Integrating IR and CBR to Locate Relevant Texts and Passages. *

Jody J. Daniels and Edwina L. Rissland
Department of Computer Science
University of Massachusetts
Amherst, MA 01003 USA
Email: {Daniels, Rissland}@cs.umass.edu

Abstract

This paper presents the SPIRE system, a hybrid case-based reasoning (CBR) and information retrieval (IR) system that (1) from a large text collection, retrieves documents that are relevant to a presented problem case, and (2) highlights within those retrieved documents passages that contain relevant information about specific case features. We present an overview of SPIRE, run through an extended example, and present results comparing SPIRE's with human performance. We also compare the results obtained by varying the method by which queries are generated. 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 process.

1 Introduction

The quantity of text available on-line is growing explosively. On the World Wide Web alone, there is information on every conceivable activity. While it is clearly a boon to have so much information at one's electronic fingertips, it is also a burden to manipulate it effectively. In particular, it is no easy task to find relevant nuggets of information buried in the mountains so readily retrieved.

For example, when researching a legal problem, there is typically a plethora of material that must be examined: statutes, regulations, precedents, court documents, etc. Some, like regulations, are fairly easy to peruse. Others, like precedents, are not. Nonetheless, they must be examined carefully so that they can be mined for their specific facts and conclusions, and then compared and contrasted with the problem at issue.

Although cases and other legal materials are readily available on-line, they are available only in full-text form. In order to harvest valuable bits of useful information from them, someone must read each one, and then manually take notes or build a case-frame or database entry for each piece of data. This can be tedious, time-consuming, expensive, and even error-prone. Perhaps the greatest danger is missing some relevant piece of information.

To address these issues, we have developed a system that provides an effective means of locating textual regions likely to discuss important problem-solving features, without incurring the expense of reading entire documents. SPIRE (Selection of Passages for Information REduction) is a hybrid case-based reasoning (CBR) and information retrieval (IR) system that works in two stages:

1. from a large text collection, retrieves documents that are relevant to the presented problem case, and
2. highlights within those retrieved documents passages that contain information relevant to specific case features.

Once extracted, this information can be used to create new data objects thus closing the loop in the problem-solving-knowledge-acquisition process.

SPIRE employs two kinds of case-bases: (1) a case-base of past problem cases (precedents) represented as case-frames of features for use by a HYPO-style CBR module; and (2) for each case feature in the case-frame, a case-base of actual text excerpts, culled from past cases, that contain useful information about the value of the feature. In both stages, SPIRE uses cases to drive the INQUERY IR engine. In both stages, the cases are used as the basis for generating queries, which are then run by INQUERY [1] in the usual way. In the first stage, the query is run on the text col-

*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.

lection; in the second stage, it is run on individual documents.

The rest of the paper is organized as follows. In Sections 2 and 3, we present SPIRE's architecture and walk through an extended example. Section 4 describes background on the domain. In Section 5 we describe different techniques for forming queries and discuss SPIRE's performance and we summarize in Section 6.

2 System Description

SPIRE operates in two-stages.

1. from a large text collection, SPIRE retrieves documents that are relevant to the presented problem case, and
2. within those retrieved documents, SPIRE highlights passages that contain information relevant to specific case features.

Figure 1 gives a system overview.

In the first stage, SPIRE is given a new problem situation. It uses its HYPO-style CBR module to analyze it and select a small number of most relevant cases from its own case-base consisting of symbolically represented texts. In standard CBR fashion, SPIRE determines the similarity of each known case to the new problem and represents the results of this analysis in a standard claim lattice.

The most relevant cases from this analysis–typically the cases in the top two layers of the claim lattice–are then used to 'prime the pump' of INQUERY's relevance feedback module. This set of cases is called the *relevance feedback case-knowledge-base* or RF-CKB. The original texts of the cases in the RF-CKB are passed to the INQUERY IR engine, which then treats them as though they had been marked relevant by a user. INQUERY automatically generates a query by selecting and weighting terms or pairs of terms from within this set. This query is then run against the larger corpus of texts, with the result that new documents are retrieved and ranked according to INQUERY's belief as to their relevance to the posed query. (Details on this process of stage one can be found in [4, 6].)

