Finding Legally Relevant Passages in Case Opinions.*

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

Abstract

This paper presents a hybrid case-based reasoning (CBR) and information retrieval (IR) system, called SPIRE, that locates passages likely to contain information about legally relevant features of cases found in full-text court opinions. SPIRE uses an example base of excerpts from past opinions to form queries, which are run by the INQUERY IR text retrieval engine on individual case opinions. These opinions can be those found by SPIRE in a prior stage of processing, which also employs a hybrid CBR-IR approach to retrieve relevant texts from large document corpora. (This aspect of SPIRE was reported on at ICAIL95.)

We present an overview of SPIRE, run through an extended example, and give results comparing SPIRE's with human performance.

1 Introduction

There is an enormous amount of legal text available on-line and it is growing every day. While this is a decided benefit for legal research, it also presents a problem of how to search it effectively. In particular, it is no easy task to find the legally relevant nuggets of information buried in case opinions. Not only can opinions be long (and rambling), but information about a particular aspect of the case (e.g., a party's employment history or financial status) can be scattered. To complicate matters, there is often no set way a factual aspect is presented—even dates and case citations can be problematic—and the information presented is often not "complete" as presented, in the sense that inferences must be made to determine a definite "value." For instance, to determine the number of years a person has worked or the amount of surplus monthly income remaining after financial obligations are serviced can require the amalgamation of information from disparate parts of the opinion and inferencing from stated and common sense information.

Thus, there is a long train of information processing beginning at the location of relevant text corpora, retrieval of texts, location of pertinent passages within the texts, amalgamation of information from possibly wide distribution within the text, inferencing about what is said, assumed, and not said in the text, and then, of course, all the great richness of legal reasoning that follows, including case comparison, argument generation, memo or brief writing, client consultation, and court activities.

Past work in our lab has addressed several aspects of this legal information processing sequence, including (adversarial) precedent-based reasoning and argumentation in both common law [Ash90, RA87] and statutory domains [RS91], perusal and harvesting of information from symbolically represented legal information [RSF94, RSF96], and preparation of stereotypical memos and reports [RDRS93]. Recently, we have considered the use of case-based reasoning (CBR) to drive traditional full-text information retrieval (IR) engines, such as INQUERY [CCH92], to retrieve case opinions from large document corpora [RD95, DR95].

In this paper, we concentrate on the process of locating within individual opinions those passages likely to contain information about legally relevant aspects of a case. The opinions are presented as ordinary text. In other words, we address the problem of how to find textual passages that will allow values or fillers to be extracted and inferred for the features or slots used to represent a case. The problem we address is a prerequisite for breaking through the knowledge acquisition bottleneck since, before the full power (and expenses) of information extraction—person or program—can be harnessed, one must determine where to concentrate the effort. It is neither computationally feasible nor cognitively reasonable to apply an equal and high-level of effort across the entire text. Successful information extraction requires focus of attention. Thus, the passages highlighted by our process can be used as input to an extraction process whose output can then be plowed back into a case-base used by our system or some other symbolic reasoner, such as a CBR or instance-based learning program.

To solve the problem of passage location (and the prior problem of locating texts to examine for passages), 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 CBR-IR system that works in two stages:

1. from a large text collection, retrieves documents that

*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.

are relevant to the presented problem case, and
2. highlights within those retrieved documents passages that contain information relevant to specific case features.

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 the cases as the basis for generating queries, which are then run by INQUERY [CCH92] in the usual way. In the first stage, the query is run on the text collection; 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. Figure 1 gives an overview of the system.

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 the usual 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 [Ash90].

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 "best cases" is called the *relevance feedback case-knowledge-base* or RF-CKB [RD95, RD96]. 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. (A detailed description of this first stage can be found in [DR95, RD96].)

