *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. Note, our CBR systems use the claim lattice as a starting point for various other aspects of CBR, such as the generation of arguments or creation of hypotheticals. However, in this project, we only make use of the claim lattice.

We did not design new case representations for this project (i.e., for representing problem cases and cases in the CKB). Rather, we used pretty much *as is* the representations developed in two past CBR projects from our lab: CABARET [15] and BankXX [13] [12]. Both of these projects use 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 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 Example

To better illustrate the approach of our system we run through the following scenario based on a real personal bankruptcy case, the *Easley* [1] case. Suppose a client, Mr. Easley, approaches a lawyer about his attempt to file a personal bankruptcy plan. The Bankruptcy Court has denied approval of the plan because it fails to meet the *good faith* requirement. However, Mr. Easley believes that he does satisfy the requirement and wants to appeal the court's decision. He tells the lawyer various facts concerning his problem case. The lawyer inputs these facts to our system.

Having practiced in this area of law, the lawyer has knowledge of a set of past bankruptcy cases and their outcomes. Assume she has represented these in her own in-house case base, which is used by the CBR portion of our system. The system begins by performing a CBR analysis of her client's problem case, with respect to this in-house case base. The results of the CBR analysis are given in a claim lattice.

With *Easley* as the problem case and and a corpus of 45 hand-coded legal cases, originally used in BankXX, as the CKB, Figure 2 shows the resulting claim lattice. Note, the *Sheets, Rasmussen, Dos Passos*, and *Ali* cases are the most on-point cases.

Although the CBR system has analyzed only a subset of all the existing home office deduction cases, those in its CKB, the 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 Bankruptcy Case Law collection.

To perform this search, the system formulates a query by employing relevance feedback on a small set of special texts–the RF-CKB–selected from the claim lattice's cases. For instance, the mopc's are a good choice for use as an RF-CKB since they are the most highly similar cases to the problem case.

Specifically, our system uses texts (i.e., the case opinions) associated with the cases in the RF-CKB as the set of marked, relevant documents for relevance feedback. To do this, the CBR module passes the indices for these documents over to the relevance feedback module within the INQUERY system.

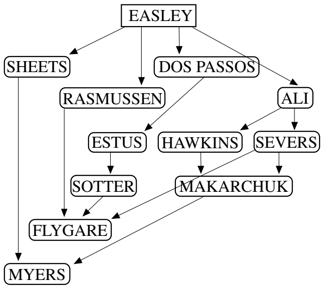

Figure 2: Claim lattice for the *Easley* case.

The relevance feedback module then selects and weights the top terms or pairs of terms from within these CBR-provided texts and forms a query. INQUERY acts on the query in the usual way to return a set of relevant documents from the larger collection. The system returns to the lawyer this set of highly relevant documents, some of which she already knows about since they were in her own personal CKB, to use in her research on Mr. Easley's legal problem. Of course, she has to read through and analyze these documents on her own. However, without any need for formulating queries or cleverly manipulating the retrieval engine, she has been able to access a large on-line document collection in a problem-based manner.

4 Relevance Feedback Module

Relevance feedback is a method for improving retrieval by having the user assess whether the retrieved documents are relevant to their information need. Relevance feedback has been found to improve precision by up to 40–60% [16]. Using information derived from the user-denoted relevant texts, a relevance feedback algorithm alters the weights of the terms in the original query and/or adds additional query terms. In all cases, the modified query is submitted back to the IR engine.

There are several variables subject to manipulation in relevance feedback experiments. They are:

1. the importance of the original query (re-weighting of the original terms),
2. the number of relevant documents to use,
3. the number of new terms to add,
4. the selection metric for finding terms to add from the relevant documents, and
5. the weighting metric for the new terms.