In the second stage, SPIRE locates germane passages within each of the texts retrieved in stage one. To locate text segments that discuss a particular feature, SPIRE once again uses a hybrid CBR-IR approach but this time its task is to locate passages (within a document) rather than documents (within a collection). This approach is motivated by our belief that past discussions provide good clues to the location of new discussions.

To locate these passages, 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 case. (Example excerpts are given in the next section.)

SPIRE gathers the existing excerpts for a feature and generates a new query to be run on individual documents. There are numerous techniques for transforming the excerpts into passage retrieval queries. (See Section 5.1.) SPIRE presents the query along with a specified document to the IR engine which, in turn, retrieves the top ranked passages for presentation to the user (or possibly to an information extraction system).

Thus, excerpts are used analogously to the RF-CKB's of stage one: their terms are used to to generate queries. The difference is that (at this point in our development of SPIRE) there is no selection of excerpts according to some model of relevance since all are used to generate the query. At some point, when these excerpt collections become larger, the question of winnowing or selecting excerpts will become an interesting one.

We created these case-bases of excerpts by asking an individual familiar with the representation of the problem domain to highlight a small number of opinions corresponding to cases in SPIRE's case-base.[1] The individual was instructed to highlight any portion of text that was useful for determining the feature's value and to highlight 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 to highlight pieces from multiple locations throughout the text.

As each document and feature is addressed in stage two, the user (or information extraction program) 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 case-base along with the feature and value. In this way, SPIRE may aid in the acquisition of additional knowledge about the context of each feature.

[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.

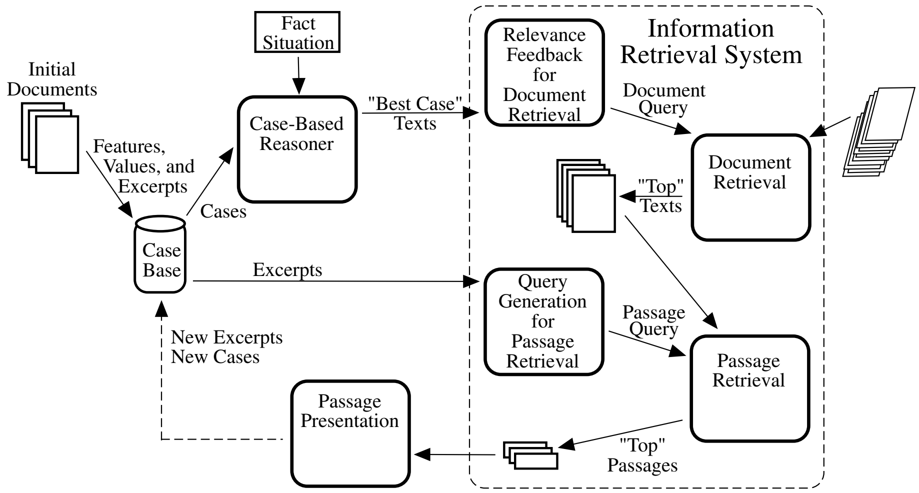

Figure 1: Overview of SPIRE.

3 Example

Our example comes from the domain of personal bankruptcy. Under dispute is the proposal of a plan for paying off 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. The full-text documents associated with the most similar cases–the RF-CKB–are passed to the IR system. The IR system creates a document-level query, poses it, and retrieves a set of documents. The top-rated documents for the *Rasmussen* situation are listed in Table 1. This completes SPIRE's stage one.

Rank	Case Name	Belief Score	Doc-Id
1	In re Sellers	(0.490157)	14180
2	In re San Miguel	(0.483656)	14289
3	In re Chura	(0.482781)	14188
4	In re LeMaire 1990	(0.479262)	14860
5	In re LeMaire 1989	(0.479195)	14751

Table 1: The most highly ranked documents for the *Rasmussen* problem.