In the second stage, SPIRE locates germane passages within each of the texts retrieved in stage one. In this stage SPIRE locates passages within a document rather than documents within a collection. Again SPIRE uses a hybrid CBR-IR approach. This was motivated by our belief that past discussions of a topic would provide good clues to the location of new, related 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. (A fuller discussion of this can be found in [Dan97].) 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 generate queries. The difference is that (at this point in our development of SPIRE) there is no selection of terms from the 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 and/or terms from them 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 read a small number of opinions corresponding to cases in SPIRE's case-base and to highlight any portion of text—whether it be just a few terms, a phrase, or several sentences or more—that was useful for determining the feature's value. It was common for pieces from different locations throughout the text to be highlighted. Normally, this step would be done in conjunction with the creation of the case-base for the domain and the encoding of the first few cases and thus would not require a full review of the textual sources. However, since we were re-using a portion of the bankruptcy case-base used in the BankXX project [RSF96], this highlighting of textual examples was done *post hoc*.

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, and add it to the representation for the text, for instance, as a case. 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.

3 Example

Our example comes from the domain of personal bankruptcy under Chapter 13 of United States personal bankruptcy law (11 U.S.C. §1301-1330). The question presented is whether the plan proposed by the debtor to settle with the creditors has been proposed in "good faith". This question is central to approval of the debtor's plan.

We use the facts as found in the *In re Rasmussen*, 888 F.2d 1030 (6th Cir. 1988), 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 in the form of a case frame, which, using the CBR module, 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 ten top-rated documents for the *Rasmussen* situation are listed in Table 1. We note that only *Chura* and *Sellers* were already known to SPIRE (i.e., represented in its case-base of documents), although none of these opinions have text in the excerpt case-base. Thus, the other eight of the top ten cases must be "read" in order for their facts to be ascertained in preparation for any use in a legal analysis for *Rasmussen*.

We also note that both of the *LeMaire* cases occurred after SPIRE's case-base was created (and after *Rasmussen*).

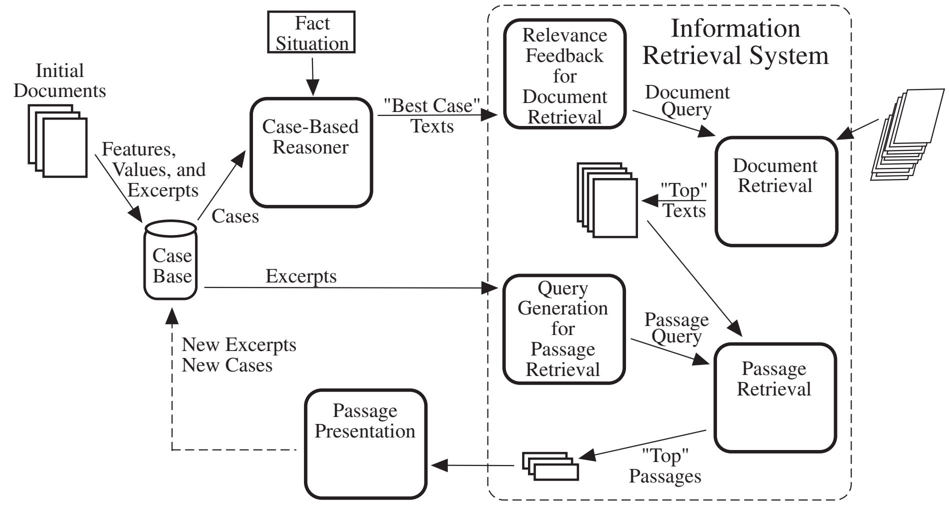

Figure 1: Overview of SPIRE.

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
6	In re Stewart	(0.479071)	14877
7	In re Chase	(0.477976)	14260
8	In re Lincoln	(0.475428)	14204
9	In re Nittler	(0.474340)	14407
10	In re Kazzaz	(0.474268)	14472

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

In a fielded version of SPIRE with a real problem case, of course only already litigated cases (having a published opinion) would be available for retrieval. However, the ability to retrieve cases from a text corpus allows a symbolic system, like the CBR submodule of SPIRE, to overcome some well-known limitations, like what we have called the "staleness problem," by allowing the system to access cases occurring after its case-base was created. Locating these documents completes SPIRE's stage one.