We add one additional variable, that of the type of term to add. For this paper we restrict these to be either terms or pairs of terms that must be found within a specified window or proximity. We use the same model for our pairs as was used for proximity pairs in [6]. We do not vary the selection metric nor the weighting metric, but use those developed by Haines and Croft [8]. They conducted a series of experiments using differing term selection and weighting schemes on two collections. One of their collections is very similar to the one used here: full-text legal cases. Therefore, we used their recommended term selection and weighting formulas. The selection and weighting criterion are described below.

For these experiments, there is no "original query," *per se*. Instead, the relevance feedback mechanism is given a null query

[1] *In re Easley*, 72 B. R. 948, 950 (Bankr. M.D. Tenn. 1987)

and a small number of legal texts, the RF-CKB, as its set of relevant documents. (See Section 5.2 below.) Because there is no original query to modify, re-weighting the original terms does not apply. The new query strictly consists of terms or pairs of terms found from within the RF-CKB.

The relevance feedback mechanism calculates the top terms within an RF-CKB. It then appropriately weights them and submits these terms as a new query against a large corpus of legal texts. Therefore, our experiments only vary the second and third considerations, that is: the number and set of relevant documents to use, that is, the RF-CKB, and the number of terms (or pairs of terms) to use.

Term Selection

To estimate the value of the words in describing the content of a text, it is desired to find terms that occur often enough within a document to describe it, yet also occur throughout the document collection with a frequency that is indicative of the number of relevant documents. For the selection of the top terms in this set of experiments, we used $(rdf * idf)$. *Rdf* is the number of documents in the RF-CKB in which the term appeared. *Idf* is $log(\frac{collection\ size}{df})$, where *df* is the total number of documents in the collection in which the term is found.

Term Weighting

To weight the selected terms, the following formula, or a variation on it, is frequently used: $tf * log(\frac{collection\ size}{df})$. *Tf* is the number of times a term occurred and aids recall. The second term is *idf*. We used a slightly different formula for weighting the top n terms: $rtf * (log(\frac{collection\ size}{tf})/max\ idf)$. *Rtf* is the number of times the term appeared within the set of relevant documents. In this case, the number of times a term was in the RF-CKB. This is different from *rdf* in that this is the raw count within the RF-CKB.

The top n terms are weighted and surrounded by a weighted sum operator to form the new query. An example query incorporating the top 10 terms is:

```
#WSUM(1.000000  1.720030 unacceler 0.860015
joinder 0.816118 sweep 0.816118 unapp 0.918787
alley 0.860015 sizabl 0.886941 realm 0.860015
appurten 0.868806 beaver 3.264471 hazard)
```

5 Methodological Information

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

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

1. *the home office deduction (HOD) domain*, which was the domain used by CABARET [15]; and

2. *the good faith bankruptcy domain*, used by BankXX [13].

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 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. When a case is run in a *de novo* manner it is deleted from the CKB and is analyzed as though never before seen by the system. The rest of the cases in the CKB become the cases against which it is analyzed. So far we have run experiments with 4 home office deduction cases and 3 bankruptcy cases as problem cases.

The bankruptcy problem cases were restricted to those cases from the BankXX corpus which were considered *meaty*, that is, they contained more than a set threshold of cited cases, theories, etc. in their opinions. We restricted ourselves to these meaty cases, since so many of the cases in the BankXX case base have very sparse hand-coded answers, which creates evaluation problems, for instance, instabilities in precision-recall statistics [14].

5.2 RF-CKB's – Cases for Seeding Relevance Feedback

For the home office deduction domain, we have experimented with several problem cases. The *Weissman*[2] case was the first with which we experimented. (The *Weissman* case is discussed in [15].) For each problem case, we experimented with different RF-CKB's. For *Weissman*, we examined the queries and resulting precision-recall results (see Figures 4 and 5) derived from six different types of RF-CKB's:

1. RF-CKB1. This RF-CKB consists solely of the set of mopc's. For the *Weissman* fact situation, there are four such cases. Coincidentally, this set of four cases happens to be *pure* in the sense that there are no other issues under consideration in them besides that of the home office deduction. An *impure* case discusses the home office deduction and one or more other issues. Of the 25 cases in the CBR module's CKB case base, seven cases are not pure. Within the other 103 home office deduction cases from the entire HOD-corpus, fewer than 10 were pure. In Figure 3, this RF-CKB is referred to as **Mopc/Pure**.