We would like to examine specific facts in these newly retrieved cases, in particular, finding out how long other repayment plans were. (Other features of bankruptcy cases are discussed in Section 4.1.) To do this, we direct SPIRE in stage two to locate passages within the top case texts that concern the feature called *duration*. SPIRE uses excerpts from its case-base of excerpts on *duration* to form a query to retrieve passages. Sample excerpts from this case-base 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, 33 to 37. 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.

SPIRE's case-base for this particular feature contains 14 excerpts collected from 13 opinions. Combined, they contain a total of 212 words, 92 unique terms after stemming, and 59 unique terms when stop words are removed.

The top-rated document for the *Rasmussen* problem is the *In re Sellers* case, so we use it to illustrate passage retrieval. The IR engine divides the *Sellers* opinion 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). SPIRE then generates a query to be run against the *Sellers* opinion, divided into these windows. INQUERY carries this out and ranks the passages according to its belief that each is relevant to the query.

For this example, we allow SPIRE to use two simple methods to generate queries. The first combines the terms from all the excerpts about a feature into a single "natural language" query. Each word in each excerpt provides a possible match against the words in the window. Regardless of whether two words were in different excerpts, each contributes to the total belief. We refer to this type of query as a *bag of words* query. The second type of query places a restriction so that terms from within an excerpt need to be found co-occurring in the passage. We refer to this type of query as the *sum* query because it is formed by wrapping an INQUERY `#Sum` operator around each excerpt. Part of both queries 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,
  ...)
#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 INQUERY to retrieve many relevant passages. Below are the top five passages for each query:

Bag of Words

Rank	Psg Strt	Belief	
1	1440	(0.404378)	Rel
2	1430	(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)	
5	1420	(0.402145)	Rel

Figure 2 gives the text of the 1440 passage, top-ranked in both retrievals. We boldface content terms that match those found in the excerpts and show word counts along with the text. (We have included and highlighted terms from the passage beginning at 1430 since it is ranked second by both queries as well as the 1420 passage as it is ranked fifth by both queries.)

From the 1440 passage we can determine that the debtor proposed a 36-month plan. (This same information can be learned from the 1430 passage, ranked second.) We can 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 3. (We include enough text to cover passage 2660, as it ranked fourth with the *bag of words* query and ninth with the *sum* query.) These passages talk to the duration of a plan that the judge is summarizing.

The fifth-ranked passage for both queries, 1420, gives introduction to the length of the plan. By looking at the next several words following the passage the reader can determine the *duration* of the plan. (Passage 1420 is given in Figure 2.)

For the *sum* query, the fourth ranked passage, 2460, is not relevant although it contains many terms in common with the excerpts for *duration*. It discusses the amount of the payments, rather than the duration. (Passage 2460 is shown in Figure 4.)

In stage two, SPIRE has thus located passages relevant to the *duration* feature without requiring a user to pose a query. Unlike other approaches, which merely retrieve entire documents, SPIRE is able to retrieve documents and then present a significantly reduced amount of text about features contained within the document. This greatly decreases the amount of text a user must inspect for information.

4 Methodology

We now describe the various types of features, generation of answer keys, and the evaluation metric.

4.1 Features examined

We selected ten features from a bankruptcy good faith case representation. There were five types of values that these features could have: Boolean, date, category, set, or numeric. For our set of ten features, we included two of each type. They were: *sincerity* (was the debtor sincere in proposing the plan), *special-circumstances* (were there any extenuating conditions affecting the debtor), *loan-due-date*, *plan-filing-date*, *procedural-status* (such as appeal or affirmation), *likelihood-income-increase* (of the debtor), *debt-type* (such as educational or consumer), *profession*,

...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 2: 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 3: Passages 2650 and 2660.

monthly-income, and *duration* (of the proposed plan in months).

