Locating Passages Using a Case-Base of Excerpts.*

Jody J. Daniels
Lockheed Martin
Advanced Technology Laboratories
1 Federal Street
Camden, NJ 08102 USA
jdaniels@atl.lmco.com

Edwina L. Rissland
Department of Computer Science
University of Massachusetts
Amherst, MA 01003 USA
rissland@cs.umass.edu

Abstract

We present the SPIRE system, a hybrid case-based reasoning (CBR) and information retrieval (IR) system. It relies on a small case-base to seed the IR system to 1) select documents that are relevant to a presented problem case, and then (2) highlight within these retrieved documents passages that contain relevant information about specific case features. SPIRE aids not only problem-solving but knowledge acquisition by focusing a text extractor—person or program—on areas of text where needed information is likely to be found. Once extracted, this information can be used to create new cases or data-base objects thus closing the loop in the problem-solving—knowledge-acquisition cycle.

1 Introduction

When confronted with a new problem, we frequently try to relate it to one that we have dealt with previously. If we have knowledge of a similar prior problem, we might try to apply the same or a related solution. Similarly, we might want to evaluate a scenario and hypothesize the likelihood of various outcomes. By analogizing the new problem, or parts of it, to past experiences, we save ourselves from having to solve every problem from scratch. By relying on prior experiences for both of these processes, we are employing "case-based reasoning".

Case-based reasoners solve problems or examine and explain possible outcomes to a scenario by relying on prior similar experiences. Case-base reasoning (CBR) systems can be found in such diverse domains as cooking [8], medical diagnostics [11], manufacturing [10], game playing [12, 4], and legal applications [1, 15].

In the legal domain, finding similar cases is crucial because of the importance precedents play when arguing a case. By examining the outcomes and features of previous cases, a legal reasoner (e.g., judge, advocate) can decide how to handle a current problem. Similarly, a social worker may want to ensure fairness in the handling of a new situation and may look back on prior cases to ensure the treatment plans are equitable. A researcher can argue the merits of funding a research proposal by showing the differences between this proposal and what has been done in the past.

In order for CBR systems to generate useful and reliable results, they need to have reasonably sized "case knowledge bases" (CKB's). For some domains this is not a problem since the representations are either fairly simple [7, 13], or can be easily automatically generated [12, 16]. However, when the domain representation for a case is complex, it is typically the situation that building a CKB with large numbers of cases becomes an expensive proposition. Converting information into a symbolic case representation will frequently make use of the time and experiences of a subject matter expert. This is costly when there are large numbers of cases to be represented.

While working with long textual documents, it is desirable to automate the case acquisition process as much as possible. Natural language understanding of lengthy documents was beyond the state-of-the-art and manual processing was too time-intensive. Therefore, we hypothesized that we could use a semi-automated approach or a user-aided/interactive approach to focus attention on selected portions of the document. Given a problem situation, the system would first retrieve relevant documents from a full-text collection, and then in a second stage, locate relevant passages discussing case features within each document.

We explored this hypothesis for focusing attention by using a case-base of "excerpts" (called an *excerpt-ckb*) derived from a small set of documents to form a query for each feature. The query was then used with an information retrieval (IR) system to locate relevant *passages* within a set of domain documents. The passages retrieved were those that discuss the important domain features comprising the representation used by the CBR system.

We validated this approach using documents drawn from a statutory legal domain: court opinions address-

*This research was supported while the first author was at the University of Massachusetts at Amherst 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.

Slot Name	Slot Type
legal-case ( )	
citation	free text: In re Rasmussen, 888 F.2d 703 (6th Cir. 1988)
year	numeric: 1985
level	category: :bankruptcy-court :appeals-court
judge	proper name: David A. Scholl
summary	free text
procedural-status	category: 'plan-confirmation 'plan-filing
decision-for	category: 'debtor 'creditor
factual-prototype-link	category: 'student-loan 'farm
legal-prototype-link	category: 'Estus 'Kitchens
legal-theory-link	set: 'memphis-theory 'estus-theory

Figure 1: Part of a good faith bankruptcy case representation along with sample values.

ing the personal bankruptcy "good faith" issue, specifically §1325(a)(3) of the U. S. Bankruptcy Code. Figure 1 gives part of a case-frame and some sample values for the personal bankruptcy domain (as used in the BankXX system [15]).

