Using CBR to Drive IR [*]

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

Abstract

We discuss the use of Case-Based Reasoning (CBR) to drive an Information Retrieval (IR) system. Our hybrid CBR-IR approach takes as input a standard frame-based representation of a problem case, and outputs texts of relevant cases retrieved from a document corpus dramatically larger than the case base available to the CBR system. While the smaller case base is accessible by the usual case-based indexing, and is amenable to knowledge-intensive methods, the larger IR corpus is not. 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. Our system works by first performing a standard HYPO-style CBR analysis, and then using texts associated with certain important cases found in this analysis to "seed" a modified version of INQUERY's relevance feedback mechanism in order to generate a query. We describe our approach and report on experiments performed in two different legal domains.

1 Introduction

One forte of case-based reasoning (CBR) systems is their ability to reason about a problem case and, in particular, to retrieve highly relevant cases. However, this ability is limited by the availability of cases actually represented in a CBR system's case base. Among current CBR systems there are few with large case bases (say, larger than 1000 cases) and fewer still with both large case bases and large-sized cases, although all CBR systems use symbolic representations of cases and many perform highly sophisticated reasoning [Kolodner, 1993].

On the other hand, full-text information retrieval (IR) systems are not hampered by any lack of available cases (in textual form). There are huge case bases and individual cases are often very large (e.g., tens of pages of text); however, the level of representation is shallow at best (i.e., the text itself), and the indexing is weak (e.g., based on statistics of the collection)[Salton, 1989].

Thus we have two well-developed technologies, each with its own strengths and limitations. 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 retrieve documents that are relevant to a problem case from commonly available large text bases, without the need for creating a symbolic case representation for every document. 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?*

Instead of a user composing a query to initiate a retrieval, in our approach a user inputs facts of a problem case in a standard frame-based representation (e.g., a case template filled by facts). What the user gets back is a set of relevant texts retrieved from a document corpus many times larger than the case base available to the CBR system. Any further analysis of these retrieved texts, for instance, for the purpose of making a case-based argument, is up to the user.

Our hybrid CBR-IR system works by first performing a standard HYPO-style CBR analysis ([Ashley, 1990]; [Rissland and Ashley, 1987]), and then using 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 important cases found during the CBR analysis, such as most on-point cases. From this small set of "seed" documents, 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 set of noteworthy cases selected from this analysis; this set is smaller than those 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. Thus, the retrievals can be done from corpora much larger than is usual in CBR.

Our hypothesis is that the quality of documents retrieved via this hybrid system is better than via IR methods alone.

[*] 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.

This hypothesis has been borne out in our experiments. Our hybrid approach achieves a very fine level of performance, as measured by standard measures of precision and recall.

In the next section, we give further background on our task. In Sections 3 and 4, we present an overview of the architecture of our hybrid system, and give an example. In Section 5, we provide some background on the mechanics of query formation and on the domains explored. In Section 6, we discuss the experiment and in Section 7 analyze the results. We summarize in Section 8.

2 Background

Even though CBR partly ameliorates the knowledge acquisition bottleneck by taking advantage of problem cases as they arise, it is still time-consuming to build a case corpus of significant size if cases are represented in any depth. If the case base is constructed after the fact from pre-existing archives of textual materials, the task can be daunting.

Most CBR systems that have represented large numbers of cases have used fairly simple case representations (e.g., MBRtalk [Stanfill and Waltz, 1986], PACE [Creecy *et al.*, 1992], JOHNNY [Stanfill, 1988], Anapron [Golding and Rosenbloom, 1991]) or have used representations easily derived from solved problems [Veloso, 1992]. In a very few situations, large case bases have been constructed through a combination of case acquisition as a side-effect of customer service and follow-up knowledge engineering by a team specifically tasked with creating a case base [Shimazu *et al.*, 1993]. Our own CBR systems, which use detailed case representations–HYPO [Ashley, 1990] [Rissland and Ashley, 1987], CABARET [Rissland and Skalak, 1991], FRANK [Rissland *et al.*, 1993], BankXX [Rissland *et al.*, 1994a] [Rissland *et al.*, 1994b]–have typically had case bases in the range of three to five dozen cases.