2. RF-CKB2. This RF-CKB, labeled **5 Impure**, consists of only *impure* cases; a random selection of five of them 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. This RF-CKB, labeled **Mixed**, 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. This RF-CKB, labeled **8 Pure**, contains all the pure texts from the top two layers of the claim lattice. It is comprised of the four mopc's and four additional cases from the second level for a total of eight texts.

5. RE-CKB5. This RF-CKB, labeled **7 Impure**, encompasses all the impure texts in the CBR module's CKB of 25 cases. There are 7 such cases.

6. RF-CKB6. The final RF-CKB uses all the cases in the **Top Two Layers** of the claim lattice. It contains 11 cases: eight pure texts (RF-CKB4) and three impure. Since it includes the top two layers, it contains RF-CKB1 consisting of only the mopc's.

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. That is, from the claim lattice generated for each problem case, we used (1) the mopc's as RF-CKB1 and (2) the top two layers of that claim lattice as RF-CKB6.

5.3 Building the Document Collections

To test our approach, we needed a collection of documents that includes cases that both the CBR and IR systems knew about. We thus constructed two test document collections against which to run retrieval experiments:

[2] *Weissman v. Comm.*, 751 F.2d 512 (2d Cir. 1984).

		RF-CKB1	RF-CKB2	RF-CKB3	RF-CKB4	RF-CKB5	RF-CKB6
	Original	Mopc/	5	9	8	7	Top 2
	FSupp	Pure	Impure	Mixed	Pure	Impure	Layers
Number of Documents	11953	4	5	9	8	7	11
Unique Terms in Collection	142749	1242	2430	2885	1952	2941	2767
Average Unique Terms per Text	530	477	842	680	516	834	589
Average Text Length	3250	1254	3321	2402	1533	3353	2031

Figure 3: RF-CKB sizes for the Home Office Deduction Experiments with the *Weissman* case.

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); the good faith sub-issue is discussed in Section 1325(a)(3).

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, [8], [18]. The additional texts came from the cases found in the CABARET CKB and those found when the query "home office" was posed to the on-line WestLaw®Federal Taxation Case Law database. We restricted the query cases to be between January 1986 and November 1993 and removed all redundant cases. The new collection contains 12,172 texts, of which, 128 cases discuss taking the home office deduction. Therefore, only about 1% of the cases in the HOD-corpus address the home office deduction (280A(c)(1)) issue we are interested in. Using the query *280A* we achieve an average precision of 81.1%.

On the other hand, 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 query *1325(a)* 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. Thus, this corpus is *very* focussed. The simple *one-phrase* query *good faith* against this corpus results in an average precision of 89.3%; this high value indicates that a high proportion of "good faith" cases actually use that phrase and that cases on other issues do not.

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. On the other hand, we found that most of our bankruptcy cases 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. For comparison purposes, Figure 3 gives the total number of unique terms in each RF-CKB for the *Weissman* case as well as the average number of unique terms for a text and the average document size for each RF-CKB. Figures for the original FSupp collection are given as well [8].

5.4 Answer Keys

For each problem case, we constructed an "answer key" that specified the documents to be considered as relevant. In the home office deduction domain, we used a very broad sense of relevance: 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. Thus, all problem cases were assigned the same set of texts as the correct answer, which includes those which CABARET would have considered on-point.

In the bankruptcy domain, we have two senses of the correct answer:

1. a general correct answer, much like that in the home office domain, in which a case must simply address the "good faith" issue; and

2. a problem-specific answer that consists of the documents for cases *actually cited* in the problem case.