We tested queries formed in various ways from the excerpts to see if they could perform as well as or better than a set of expertly crafted manual queries. The objective was to retrieve the most informative passages for each feature. By doing so, we did not automatically fill in the case representation, but we were able to focus attention on the places in the document that were most likely to contain the important feature information. Being able to focus attention in this way should reduce the effort required to search through lengthy documents for pieces of information.

The rest of the paper is organized as follows. We next present the architecture of the system and walk through an extended example. We follow this with a description of the methodology used, the experiments run, and the results. We give conclusions in the final section.

2 System Description

SPIRE operates in two stages. In the first stage a case-based reasoner interacts with an IR system to retrieve a set of textual documents relevant to the problem at hand [6]. The second stage, which we focus on here, relies on a case-base of excerpts to locate passages within the newly retrieved documents. This approach is motivated by our belief that past discussions provide good clues to the location of new discussions. (Figure 2 gives an overview of the system.) The CBR component is a HYPO-style CBR system [1] and the IR system is built upon INQUERY [3].

To locate the passages in the second stage, SPIRE generates queries by using excerpts from past discussions of a feature. Each excerpt is an actual piece of text containing relevant information about a case feature and comes from an episode of information location/extraction performed on a past document.

SPIRE gathers the existing excerpts for a feature and generates a new query to be run on an individual document. SPIRE presents the query along with a specified document to the IR engine, which divides the document into overlapping windows of 20 words each, approximating the length of a sentence. (Each word in the opinion will appear in two windows, except for the first 10 words.) Division of the document into 20 word windows is accomplished with the use of a `#Passage20` operator, which is wrapped around the entire query. Comparison between the query and each of the windows yields a set of ranked passages for presentation to the user (or possibly to an information extraction system).

This method of breaking a document into equal-sized, overlapping windows is based on work done with much larger-sized passages [2]. Hearst also worked with passages, although hers were thematically based [9].

We created the excerpt-ckbs by asking an individual familiar with the representation of the problem domain to read a small number of opinions corresponding to cases in SPIRE's case-base.[1] For each feature for which we wanted to create an excerpt-ckb the individual was instructed to select any portion of text that was useful for determining the feature's value and to select as much or as little text as was necessary. Excerpts could be a few terms, a few phrases, a full sentence, or even several sentences. It was also permissible and typical to select pieces from multiple locations throughout the text.

As passages are retrieved for each document and feature in stage two, the user can examine the presented passages, determine (if possible) the actual value of the feature in the document, and add it to the representation for the text. The user may also decide to add one or more of the retrieved passages, or selected portions of them, to the appropriate excerpt-ckb along with the feature and value. In this way, SPIRE may aid in the acquisition of additional knowledge about the context of each feature.

In summary, given a new problem or topic, SPIRE retrieves documents from a text collection in its first stage. It then highlights passages relevant to knowledge about specific features from each of these documents in its second stage.

3 Example

Our example comes from the personal bankruptcy domain. Under dispute is the debtors' plan for paying off

[1] This step would normally be done in conjunction with the creation of the representation for the domain and the encoding of the first few cases; thus eliminating the need for a full review of the texts.