Text-based IR can be used to access many extensive and widely-used commercial text collections in a variety of domains, such as commerce, medicine, and the law. 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 either West Publishing Company's *WestLaw*®or Mead Data Cental's *Lexis*®systems. These massive on-line corpora represent a tremendous resource and investment of capital.

However, users of current IR systems (even those accepting queries in natural language) must know how to manipulate them in order to get back truly relevant information. Often users are not even aware of the difficulties because nothing appears to go wrong. For instance, one study found that although many users felt that they had retrieved most of the right documents (i.e., that recall was high), in fact, they had retrieved only a mere 25% of the relevant texts [Blair and Maron, 1985].

The other typical problem is that of retrieving too much information, only some of which is relevant. For example, if one were gathering precedents to be used in writing a brief for a personal (Chapter 13) bankruptcy case involving the legal question of court approval of the plan proposed by the debtor, WestLaw could be used to query its collection of bankruptcy cases, for instance, with the query "1325(a)" (the cite to the relevant section of the bankruptcy statute). Even with an additional restriction to cases decided between 1982 and 1990, this query produces 959 cases; far too many to be looked over by even the most dedicated legal researcher or research team. A more restricted query "1325(a)(3)," the cite to the subsection addressing the narrower "good faith" requirement for plan approval, retrieves 386 cases; still too many. Adding information about the case at hand (e.g., profession of debtor, amount of debts, duration of plan) or placing further restrictions on date and jurisdiction would be ways to narrow down further the set of cases retrieved.

While traditional IR systems can access huge document bases, users of IR systems make the implicit assumptions that not all the relevant documents will be retrieved (i.e., recall will not be perfect) and that not all of those retrieved are relevant (i.e., precision will not be perfect). Users of CBR systems, on the other hand, often assume higher, if not perfect, levels of precision and recall. Our goal is to extend case-based retrieval to the IR context without sacrificing recall and precision and without enlisting the aid of an army of knowledge engineers to re-tool existing text collections.

By bringing in specifics of the case at hand–exactly the sort of information used by CBR systems–it is possible to retrieve a workable set of truly relevant cases, not just those that happen to share a particular statutory cite. This is what an experienced user does. 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, one can drive a retrieval engine to produce better results.

In our approach, knowledge about the problem case is input directly by the user. Knowledge about what makes one case similar to another–particularly what makes one case a good precedent to appeal to in making a legal argument about another–is embedded in HYPO-style CBR. Knowledge of the mechanics of forming a query is handled by the relevance feedback mechanism of INQUERY. Knowledge about the domain (e.g., personal bankruptcy law) is used in the CBR module, and knowledge about text (e.g., word frequencies) is used in the IR module. Thus we enhance traditional IR with knowledge through inclusion of CBR.

3 System Overview

Our system takes as input a problem case given in the form of a generic case-frame filled in with specific features. It outputs a set of documents considered relevant to the problem case. (See Figure 1.)

We did not design new case representations for this project. Rather, we used pretty much *as is* the representations developed in two past CBR projects from our lab: CABARET [Rissland and Skalak, 1991] and BankXX [Rissland *et al.*, 1994a] [Rissland *et al.*, 1994b]. We only added one additional slot to each case: the document identifier of the case's opinion in the text collection. The same case representation is used for representing a problem case and cases in the CBR module's *case-knowledge base* or *CKB*.