We were able to use the second stricter definition of relevance because we already had hand-coded answers for individual problems available from our empirical evaluation of the BankXX system [14] 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.

6 Overview of Experiments

For each RF-CKB that we experimented with, the relevance feedback module selected, weighted, and formed a query with 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.

We initially tested queries generated by each of the six RF-CKB's described above (Section 5.2) with the *Weissman* case as the problem case. Results from the term experiments on the *Weissman* case are shown in Figure 4. Based on the results from this initial set of experiments, we tried using other home office deduction problem cases. However, we focused on RF-CKB1 and RF-CKB6 from the original half-dozen RF-CKB's to select our texts for the relevance feedback module. We also selected three problem cases from the bankruptcy domain and again used these same two RF-CKB's for use in the feedback module.

For each of the original six RF-CKB's and the *Weissman* problem case we also experimented with pairs of co-occurring terms: the relevance feedback module selected, weighted, and formed a query with the top 5, 10, 15, 20, 25, 30, 35, and 40 pairs found in the RF-CKB. For the *Weissman* RF-CKB's we also examined the top 50 and 100 pairs. (See Figure 5.) Window sizes ranged from 3 to 10, plus 15, 20, and 25 for some RF-CKB's. For experiments other than with the *Weissman* case, we only ran window sizes of 10 or less because the larger window sizes yielded worse results and were computationally much more expensive for all six RF-CKB's. We again examined both the mopc and top two layer RF-CKB's (RF-CKB1 and RF-CKB6, respectively) for the pairs experiments with other problem cases.

7 Results

7.1 Terms

Eleven point precision and recall tables were generated for each query. Figure 4 gives the average precisions for the six RF-CKB's on the *Weissman* case with different numbers of terms to form a query.

	RF-	RF-	RF-	RF-	RF-	RF-
	CKB1	CKB2	CKB3	CKB4	CKB5	CKB6
Num	Mopc/	5	9	8	7	Top 2
Terms	Pure	Impure	Mixed	Pure	Impure	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

Figure 4: For the top n terms, the 11 point average precision scores achieved with the *Weissman* RF-CKB's.

RF-CKB1, the mopc/pure texts, takes the longest to find a good 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%. The small impure RF-CKB2 achieves this average between 11 and 15 terms and the mixed RF-CKB3 needs 5 or less terms. Overall, RF-CKB6 achieves the best set of average precisions, RF-CKB4 next, and RF-CKB5 the worst.

Every RF-CKB results in improvement over the baseline average precision of 81.1% by the time 100 or fewer terms have been included. Significant results 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.

There is a large jump in the average precisions for most of the RF-CKB's. For example, within RF-CKB1, the jump is from 36.3% to 79.3% and occurs between 16 and 20 terms. For RF-CKB2, the jump is from 54.0 to 88.1% and happens with the addition of terms 11 to 15. 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 the mixed RF-CKB3 and both terms can be found in queries using 5 or more terms. Note that while *280A* is an obvious term to use in a query in the home office deduction area, *dwell* is less so. However, it was found automatically by our CBR-IR hybrid.

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 has (1) a limited number of smaller documents (see Figure 3) 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.

Similar results were found in other problem cases. In particular, RF-CKB6 always scored the best overall. 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. For this reason, we feel that RF-CKB6 is an excellent candidate to use as an RF-CKB to seed the retrieval engine. Using RF-CKB6 with a good number of terms (e.g., 100 or more) seems especially promising since including more terms requires no added effort.

In *Weissman*, RF-CKB1 may be handicapped since its ability to select high-value terms may be restricted by the purity of the texts. Since these texts only discuss one issue, the documents will contain many terms descriptive of the home office deduction. Yet, because so many terms occur across all four relevant documents, the high-value terms may be hard to discriminate and would thus be undervalued by the RF mechanism. Discriminating the high-value terms within the impure and mixed RF-CKB's might be more easily done than within this pure RF-CKB. 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. Further, within a mixed RF-CKB, the impure documents may provide the "noise" necessary for these 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 these high value terms", and where the high-value terms will vary according to which RF-CKB is used.