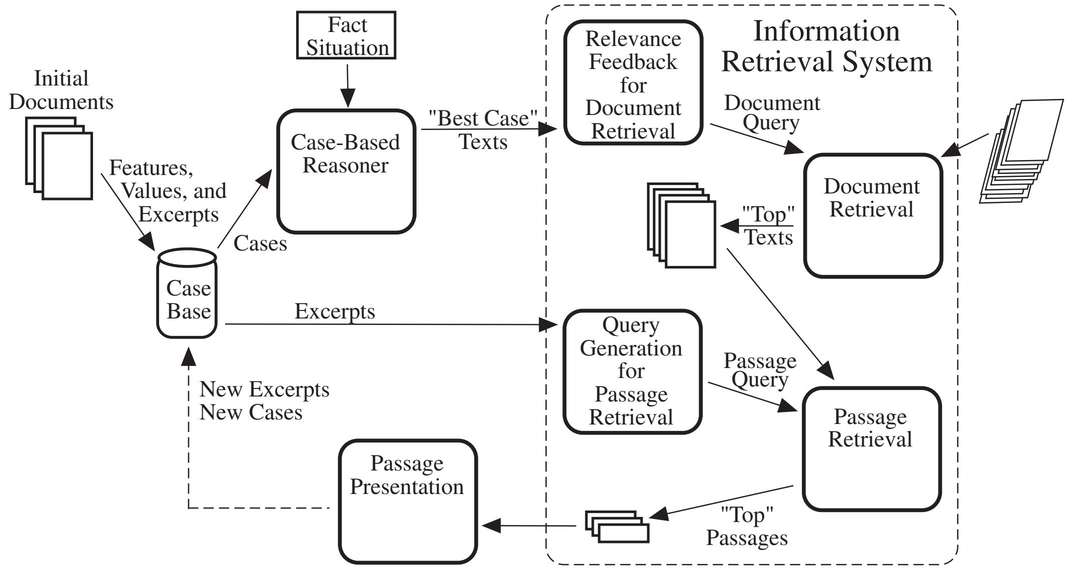

Figure 2: Overview of SPIRE.

debts under Chapter 13 of the bankruptcy code. The question presented is whether the plan has been proposed in "good faith".

We use the facts as found in the *In re Rasmussen* opinion as our problem case. In *Rasmussen*, the debtors proposed a plan to discharge a large debt that had been fraudulently obtained. The debtors had recently used a different section of the bankruptcy code to discharge other debts.

We submit this case to SPIRE, which compares it to those situations found in its own case-base. By examining the text opinions of the most related cases, SPIRE forms a query that it passes to the IR engine, which in turn retrieves what it believes are the most relevant opinions.

The next step is to examine these newly retrieved opinions for specific facts, in particular, the length of previous debt repayment plans. To do this, we direct SPIRE to locate passages within the top opinions that concern the feature called *duration*. SPIRE uses its excerpt-ckb on *duration* to form a query to retrieve passages.

The excerpt-ckb for *duration* contains 14 excerpts collected from 13 opinions. Combined, they contain a total of 212 words, 92 unique terms after stemming, and 59 terms after removing stop words. Sample excerpts are:

- "just over 25 monthly payments"
- "the plan would pay out in less than 36 months."
- "proposed a three-year plan for repayment,"
- "The Court would require the Ali's [sic] to pay $89 per month for 36 months."
- "Debtors propose payments of $25.00 weekly for 33-37 months."
- "would be paid in full after two years. In the four or five months following this two-year period, the unsecured creditors would be paid the proposed amount of 10% of their claims."

Notice that the first three excerpts are only fragments of sentences and that the third contains the value for the plan's duration, but expressed as a string. The fifth, a complete sentence, yields a range of values for the feature. Determining the value in the sixth, a sentence fragment plus a complete sentence, requires combining evidence from each portion to determine that the plan would run for a total of 28 or 29 months.

The top-rated retrieval for the *Rasmussen* problem is the *In re Sellers* document, so we use it to illustrate SPIRE's second stage. The IR engine divides the *Sellers* opinion into overlapping windows of 20 words each. SPIRE then generates a passage query from a feature's excerpt-ckb to be run against the *Sellers* opinion, divided into these windows. The IR engine carries this out and ranks the passages according to its belief that each is relevant to the feature query.