We use a standard HYPO-styled CBR module to perform the case-based reasoning [Ashley, 1990], [Rissland and Ashley, 1987]. It analyzes a problem case with respect to the cases in its CKB and generates a data structure called a *claim lattice*, which represents a sorting of cases relevant to the

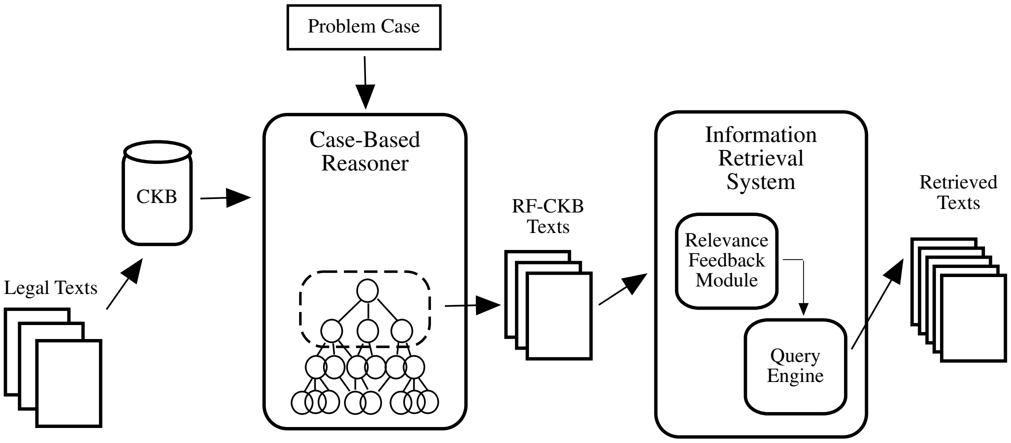

Figure 1: Overview of hybrid CBR-IR Architecture

problem case according to how on-point they are. From the claim lattice, our system selects certain special classes of cases to use in relevance feedback. We call the subset of the CKB cases selected via CBR analysis and used in relevance feedback the *RF-CKB*.

In brief, the CBR analysis is done as follows. First, the CBR module determines the *relevant* cases: these are cases that share at least one *dimension* in common with the problem case. Dimensions address important legal aspects of cases and are used both to index and compare cases. Next, the relevant cases are sorted according to *how* relevant or on-point they are. This is done by examining the intersection of each case's set of applicable dimensions with those applicable in the problem case. (Cases with no shared dimensions–that is, irrelevant cases–are not considered.) 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 it shares with the problem case contains those shared by *B* and the problem case. Maximal cases in this ordering are called *most on-point cases* or *mopc's*. The resulting sort of relevant cases can be shown in a so-called *claim lattice*. (See Figure 2 for an example.) Cases just below the root are the mopc's.

We use the INQUERY retrieval engine as our IR component. INQUERY uses an inference network model [Turtle and Croft, 1991], specifically, a Bayesian probabilistic inference net. 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.

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 can improve precision significantly [Salton, 1989]. In relevance feedback, a user tags texts as to their relevance. 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 modified query. The new query is then submitted to the IR engine with the hope of achieving improved recall and precision.

An RF module uses a selection metric to extract a set of terms from the relevant texts. The top n terms are then weighted according to another metric. For our experiments, we apply the selection and weighting metrics used in a similar application [Croft and Das, 1990]. A query consists of a weighted sum of terms.

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, there is no need for an initial user-provided query nor user-provided relevance judgments.

4 Example

To illustrate the workings of our system we run through the following scenario. A client approaches a lawyer about his attempt to take a tax deduction for his home office. The Internal Revenue Service has questioned the deduction, but the client, a college professor, believes that he is entitled to take it. He tells his lawyer various facts concerning his problem. She inputs these to the CBR-IR system.

Suppose the lawyer has knowledge of a set of previously decided home office deduction cases, for instance, cases she knows about from her own tax practice, and these make up the CKB used by the system. To be specific, suppose the problem case is the *Weissman* [1] home office deduction case and that the lawyer's CKB contains cases originally used in CABARET. Figure 2 shows the top two layers of the resulting claim lattice. *Drucker, Gomez, Honan,* and *Meiers* are the mopc's.