We would like to examine specific facts in these newly retrieved cases, such as, 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 the 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. The belief for each passage will be based on the single best-matching 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)
```

. . . 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.

```
#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, annotated with whether or not each is relevant:

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.

For comparison, suppose we intervene after SPIRE's first stage and manually generate a query for the topic of *duration*. A sophisticated query might look like:

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

The `#Passage20` operator tells the IR engine to retrieve passages with a window size of 20. The `#Phrase` and `#3` operators add even more 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 collection.)

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)	
4	2090	(0.410598)	
5	2080	(0.410598)	Rel
6	1990	(0.410148)	
7	1980	(0.410148)	
8	1940	(0.410148)	
9	1930	(0.410148)	
10	1430	(0.410148)	

The top two ranked passages do, in fact, contain information about the *duration* of the plan. We display them in Figure 5.

. . . 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 2660.

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 5: Passages 2610 and 2620.

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.

4 Domain Knowledge

We now describe the various types of features we examined, the case-bases of textual excerpts, 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), *future-income* (likelihood that there will be an increase in the debtor's income), *debt-type* (such as educational or consumer), *profession*, *monthly-income*, and *duration* (of the proposed plan in months).

4.2 Excerpts

For the above set of ten features we gathered excerpts from 13 case opinions. Once SPIRE stemmed and removed non-content terms, the average number of remaining unique content terms for the ten features was 46.7, although two of the features only have 18 content terms. Table 2 gives more information on the size of the excerpt case-base.

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 the Section 4.3), of three 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.

- "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."

Feature	Num Excerpts	Total Words	Unique Terms	Content Terms
Duration	14	212	92	59
Monthly Income	13	110	52	34
Debtor Sincerity	9	123	89	52
Special Circ.	8	188	117	71
Loan Due Date	4	47	32	18
Plan Filed Date	10	145	66	45
Debt Type	10	164	102	63
Profession	3	36	29	18
Future Income	8	88	68	36
Procedural Status	13	194	100	71

Table 2: Number of terms contained in the excerpts.

Special circumstances – these are any unusual events or factors in the debtor's life that may have lead to the bankruptcy or may affect their ability to repay debts. These may be things like: medical issues or expenses, moving to some location where the cost of living exceeded the debtor's expectations, being in prison, stress related problems, an inability to get a job, a pending divorce, etc.

- "The Court believes the debtors' medical expenses will increase as time goes on and believes this is a 'special circumstance' under factor 8."
- "This debtor has not been the victim of extraordinary 'outside' forces."
- "The debtor is now in treatment for the condition that may have contributed to the debtor's need for Chapter 13 relief."
- "She thereupon encountered some difficulties in her personal life. A medical condition forced her to leave work for a two-week stay in the hospital and her marital relationship began to deteriorate. She also claims to have suffered from a nervous condition during that time."
- "Debtor was incarcerated in the New Mexico State Penitentiary for fraudulent practices"

Loan due date – this is text that describes when a debtor should have ed start making payments on a loan. The text might not give a date, but instead, describe a time period when payment should or did commence. In all cases, there should be some reference to a date, whether explicit or implicit.

- "Repayment on the loan, after the expiration of the grace period, began on January 1, 1983, but Mr. Ali

was given an additional deferment until July 1, 1983."
- "loan which became due in March 1980."
- "became due one year after graduation."

4.3 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.

4.4 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)[Coo68]. *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. Simply, it is the amount of wasted effort.

5 Experimental results

We used SPIRE to retrieve relevant documents for 3 different problem cases. For each problem case we gathered the top set of documents and created a test set of 20 opinions. For each of these documents, we asked SPIRE to retrieve passages relevant to the 10 case features given in the previous section. We then computed esl values. We experimented with a dozen different methods by which SPIRE generated passage queries from the excerpts.