For this example, we allow SPIRE to use two simple methods to generate queries. The first combines the terms from all the excerpts into a single "natural language" query. Each word in each excerpt provides a possible match against the words in the retrieval window. Regardless of whether two words were in different excerpts, each contributes to the total belief score. We refer to this type of query as a *bag of words* query.

The second type of query is more restrictive when matching terms. Rather than amalgamating all the individual terms from the feature's excerpt-ckb into a single long query, each excerpt is individually compared to the twenty word window. The value achieved by the best single excerpt is the score given to the passage. (We refer to this type of query as the *sum* query because it is formed by wrapping a `#Sum` operator around each excerpt.) The *sum* queries test the ability of the individual excerpts to match against each passage. Part of both of these types of queries for the *duration* feature are shown below:

```
#Passage20(
 just over 25 monthly payments
 the plan would pay out in less than 36 months.
 proposed a three-year plan for repayment, ...)
```

...spirit
1420 and purpose of Chapter 13. The **debtor's proposed** Amended **Plan**
1430 called for **payments** of $260.00 per **month** for a **period**
1440 of **36 months**. Pursuant to [the] **Court** Order, the **debtor** has
1450 made 24 **monthly payments** without a default. Of course, at
the time of the original hearing...

Figure 3: Passages 1420, 1430, and 1440.

...The debtor's plan
2650 is scheduled to run for only fifteen **months** instead of
2660 the more common **period** of **three years**. This proposal to
2670 **pay** for only a limited time seems to relate with
particularity to repaying only...

Figure 4: Passages 2650 and 2660.

```
#Passage20(
 #Sum( just over 25 monthly payments)
 #Sum( the plan would pay out in less than 36
       months.)
 #Sum( proposed a three-year plan for
       repayment,) ...);
```

Posing these two queries over the *Sellers* opinion causes the retrieval of many relevant passages. Below are the top five passages for each query annotated with whether or not each is relevant:

Rank	Psg Strt	Belief
	Bag of Words	
1	1430	(0.404378) REL
2	1440	(0.404199) REL
3	2650	(0.402939) REL
4	2660	(0.402002) REL
5	1420	(0.401956) REL
	Sum of each Excerpt	
Rank	Psg Strt	Belief
1	1440	(0.405236) REL
2	1430	(0.405234) REL
3	2650	(0.403057) REL
4	2460	(0.402278) not REL
5	1420	(0.402145) REL

Figure 3 gives the text of the 1430 and 1440 passages, the top two passages for both retrievals. We boldface content terms that match those found in the excerpt-ckb and show word counts along with the text. (We include terms from the passage beginning at 1420 as it is ranked fifth by both queries.)

From either the 1430 or the 1440 passage we can ascertain that the debtor proposed a 36-month plan. From the 1440 passage we also learn that 24 payments had already been paid at the time of the hearing.

The third ranked passage for both queries is 2650. We display it in Figure 4 along with the 2660 passage. (We include passage 2660 in the Figure, as it ranked fourth with the *bag of words* query and ninth with the *sum* query.) The passages address the *duration* of a repayment plan in a related case that the judge is summarizing.

For the *sum* query, the fourth ranked passage, 2460, (given in Figure 5) is not relevant even though it contains many terms in common with the excerpts for *duration*. It discusses the amount of the *monthly payments* and the *monthly surplus*, rather than the *duration*.

In stage two, SPIRE has thus located passages in an individual document that are relevant to a specific case feature without requiring a user to pose a query. SPIRE can do this for any feature for which there is an excerpt-ckb, and for any individual document.