Using the lawyer's CKB, the system analyzes Mr. Weissman's problem and uses various important cases to seed a search for additional relevant cases from a larger corpus, say the WestLaw Federal Taxation Case Law collection. Suppose the lawyer asks the system to use the set of cases in the top two layers of the claim lattice as the RF-CKB because she knows these are always very relevant. This set contains all 11 cases shown in Figure 2. The indices for the texts associated

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

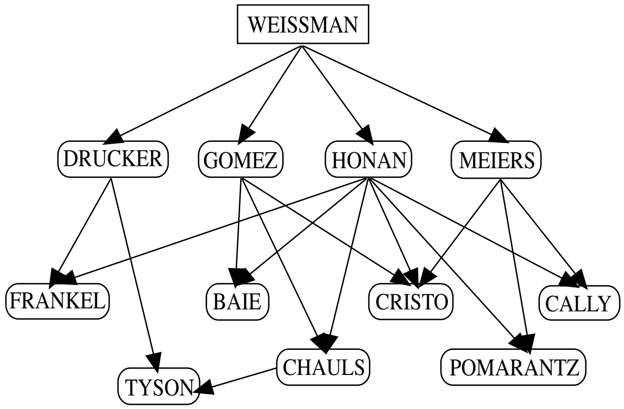

Figure 2: Top two layers of the claim lattice for the *Weissman* case.

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. Below is a sample query for this case, using the top 15 terms:

```
#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)
```

Some of these terms, like *280A*, are perfectly obvious. It is not hard to imagine 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 home deduction issue and *dwell* is the stem of *dwelling*, a term used frequently in the language of Section 280A of the IRS Code concerning deductions of various expenses in connection with business use of a home, rental of vacation homes, etc.; these are often quoted in case opinions. Others are not obvious at all, such as, *opera*, which no doubt comes from the *Drucker* case which concerned a musician in the Metropolitan Opera Orchestra.

Even an experienced user would be unlikely to use some of these terms if she needed to compose the query herself. Case names for cases that are not known or memorable, like *Curphey*, would surely not be used. (Presumably memorable cases would be included in the CKB). In fact, 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*.

Finally, our system returns to the lawyer those texts retrieved with the system-generated query. These include cases, like *Drucker*, *Baie*, etc. from the top two layers, which the lawyer already knew about, and new cases like *Dudley*, about a married couple, both of whom are college professors, which she didn't.

The lawyer now has a larger set of relevant documents for her research on Mr. Weissman's problem. It has located new cases unknown to the CBR module. Of course, she, herself, has to read and analyze these. However, without any need for formulating queries or cleverly manipulating the retrieval engine directly, she has been able to access a massive on-line document collection in a problem-based manner and discover relevant cases she might not have considered otherwise.

5 Methodology

In this section, we describe briefly domains of application, how we defined baselines and answer keys, and the main parameters varied in our experiments.

5.1 Domains

We have used two domains in our work thus far:

1. *the home office deduction domain*, used originally in our CABARET project [Rissland and Skalak, 1991];
2. *the good faith bankruptcy domain*, used in our BankXX project [Rissland *et al.*, 1994b].

CABARET's original case base consisted of 36 real and hypothetical cases concerning the home office deduction, as specified in Section 280A(c)(1) of the Internal Revenue Code. For this project, we re-used 25 of these cases as the CKB of our CBR-IR system in the first domain. BankXX's original case base consisted of 55 cases concerning the "good faith" issue for the approval of plans for (individual) debtors under Chapter 13 of the Bankruptcy Code, as specified in Section 1325(a)(3). For this project, we re-used 45 of these.

5.2 Problem Cases

In each domain, we have run a series of experiments by submitting problem cases, chosen from the CKB, to the CBR-IR system, which then treats it in a *de novo* manner by (temporarily) deleting it from the CKB and treating it as a new case. That is, we run the system on a problem case in a minus-one manner against a CKB consisting of the other cases. So far we have run experiments with 4 home office deduction and 3 bankruptcy cases.