5.1 Query types

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 individual 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 selection tool over the excerpts. The fifth type of query was created by taking a *bag of words* type query and removing duplicate words.

We also tried another set of queries based on a weighting scheme suggested by Kwok [Kwo96]. To create these queries, SPIRE generated an initial query that included all the words found in all the excerpts. The terms were then given weights based on the Kwok scheme, which ranks terms in order of perceived importance. We investigated five types of Kwok-derived queries but found results with them to be not as strong as with the set of base queries. (See [Dan97] for details.)

We also investigated what we called *semi-random* queries. To create these, SPIRE randomly selected either one-half or one-third of the available query terms from the excerpt case-base. Each query term was considered equally likely (i.e., excerpt frequency was not used) and available only once (i.e. selection without replacement). These queries addressed our concern that there might be too many terms available, which might allow too many passages to receive high belief scores. Again, the results were not as good as with the base queries, and details can be found in [Dan97].

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 queries. These manual queries were refined over time and use many more types of operators than the SPIRE-generated queries. Further, the expert generating these queries had access to the excerpt case-base as well as many of the solution passages. We used the best manual query for each feature as a baseline. We refer to this set as the *manual* queries.

5.2 Results

We ran SPIRE using the various methods for passage query generation over three problem cases, with their sets of top 10 documents, and 10 case features. (Removing duplicates and documents already known to the system, we created a collection of 20 test documents.) We report values when one, three, and five relevant passages were requested for computing esl scores.

Comparison of the first two types of base queries—the *bag of words* and *sum* queries—reveals that they performed about equally 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.

Among the set of base queries, the first two generally outperformed the second two. This indicates that phrases were detrimental, rather than beneficial in our application. This is surprising since phrases typically increase retrieval performance [CTL91].

Closer examination of the SPIRE-generated phrases offers an explanation for this. In situations where phrases have proved beneficial, (for full-document retrieval), the automatically generated phrases were were noun phrases. Whereas, the phrases more closely associated with discussion of some of the case features (e.g., "payments of" and "proposes to pay" for *duration*) are not noun phrases and are not easily automatically generated.

Our next result is that term frequency information, when given in a query, is important. The *bag of words* queries achieved results significantly better than the *set of words* queries. This indicates that if a term occurs in multiple excerpts, it should be given additional weight in a passage query.

In summary, among the SPIRE-generated queries, *bag of words* and *sum* queries performed about equally well, and both of these types of query performed better than both the Kwok-derived and semi-random ones. They also outperformed the queries that used phrases. Allowing multiple occurrences of a term (*bag of words*) does better than re-

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
442	SP	=	=	=	M	M	M	M	SP	M
472	s	SP	=	=	M	b	M	SP	=	b
693	=	M	M	M	s	SP	M	M	SP	b
733	=	b	=	b	=	M	M	=	SP	=
751	SP	s	SP	SP	=	SP	M	s	=	b
764	SP	SP	M	SP	M	SP	M	SP	SP	=
860	=	b	=	=	=	M	M	M	=	M
877	=	SP	=	SP	=	M	M	=	SP	=
915	b	SP	=	=	SP	M	M	M	=	SP
961	=	M	SP	=	=	SP	M	SP	=	=

Table 3: Comparison between the esl_3 of manual and SPIRE-generated queries.

stricting a word to appear only once in the query (*set of words*).

Lastly, we compared the *bag of words* and *sum* queries with the manually-generated queries. When requesting one and three relevant passages, the manual queries do a bit better than the SPIRE-generated ones. When five relevant passages are requested, the two excerpt-based queries are slightly better. There are several noticeable differences between these queries. The first is that SPIRE has trouble with the features of *date plan filed* and *procedural status*, while the manual queries have trouble with the features of *loan due date* and *special circumstances*. The other major difference is that there were a number of documents and features where the manual queries were unable to retrieve the requested number of relevant passages, while the SPIRE-generated queries were able to do so.