By comparison, suppose we were to intervene after SPIRE's first stage, and manually craft a query for the topic of *duration*. A sophisticated query would look like:

```
#Passage20( duration #Phrase(monthly payments)
   #Phrase(per month) #3(propose to pay) );
```

The `#Phrase` and `#3` operators add belief when the enclosed words are within 3 of each other, order dependent. (The `#Phrase` operator allows for a slight bit of flexibility based on the frequency of the enclosed terms within the entire document collection.) The term "plan" is not included in this query as it appears often within documents and its presence is not necessarily indicative of the *duration* of a plan. In fact, this particular query is the best of many manual queries that were tested over a set of documents. Posing this expert manual query against the *Sellers* opinion yields the following ranking of passages:

Rank	Psg Strt	Belief
1	2620	(0.415226) REL
2	2610	(0.415226) REL
3	2100	(0.410598) not REL
4	2090	(0.410598) not REL
5	2080	(0.410598) REL
6	1990	(0.410148) not REL
7	1980	(0.410148) not REL
8	1940	(0.410148) not REL
9	1930	(0.410148) not REL
10	1430	(0.410148) not REL

We display the top two passages in Figure 6. They do, in fact, contain information about the *duration* of the plan. However, the next relevant passages are not found until ranks 5, 13, and 19. Of the top ten passages seven are not at all pertinent. In general, one would like to achieve a higher percentage of hits in the top-ranked passages.

...the debtors'
2460 **proposed monthly payment** under the Amended **Plan** is $260.00, and
2470 the **monthly** surplus of income is now over $1,000. The

Figure 5: Passage 2460.

2610 ...history and likelihood of continued future advances was good even
2620 in 1981. Third, the **duration** of the Plan is three
2630 years. The court in *In re Estus*, supra, sheds some
light on this factor...

Figure 6: Passages 2610 and 2620.

4 Methodology

We now describe the various types of features, generation of answer keys, and the evaluation metric that we used in our experiments on focusing attention.

4.1 Features examined

We selected ten features from a bankruptcy good faith case representation for use in testing. The feature set included: *sincerity* (was the debtor sincere in proposing the plan), *special-circumstances* (were there any extenuating events affecting the debtor), *loan-due-date*, *plan-filing-date*, *procedural-status* (e.g., appeal or affirmation), *likelihood-income-increase* (probability of the debtor's income increasing in the next few years), *debt-type* (e.g., educational or consumer), *profession*, *monthly-income*, and *duration* (of the proposed repayment plan in months).

For this set of ten features we gathered excerpts for the excerpt-ckbs from a small set of documents, only 13 opinions. After stemming and removing non-content terms, the average number of remaining content terms for the ten features was 46.8.

4.2 Answer Keys

In order to evaluate SPIRE's ability to locate relevant passages, we needed to create answer keys specifying where within our test documents there was text discussing each of the features. We hired several undergraduates to make the relevance judgments. Given a set of written instructions describing each feature and samples of the sort of text that was relevant, they created the answer keys.

4.3 Evaluation metric

Most information retrieval systems are judged on precision and recall. These measure what percentage of the retrieved items are relevant and what percentage of the relevant items are retrieved, respectively. In our scenario, however, we are not concerned with locating every relevant item. Rather, we are more interested with how quickly we can get some of the relevant information to the user. Since our perspective is user oriented we calculated precision over the top 1, 3, 5, 10, 15, and 20 passages. (Complete recall and precision results, as well as expected search length values, can be found in [5].)

5 Experiments

We used SPIRE to retrieve relevant documents for two different initial problem situations. From these we gathered the top documents retrieved and created a collection of 20 test documents. Using various methods for passage query generation, we examined results for the queries for the 10 features over these 20 documents. We experimented with a dozen different methods by which SPIRE could generate passage queries from the excerpt-ckbs.

To provide another point of comparison, we also had a human expert, familiar with both the domain and INQUERY query operators, create queries. The expert could use any of the available query operators and was able to refine the queries as many times as desired. There was no time limit imposed. We allowed some additional query refinement after judgments for the 20 test documents became available. The goal was to create the best possible query, thus setting a high standard for SPIRE to attempt to achieve. The best query for each feature was retained for use as a baseline. We refer to this set as the *manual* queries.