It is interesting to note that the mopc/pure RF-CKB1 had seven terms with weights exceeding 2.0, the impure RF-CKB2 had 13, and the mixed signRF-CKB3, 23. In fact, the largest weight in the mopc/pure RF-CKB1 was only 3.41. This would support the assertion that it is difficult to discriminate terms within the mopc/pure RF-CKB1 and easier to do so within the other RF-CKB's. In the impure and mixed RF-CKB's, there were several terms with much larger values; in the impure RF-CKB2 *use* received an 8.10, and in the mixed RF-CKB3 *use* received a 10.88, while *Lopkoff* (a case name) was weighted at 12.30. While these numbers of high valued terms are significantly different, it is difficult to judge whether they make an impact on the retrieval.

It is also noteworthy that all six RF-CKB's have more than one peak in their curve. For example, the mopc/pure RF-CKB1 has peaks at 5, 20, and 300 terms, and the small impure RF-CKB2 at 20 and 200 terms. (See Figure 4.) 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 in the set of all terms as they could be. Further, although they might not be selected until later, their weights compensate for the addition of these terms, plus the addition of the other, less descriptive terms. An alternative explanation is the well-known fact that legal concepts do not have necessary and sufficient descriptors.

We ran a similar set of experiments for three other cases from the home office deduction domain. These were *Honan*, *Meiers*, and *Soliman*.[3] This time we used only RF-CKB1 and RF-CKB6: that is, the mopc's and the top two layers of the claim lattice for generating terms.

These results were similar to those found with the *Weissman* case. In two cases, the system exceeded the baseline with the mopc RF-CKB's (although not significantly) using only 100 or fewer terms. The third case did not perform quite as well. It achieved average precisions in the 70's. For all three problem cases, using

[3] *Honan v. Comm.*, T.C. Memo. 1984-253; *Meiers v. Comm.*, 782 F.2d 75 (7th Cir. 1986); *Soliman v. Comm.*, 935 F.2d 52 (4th Cir. 1991).

RF-CKB6 the CBR-IR system exceeded the baseline within 15 or fewer terms and achieved better overall results than with the mopc RF-CKB's. The system exceeded the baseline using RF-CKB6 by between 8.6 and 11.9%.

Within the bankruptcy domain we selected three problem cases and also used these two same RF-CKB's. At this point, the bankruptcy term results do not appear to be as spectacular. The CBR-IR system achieved average precisions ranging from 48 to 67%. Better average precision occurs with higher numbers of terms (150 to 400). Once again, when the system uses RF-CKB6, the results are better than those with RF-CKB1. Random sets of four or five documents achieved average precisions in approximately the same range. It should be noted that the total number of documents used by the relevance feedback module was still very small; the largest RF-CKB only contained nine documents. Note however, that we restricted our queries to simple terms but compared them against a baseline query composed of a **phrase**.

7.2 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 in previous documents, 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.

Another filter for the selection metric was that if a pair was not contained within at least 20% of the relevant texts, it was discarded. In our scenario, that was fine since that basically meant that the pair had to appear at all.

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 a percentage of average precision in the mid to upper 80's, **pairs were in the low to high 90's. These queries greatly exceeded our expectations and surpassed the baseline by 15 – 20%.** (See Figure 5.)

	RF-	RF-	RF-	RF-	RF-	RF-
	CKB1	CKB2	CKB3	CKB4	CKB5	CKB6
Num	Mopc/	5	9	8	7	Top 2
Pairs	Pure	Impure	Mixed	Pure	Impure	Layers
5	93.5	88.2	93.5	92.6	71.4	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		93.0	95.9	97.0	90.7	96.9
40		92.8	95.8	97.3	91.1	97.1
50		92.7	95.1	97.5	91.2	96.8
100		91.6	94.5		89.8	96.9

Figure 5: For the top n pairs, window size 3, the average precision scores achieved with the *Weissman* RF-CKB's.