Table 3 lists all the features and the 20 documents from the test collection. It compares the esl when three relevant passages were requested. An "SP" is given when both excerpt-based SPIRE queries perform better than the manual. When the manual query performs better, an "M" is listed. If the manual fell between the two, the SPIRE query performing the best is given: "b" for *bag of words* and "s" for *sum*. Finally, if all three queries performed equally well, an "=" is shown.

6 Conclusion

We have presented the SPIRE system, which incorporates a two-stage approach to first, retrieve documents relevant to a given problem situation and second, concentrate attention in onto passages within these documents that discuss particular legal aspects of the case. SPIRE automatically generates the queries needed for both of these stages in a case-based manner.

Our prime concern in these retrievals is minimizing the amount of effort expended by the user—whether human or machine—in locating important pieces of information. We, therefore, examined SPIRE's performance at retrieving relevant passages by using a measure that evaluated the amount of wasted effort, expected search length. We found that SPIRE does as well or better than manually crafted passage queries for many of the case features we tested.

SPIRE also has the potential for ameliorating the knowledge acquisition bottleneck in a way particularly useful in law. By pointing out the passages in full-text documents where valuable information is likely to be found, SPIRE can help a legal reasoner locate key bits of information quickly and automatically. When the documents are long, as many opinions are, and time, short, and costs of expert labor, high, this is a potentially significant benefit.

References

[Ash90] Kevin D. Ashley. *Modeling Legal Argument: Reasoning with Cases and Hypotheticals.* M.I.T. Press, Cambridge, MA, 1990.

[CCH92] 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.

[Coo68] 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.

[CTL91] W. Bruce Croft, Howard R. Turtle, and David D. Lewis. The Use of Phrases and Structured Queries in Information Retrieval. In *Proceedings of the 14th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pages 32–45, Chicago, IL, October 1991. ACM.

[Dan97] Jody J. Daniels. *Retreival of Passages for Information Reduction.* PhD thesis, University of Massachusetts, Amherst, Amherst, MA, May 1997.

[DR95] 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.

[Kwo96] 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.

[RA87] Edwina L. Rissland and Kevin D. Ashley. A Case-Based System for Trade Secrets Law. In *Proceedings, 1st International Conference on Artificial Intelligence an Law*. ACM, ACM Press, May 1987.

[RD95] Edwina L. Rissland and Jody J. Daniels. A Hybrid CBR-IR Approach to Legal Information Retrieval. In *Proceedings, 5th International Conference on Artificial Intelligence an Law*, pages 52–61, College Park, MD, May 1995. ACM.

[RD96] Edwina L. Rissland and Jody J. Daniels. The Synergistic Application of CBR to IR. *Artificial Intelligence Review*, 10:441–475, 1996.

[RDRS93] E. L. Rissland, J. J. Daniels, Z. B. Rubinstein, and D. B. Skalak. Diagnostic Case Retrieval Guided by Evaluation and Feedback. In *Proceedings of the CBR Workshop at the 11th National Conference on Artificial Intelligence*, pages 135–140, Washington D.C., July 1993. AAAI.

[RS91] Edwina L. Rissland and David B. Skalak. CABARET: Rule Interpretation in a Hybrid Architecture. *International Journal of Man-Machine Studies*, 34:839–887, 1991.

[RSF94] Edwina L. Rissland, D. B. Skalak, and M. Timur Friedman. Heuristic Harvesting of Information for Case-Based Argument. In *Proceedings, The 12th National Conference on Artificial Intelligence*, pages 36–43, Seattle, WA, August 1994. AAAI.

[RSF96] Edwina L. Rissland, D. B. Skalak, and M. Timur Friedman. *BankXX: Supporting Legal Arguments through Heuristic Retrieval*. *Artificial Intelligence Review*, 10(1-71), 1996.