In the experiments reported here, we are concerned primarily with five passage query formation methods, which we call the *base* set:

1. bag of words
2. sum
3. bag of words plus phrases
4. sum plus phrases
5. set of words

We have already described the first two methods. The third and fourth queries were created by replacing some of the terms with phrases in the first and second type of query. Phrases (i.e., pairs of words found within a specified proximity to each other) were selected by running INQUERY's phrase generation tool over the excerpts. The fifth type of query was created by taking a *bag of words* type query and removing all duplicate terms.

We also tried several other sets of queries where we altered the term weightings and varied the number of terms from the excerpt-ckb that could be included in the query. The former addressed a weighting that had proven useful in other contexts and the latter addressed the concern that the excerpt-ckbs might contain too many terms, which might allow too many passages to

Feature	Manual	Bag of Words	% Chg	t-test	Sum	% Chg	t-test
Debt Type	0.6950	0.7450	7.2	0.2981	0.7550	8.6	0.1625
Duration	0.5050	0.5200	3.0	0.6513	0.5200	3.0	0.6663
Future Income	0.7056	0.6167	-12.6	0.0722	0.6167	-12.6	0.0453*
Loan Due Date	0.1769	0.2308	30.4	0.4531	0.1923	8.7	0.8243
Monthly Income	0.6000	0.6211	3.5	0.5311	0.6316	5.3	0.3012
Plan Filing	0.4150	0.4150	0.0	1.0000	0.4050	-2.4	0.8098
Proc Stat	0.7850	0.3650	-53.5	0.0000*	0.3650	-53.5	0.0000*
Profession	0.4111	0.4222	2.7	0.8704	0.3556	-13.5	0.4136
Sincerity	0.4300	0.5950	38.4	0.0095*	0.6100	41.9	0.0040*
Special Circ	0.3789	0.4211	11.1	0.4241	0.3368	-11.1	0.3227

Table 1: Comparison of the averages of the precisions at a cutoff of 10 passages between the manual and the *bag of words* and *sum* queries. Values with a * are significant.

receive high belief scores. None of these techniques performed nearly as well as the base queries and we do not discuss them further here.

6 Results

Comparison of the first two types of base queries—the *bag of words* and *sum* queries—reveales that they performed about equally. Each did slightly better than the other on specific features, but overall the differences were not large.

Comparing each of these query types to their phrase-enhanced counterparts reveals that the queries without phrases generally perform better. This indicates that, in our application, phrases are slightly detrimental, rather than being beneficial. This is surprising since phrases have typically increased retrieval performance, although, recent improvements in term weighting methods have reduced their impact [14].

Closer examination of the automatically-generated phrases offers an explanation for this. In the situations where phrases have proven beneficial, the phrases were typically noun groups. Unfortunately, in this domain, some of the noun phrases generated were not at all indicative of the feature. For example, the tool generated "two unsecured" and "2000 bank" for *duration*, "net disposable" and "sales capacity" for *monthly income*, and "percentage repayment" and "student loan" for *procedural status*.

The phrases that a user might cite as being descriptive for our feature set include words that are typically removed from queries: "payments of" and "proposes to pay" for *duration* and "became due" for *loan due date*. These all contain important prepositions that the IR system removes when stopping the query.

Our next result is that term frequency information, when given in a query, is important. The *bag of words* queries achieve results significantly better than the *set of words* queries. (For significance testing we used a t-test at the .05 level on average non-interpolated precision and precision at six cutoff levels. The cutoff levels were 1, 3, 5, 10, 15, and 20 passages.) This indicates that if a term occurs in multiple excerpts, it should be given additional weight in a passage query.

In summary, among the excerpt-based queries, the *bag of words* and *sum* queries performe about equally well, and both of these types of query performedslightly better than the queries that use phrases. Allowing multiple occurrences of a term query, *bag of words* does better than restricting a word to appear only once in the query, *set of words*.

Lastly, we compare the manually-crafted queries to the *bag of words* and *sum* queries. Table 1 compares the averages of the precisions at a cutoff of 10 passages between the manual and two best excerpt-based queries. The table shows that the excerpt-based queries perform quite well.