For this set of ten features we gathered excerpts from 13 opinions. Once SPIRE stemmed and removed non-content terms, the average number of remaining unique content terms for the ten features was 46.8.

We have already shown some of the excerpt case-base for the feature of *duration*. Below we give descriptions, similar to those given to the outside readers (see Section 4.2), of two additional features and examples of the excerpts we collected for these features.

Future income – this is text that discusses whether the debtor's income is projected to increase in the future. The text might be negative or positive on this matter.

- "the Court cannot see any likelihood of future increases"
- "the prospect of a regular job with substantially increased income is not great. "
- "her health brings into question her future ability to work."
- "no evidence that raises are likely."

Sincerity – This is text that mentions whether or not the debtor(s) really intended to pay off their debts, is making an honest effort, or isn't trying to deceive the court. The text may indicate that the debtor might or might not have been *sincere.*

- "represents an earnest effort to repay his unsecured creditors"
- "sincerity is tempered by her desire to avoid returning to Maine."
- "The Court believes the debtors' motivation and sincerity are genuine."
- "The Chapter 13 petition was intended to wipe out BNY's claims rather than to repay them."
- "this couple makes a concerted effort to live sensibly and substantially within their means."

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. These answer keys were created by outside readers.

We hired two undergraduates to read case opinions and underline any text that they perceived as being about a given feature. They were given a set of written instructions that described each feature and were provided samples of the sort of text they should mark.

4.3 Evaluation metric

Most 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. However, in our scenario we are not concerned with locating every relevant item. Rather, we are concerned with how much non-relevant data a user must go through before finding some number of relevant items. This value can be measured by what is called *expected search length* (ESL)[2]. *Expected search length* measures the number of false hits a user or system would have to read through or process before finding a specified number of relevant items.

5 Experiment Results

We ran SPIRE using three problem cases and collected the top documents for each. Removing duplicates and documents that had been used to derive

. . . 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 4: Passage 2460.

the excerpt case-bases, we made a test collection from among the top 10 retrievals for each problem to make a test set of 20 documents. Using various methods for passage query generation, we tested SPIRE on these 20 documents with 10 different case features. We report values when one, three, and five relevant passages were requested for computing esl scores. We denote these as esl_1, esl_3, and esl_5.

5.1 Query types

In the experiments reported here, we are concerned primarily with the two previously mentioned query formation methods: *bag of words* and *sum*. These came from a set of five *base* methods. The others in this set are: *bag of words plus phrases, sum plus phrases*, and *set of words*. Formation and results for these queries is discussed in more detail in [3].

We had SPIRE build two other sets of queries. The first is based on a term weighting scheme suggested by Kwok [5] and the second set is what we called *semi-random*. The latter incorporated only one-half or one-third of the available query terms from the excerpt case-base. Neither of these sets performed better than the two base queries. (See [3] for details.)

To provide another point of comparison, we also had a human expert, familiar with both the domain and INQUERY query operators, create a set of sophisticated queries. The manual queries were refined over time and may use many more types of operators than the SPIRE-generated queries. We used the best manual query for each feature as a upper guideline as to how well we can hope to do. We refer to this set as the *manual* queries.

5.2 Results

Among the set of base queries, the first two—the *bag of words* and *sum* queries—generally outperformed all of the others. They also outperformed the Kwok-based and semi-random queries, and we therefore, restrict discussion to comparison between these two types of query and the manual ones.

Comparison of the *bag of words* and *sum* queries reveals that they performed about equally well as measured by esl scores. Across all 20 documents and 10 features, the *sum* queries performed slightly better when requesting one or five relevant passages, and the *bag of words* queries performed slightly better when requesting three passages.

Table 2 provides a comparison between the manual and two SPIRE-based queries on half of the test document collection; results on the other half are similar. Overall, SPIRE-generate queries performed just about equally to the expert manual queries.