5.3 Building the Corpus

To test our approach, we constructed two test document collections:

1. *HOD-corpus* consists of over 12,000 legal case texts addressing a variety of legal areas;
2. *Bankruptcy-corpus* consists of over 950 legal case texts addressing the issue of approval of a debtor's plan, as specified in Section 1325(a), and the sub-issue of *good faith* from 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 [Haines and Croft, 1993], [Turtle, 1994]. The additional texts are for cases contained in the CABARET CKB and cases found when the query *home office* was posed to the on-line WestLaw Federal Taxation Case Law database. We restricted the query to cases decided between January 1986 and November 1993. We added in these cases (with redundant cases removed) to build our HOD-corpus. All 25 of the CABARET cases are contained in the resulting collection. The HOD-corpus contains 12,172 texts in total. Of these, only about 1% (128 cases) discuss the home office deduction (280A(c)(1)) issue we are interested in.

We established a baseline for the HOD-corpus by using the simple one-term query *280A*. It is realistic query given that it is the relevant statutory cite and the one keyword that most users would probably start with. It does very well: 81.1% average precision. (*Average precision* is defined in Section 5.4). This represents a baseline for retrieval performance *using IR alone*.

By contrast, the Bankruptcy-corpus contains cases dealing only with the specific issue of debtor plan approval, as specified in Section 1325(a). We built this corpus by downloading all the cases that were found with the query *1325(a)* to the WestLaw Federal Bankruptcy Case Law database. We restricted the query to cases decided between 1982 and 1990. It contains all but the 10 earliest cases from the original 55-case BankXX CKB. In this corpus about 40% (385 cases) make specific reference to the narrower "good faith" issue. Thus, this corpus is *very* focussed.

For the bankruptcy domain, we established a baseline by using the simple *one-phrase* query *good faith* on the Bankruptcy-corpus. This baseline query, which uses IR alone, achieves 89.3% average precision. This high value indicates that a high proportion of "good faith" cases actually use that phrase and that cases on other issues do not.

Both text collections were built using the standard IR procedure of removing predefined "stop" words, that is, high frequency words that do not represent content and add little value for discrimination between documents (e.g., *and*, *but*, *the*, *a*), and stemming, that is, removing suffixes, to get at the root form of a word. 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 on the RF-CKB texts to produce a list of terms that may constitute a query. In addition, the RF module also gives each term a weight that represents its relative importance in the query.

Figure 3 gives the total number of unique terms in the various RF-CKB's from our experiments with the *Weissman* case, the average number of unique terms for a text, and the average document size for each RF-CKB. The figures for the original FSupp collection are taken from [Haines and Croft, 1993].

5.4 Answer Keys

For each problem, we constructed an "answer key" that specifies the documents to be considered as relevant. In these experiments, we used a very broad sense of relevance.

In the home office deduction domain, any of the 128 cases from the HOD-corpus that actually concerns a taxpayer trying to take the home office deduction is considered relevant. In the bankruptcy domain, any of the cases from the Bankruptcy-corpus that discusses the "good faith" issue is considered relevant. Thus, all problem cases in a given domain were assigned the same set of texts as the correct answer. For the most part, our answer keys contain cases that CABARET or BankXX would have considered relevant.

Answer keys are used to calculate precision and recall 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. When we use 11 levels of recall, it is called *11-point* average precision.

6 Experiments

In this section, we discuss our experiments with different RF-CKB sets and different numbers of terms that are used in the resulting query.

6.1 System Parameters Varied

For each problem case, we varied the following:

1. the RF-CKB used to seed the RF mechanism; and
2. the number of terms used in the INQUERY query.

We did not vary other parameters used in relevance feedback, such as the weighting metric. For our experiments, there is no "original query" *per se*. Instead, the RF module is given a null query and the RF-CKB as its set of relevant documents. Because there is no original query to modify, some concerns of relevance feedback, such as re-weighting of terms, do not apply.

For each RF-CKB, 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 because of a limitation of the RF module. Therefore, longer queries, such as all the terms from within a RF-CKB, were not tested.