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*[4], 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-CKB1 for all four problem cases were 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.

8 Conclusion

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;
- leverages the strengths of each
- achieves robust, decent results with minimal effort
- requires no human in the loop, other than case entry
- is 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 150 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%. 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 detrimental effect.

[4] *In re Rasmussen*, 888 F.2d 703

These results stand in contrast to those of Croft and Das, [5], 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-CKB*, 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.

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.

References

[1] Kevin D. Ashley. *Modeling Legal Argument: Reasoning with Cases and Hypotheticals*. M.I.T. Press, Cambridge, MA, 1990.

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

[3] James P. Callan, W. Bruce Croft, and Stephen M. Harding. The INQUERY Retrieval System. In A. M. Tjoa and I. Ramos, editors, *Database and Expert Systems Applications: Proceedings of the International Conference in Valencia, Spain*, pages 78–83, Valencia, Spain, 1992. Springer Verlag, NY.

[4] Robert H. Creecy, Brij M. Masand, Stephen J. Smith, and David L. Waltz. Trading MIPs and Memory for Knowledge Engineering. *Communications of the ACM*, 35(8):48–64, August 1992.

[5] W. Bruce Croft and Raj Das. Experiments with Query Acquisition and Use in Document Retrieval Systems. In *13th International Conference on Research and Development in Information Retrieval*, pages 349–365, 1990.

[6] W. Bruce Croft, Howard R. Turtle, and David D. Lewis. 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*, pages 32–45, Chicago, IL, October 1991. ACM.

[7] Andrew R. Golding and Paul S. Rosenbloom. Improving Rule-Based Systems Through Case-Based Reasoning. In *Proceedings, Ninth International Conference on Artificial Intelligence*, volume 1, pages 22–27, Anaheim, CA, July 1991. AAAI.

[8] David Haines and Bruce Croft. Relevance Feedback and Inference Networks. Technical report, University of Massachusetts at Amherst, Amherst, MA, April 1993.

[9] Janet L. Kolodner. *Case-Based Reasoning.* Morgan Kaufmann, 1993.

[10] E. L. Rissland, J. J. Daniels, Z. B. Rubinstein, and D. B. Skalak. Case-Based Diagnostic Analysis in A Blackboard Architecture. In *Proceedings, The 11th National Conference on Artificial Intelligence*, pages 66–72, Washington D.C., July 1993. AAAI.

[11] Edwina L. Rissland and Kevin D. Ashley. A Case-Based System for Trade Secrets Law. In *Proceedings, First International Conference on Artificial Intelligence and Law*. ACM, ACM Press, May 1987.

[12] Edwina L. Rissland, D. B. Skalak, and M. Timur Friedman. *BankXX: Supporting Legal Arguments through Heuristic Retrieval.* Technical Report 94-76, University of Massachusetts at Amherst, Amherst, MA, 1994.

[13] Edwina L. Rissland, D. B. Skalak, and M. Timur Friedman. Heuristic Harvesting of Information for Case-Based Argument. In *Proceedings, The 12th National Conference on Artificial Intelligence*, pages 36–43, Seattle, WA, August 1994. AAAI.

[14] Edwina L. Rissland, D. B. Skalak, and M. Timur Friedman. *Evaluating a Legal Argument Program.* (95-30), 1995.

[15] Edwina L. Rissland and David B. Skalak. CABARET: Rule Interpretation in a Hybrid Architecture. *International Journal of Man-Machine Studies*, 34:839–887, 1991.

[16] Gerard Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.* Addison-Wesley, 1989.

[17] Craig Stanfill and David Waltz. Toward Memory-Based Reasoning. *Communications of the ACM*, 29(12):1213–1228, December 1986.

[18] Howard Turtle. 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*, pages 212–220, Dublin, Ireland, July 1994. ACM.