Across all 20 documents, 10 features, and the 3 esl levels, the *bag of words* and *sum* queries do better than the manual queries. However, when we look at the results broken down by feature, there are noticeable differences. There are two features where the manual queries do better: *procedural status* and *plan filed date*, and two features where the SPIRE-based queries do distinctly better: *sincerity* and *loan due date*. With the other features, the results were closer.

For *procedural status* this difference is easily explained: discussion about the feature will normally include at least one of a small set of easily enumerated keywords, such as "confirmation" and "appeal". Not all of these terms were present in SPIRE's excerpt case-base, but all were included in the manual query. For example, "affirmation", "remand", and "convert" were never given as the status of any of the cases found in our small corpus. This is an instance where knowledge of a domain-specific vocabulary, particularly one of limited terms, is easily enumerated and should be used as the basis for forming queries on a feature.

The difficulty SPIRE had in finding the *plan filed date* is partially due to the way in which the opinions express the date. Below are some of the relevant passages:

- "At the time of filing the Chapter 13 proceeding," [case opinion 289]
- "LeMaire signed a promissory note evidencing a debt to his parents of $12,722 only one day prior to filing his bankruptcy petition. Prior to this filing, LeMaire had . . . " [case opinion 860]

In neither case is a calendar date given, and additionally, the first text fragment is the only relevant passage within that text. We note that pattern matching techniques, or concept recognizers would also be unable to locate these passages.

In summary, SPIRE did well on most features. Where there exist short, domain-specific lists, partic-

Doc-ID	Debt Type	Duration	Future Income	Loan Due	Mthly income	Plan Filed	Proc. Status	Profes-sion	Sincere	Special Circ
001	=	M	=	=	=	SP	M	SP	M	s
180	M	SP	=	M	=	M	M	=	=	s
188	s	M	M	SP	SP	M	M	=	SP	s
204	SP	SP	=	SP	SP	M	SP	M	SP	SP
206	M	M	M	=	SP	SP	=	SP	SP	=
260	M	SP	SP	=	SP	M	M	=	=	b
289	=	M	=	=	=	M	M	M	SP	SP
353	=	M	=	SP	=	=	s	SP	=	=
407	SP	=	=	=	s	M	b	b	=	SP
427	=	M	M	=	SP	SP	M	=	=	s

Table 2: Comparison between the esl_3 of manual and SPIRE-generated queries. An "SP" indicates that both SPIRE queries performed better than the manual. An "M" indicates that the manual query performed better. If the manual fell between the two, the SPIRE query performing the best is given: "b" for *bag of words* and "s" for *sum*. If all three queries performed equally well, an "=" is shown.

ularly those of technical keywords, it is better to manually compose queries. Finding passages that discuss relevant dates without giving a calendar date are currently hard to find with either SPIRE's or a manual approach.

6 Conclusion

In this paper, we have presented the SPIRE system that, given a problem case, first retrieves relevant documents from a full-text collection, and then within each document locates relevant passages discussing case features. The queries needed for both the document and passage retrievals are generated automatically by SPIRE in a case-based manner.

We examined the performance of SPIRE at retrieving relevant passages using the measure of expected search length. We found that SPIRE does as well or better than manually crafted passage queries for most case features we tested.

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 locate new passages. Context in this limited form enables SPIRE to do quite well. SPIRE achieves its results robustly and with minimal effort–only problem entry–on the part of the user.

References

[1] 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.

[2] William S. Cooper. Expected Search Length: A Single Measure of Retrieval Effectiveness Based on the Weak Ordering Action of Retrieval Systems. *American Documentation*, 19:30–41, 1968.

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

[4] 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.

[5] K. L. Kwok. A New Method of Weighting Query Terms for Ad-Hoc Retrieval. In *Proceedings of the 19th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pages 187–195, Zurich, Switzerland, August 1996. ACM.

[6] Edwina L. Rissland and Jody J. Daniels. The Synergistic Application of CBR to IR. *Artificial Intelligence Review*, 10:441–475, 1996.