6.2 RF-CKB's – Documents for Seeding Relevance Feedback

For the home office deduction domain, we selected 4 cases to use as problem cases. The *Weissman* case, discussed in our example, was the first problem case with which we experimented. We examined the queries and resulting precision-recall results derived from six different types of RF-CKB's:

1. RF-CKB1 consists solely of the set of mopc's. For the *Weissman* problem, there are 4 such cases. Coincidentally, these happen 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, 18 are pure. Of the other 103 home office deduction cases in the HOD-corpus, more than 90 were pure. In Figures 3 and 4, this RF-CKB is referred to as *Mopc/Pure*.

2. RF-CKB2 consists of only impure cases; a random selection of 5 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 is the union of RF-CKB1 and RF-CKB2 and so has both pure and impure texts. RF-CKB3 has the

		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.

advantage of having a large number terms from which to select the important ones.

4. RF-CKB4 contains all the pure texts in the top two layers of the claim lattice. It is comprised of the 4 mopc's and 4 additional cases from the second level for a total of eight texts.

5. RE-CKB5 contains all 7 impure texts in the home office deduction CKB.

6. RF-CKB6 contains all the cases in the *Top Two Layers* of the claim lattice. It contains 11 cases: 8 pure texts (RF-CKB4) and 3 impure. Since it includes the top two layers, it contains RF-CKB1 consisting of only the top layer (i.e., the mopc's).

After conducting experiments with these RF-CKB's on 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.

7 Results

For each RF-CKB used on a problem case, we calculated 11-point average precision scores. Figure 4 lists the scores for the six RF-CKB's used on the *Weissman* case with different numbers of terms used to form a query.

RF-CKB1 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 that exceeds the baseline of 81.1%. RF-CKB2 achieves this average between 11 and 15 terms, while 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 significant improvement over the baseline average precision of 81.1% by the time the queries have included 100 terms. The relative improvement over the baseline is nearly 10% in many cases. 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 RF-CKB3 and both terms can be found in all queries.

	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.

We did not expect that the mopc RF-CKB would do the worst among the set of RF-CKB's. In fact, we had hypothesized that it would perform the best. Its failure to do better may be due to the limited number and size of the documents in it since these are the texts from which the RF module draws and weights terms. For instance, RF-CKB1 had only 4 documents, but RF-CKB3 had 9 and RF-CKB6 had 11. Also, the average document in RF-CKB3 is approximately twice as large as that in RF-CKB1. The average RF-CKB6 document is not quite twice as large.

RF-CKB1 may also do poorly because its ability to select high-value terms may be handicapped by the purity of its texts. Its cases discuss *only* the home office deduction. Although its texts contain lots of terms highly descriptive for the home office deduction issue, their discriminatory power is probably undervalued by the RF mechanism because so many of them occur across all four texts. By contrast, discriminating high-value terms within the impure and mixed RF-CKB's is probably easier because they comprise a smaller proportion of each text, which may help the selection metric. The impure documents may provide the "noise" necessary for these high-value terms to stand out. A totally impure RF-CKB like RF-CKB2 and RF-CKB5 might contain too much noise however.

Thus the query to the IR system is *find me cases that look like this* where similarity for the IR engine is defined by the terms generated from the RF-CKB that is used. Different RF-CKB's provide different senses of similarity for the IR engine.

Because the top two layers of the claim lattice did so well, and knowledge about the "purity" of a text would generally not be known to the CBR system, we decided to continue experiments with the following RF-CKB's:

1. **RF-CKB1:** the set of mopc's for a problem case.
2. **RF-CKB6:** the top two layers of a problem case's claim lattice.

We ran a similar set of experiments for three other cases from the home office deduction domain. These were *Honan*, *Meiers*, and *Soliman*. [2]

These results were similar to those found with *Weissman*. Most of the queries generated using RF-CKB1 exceeded the baseline by the time 100 or fewer terms are used. Further, queries generated using RF-CKB6 always exceeded the baseline within 10 or fewer terms and achieved better overall results than those using RF-CKB1.