The reason for the success of the manual query on the feature of *procedural status* is easily explained: discussion about the feature will normally include at least one of a small set of easily enumerated keywords, such as "confirmation" or "appeal". Not all of these terms are present in the excerpt-ckb, but are included in the manual query. This is an instance where knowledge of a domain-specific vocabulary, particularly one with a limited number of terms, is easily enumerated and should be used as the excerpt-ckb.

Conversely, there are quite a few ways to describe the feature of *sincerity*, many of which are not included in the expert manual query. Due to the variance in the ways that this feature could be discussed, having a case-base of excerpts upon which to draw terms works particularly well in locating similar portions of text in new documents.

7 Conclusion

In this paper, we have presented the SPIRE system that, given a problem case, first uses a case-based reasoner to assist with the retrieval of relevant documents from a full-text collection, and then using a case-base of excerpts for generating a feature-based query locates relevant passages within each document. The queries needed for both the document and passage retrievals are generated automatically by SPIRE in a case-based manner. We found that the queries based on our excerpt-ckbs did as well as or better than most of the expertly crafted queries.

One advantage of SPIRE's hybrid CBR-IR approach is that it creates queries using terms the user may not have thought to include. It also makes use of past experience in the form of excerpted passages known to be relevant to the feature in order to locate new passages. Context in this limited form enables the excerpt-ckb queries to do quite well. SPIRE achieves its results robustly and with minimal effort on the part of the user.

References

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

[2] James P. Callan. Passage-Level Evidence in Document Retrieval. In *Proceedings of the 17th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pages 302–310, Dublin, Ireland, July 1994. ACM.

[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] James P. Callan, Tom E. Fawcett, and Edwina L. Rissland. CABOT: An Adaptive Approach to Case-Based Search. In *Proceedings, 12th International Joint Conference on Artificial Intelligence*, volume 2, pages 803–808, Sydney, Australia, August 1991. IJCAI.

[5] Jody J. Daniels. *Retreival of Passages for Information Reduction*. PhD thesis, University of Massachusetts, Amherst, Amherst, MA, September 1997.

[6] Jody J. Daniels and Edwina L. Rissland. A Case-Based Approach to Intelligent Information Retrieval. In *Proceedings of the 18th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pages 238–245, Seattle, WA, July 1995. 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] Kristian J. Hammond. *Case-Based Planning*. Academic Press, Inc., 1989.

[9] Marti A. Hearst and Christian Plaunt. Subtopic Structuring for Full-Length Document Access. In *Proceedings of the 16th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pages 59–68, Pittsburgh, PA, June 1993. ACM.

[10] Daniel Hennessy and David Hinkle. Initial Results from Clavier: A Case-Based Autoclave Loading Assistant. In *Proceedings of the Case-Based Reasoning Workshop*, pages 225–232, Washington D.C., May 1991. DARPA.

[11] Phyllis Koton. *Using Experience in Learning and Problem Solving*. PhD thesis, Massachusetts Institute of Technology, Boston, MA, May 1988.

[12] Wendy G. Lehnert. Case-Based Problem Solving with a Large Knowledge Base of Learned Cases. In *Proceedings, Sixth National Conference on Artificial Intelligence*, volume 1, pages 301–306, Seattle, WA, July 1987. AAAI.

[13] Wendy G. Lehnert. Case-Based Reasoning as a Paradigm for Heuristic Search. Technical report, University of Massachusetts at Amherst, Amherst, MA, October 1987.

[14] Mandar Mitra, Chris Buckley, Amit Singhal, and Claire Cardie. An Analysis of Statistical and Syntactic Phrases. In *5TH RIAO Conference, Computer-Assisted Information Searching On the Internet*, pages 32–45, Chicago, IL, October 1997.

[15] Edwina L. Rissland, D. B. Skalak, and M. Timur Friedman. *BankXX: Supporting Legal Arguments through Heuristic Retrieval*. *Artificial Intelligence Review*, 10(1-71), 1996.

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