Within the bankruptcy domain we selected three problem cases and again 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, composed of the top two layers of the claim lattice of a problem case, it outperforms RF-CKB1, composed of the mopc's. Random sets of four or five documents achieved average precisions in the same range. It should be noted that the total number of documents used by the RF mechanism was still very small; the largest RF-CKB contained only 9 documents. Note however, that we restricted our queries to simple terms, but that the baseline query was composed of a **phrase**. Phrases can be much more descriptive of a text's content.

We are in the process of evaluating our CBR-IR approach with a change in the RF module that allows for the selection of pairs of terms found in proximity of each other. These pairs can be loosely thought of as phrases since we can specify how close the terms must be to each other. In on-going work, we are evaluating a more problem-specific sense of a "right" answer: a case is listed in the answer key only if the court opinion of the problem case actually cites it.

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 important cases selected from this analysis are used to drive a traditional text-based IR engine on a large corpus. 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. In this two-stage approach, the first stage is knowledge-intensive and depends on a highly articulated CBR notion of similarity; the second uses weak but easily applied text-based notions.

In summary, our approach integrates CBR with IR to:

- extend the range of retrievals to materials outside the scope of the CBR system;
- improve the recall and precision of ordinary information retrieval;
- 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.

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.

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 set 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 top-two-layers RF-CKB's did, generally by nearly 10%. Using a large number of terms (300-400) does not degrade the query as much as might be expected. In fact, in most instances our system achieved results as good as or better than with queries with fewer terms. Thus, not only is there limited cost associated with using this many terms, there is no detrimental effect.

Our results stand in contrast to those of Croft and Das, [Croft and Das, 1990], who claimed that relevance feedback may not be beneficial when using only a small set of relevant documents. We found this not to be the case. Their doubts are 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. Furthermore, our RF-CKB's are drawn from the top portion of the claim lattice and hence the terms generated in our approach are probably more descriptive.

Both case-base reasoning and information retrieval have their strengths and weaknesses. We are seeking to exploit the strengths, and remediate the weaknesses, of each, by pursuing a hybrid CBR-IR approach. Our preliminary results show that CBR and IR indeed lend themselves to beneficial cross fertilization.

9 Acknowledgements

We thank Michelle LaMar for the use of her relevance feedback code and providing assistance in building the text collections. We also thank Jamie Callan, David Aha, and the anonymous referees for their comments and advice.

References

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

[2] *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).

[Blair and Maron, 1985] 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.

[Callan *et al.*, 1992] 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*, pp 78–83, Valencia, Spain, 1992. Springer Verlag, NY.

[Creecy *et al.*, 1992] 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.

[Croft and Das, 1990] 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*, pp 349–365, 1990.

[Golding and Rosenbloom, 1991] 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, pp 22–27, Anaheim, CA, July 1991. AAAI.

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

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

[Rissland and Ashley, 1987] 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.

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

[Rissland *et al.*, 1993] 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*, pp 66–72, Washington D.C., July 1993. AAAI.

[Rissland *et al.*, 1994a] 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.

[Rissland *et al.*, 1994b] 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*, pp 36–43, Seattle, WA, August 1994. AAAI.

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

[Shimazu *et al.*, 1993] Hideo Shimazu, Hiroaki Kitano, and Akihiro Shibata. Retrieving Cases from Relational Data-Bases: Another Stride Toward Corporate-Wide Case-Base Systems. In *Proceedings, 13th International Joint Conference on Artificial Intelligence*, volume 2, pp 909–914, Chambery, France, 1993. IJCAI, Morgan-Kaufmann.

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

[Stanfill, 1988] Craig Stanfill. Learning to Read: A Memory-Based Model. In *Proceedings of the Case-Based Reasoning Workshop*, pp 402–413, Clearwater Beach, FL, May 1988. DARPA.

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

[Turtle, 1994] 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*, pp 212–220, Dublin, Ireland, July 1994. ACM.

[Veloso, 1992] Manuela M. Veloso. *Learning by Analogical Reasoning in General Problem Solving.* PhD thesis, Carnegie Mellon University, Pittsburgh, PA, August 1992.