From Cases to Documents to Passages.*

Jody J. Daniels
Lockheed Martin
Advanced Technologies Laboratory
1 Federal Street
Camden, NJ 08002 USA
Email: jdaniels@atl.lmco.com

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 database objects, thus working on closing the loop in the problem-solving—knowledge-acquisition cycle.

Introduction

In order for case-based reasoning (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 much of a problem since the representations are either fairly simple (Golding & Rosenbloom 1991; Lehnert 1987b), or can be easily automatically generated (Lehnert 1987a; Veloso 1992). However, when the domain representation for a case is complex, building a case knowledge base with large numbers of cases typically becomes an expensive proposition. Converting case information into a symbolic 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 in a domain with long textual documents, we wanted 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 case, 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 approach by building a hybrid CBR and information retrieval (IR) system, SPIRE, to first locate relevant documents and then within these documents locate passages. The passages retrieved were those that described the important features for the domain. The features were those that make up the domain representation used by the CBR system in the form of feature-value pairs. We validated this integrated approach by using texts drawn from a statutory legal domain.

We next present the architecture of the system and walk through an extended example. We then discuss future directions and conclude in the final section.

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. The second stage relies on a case-base of excerpts to locate passages within the newly retrieved documents. This integrated approach is motivated by our belief that past discussions provide good clues to finding new documents as well as assist in locating relevant passages within the new documents. The CBR component is a HYPO-style CBR system (Ashley 1990) and the IR system is built upon INQUERY (Callan, Croft, & Harding 1992).

In the first stage, SPIRE is given a new problem or fact situation. The facts are input into a case-frame representation. We assume that the representation was designed by domain experts based on their expertise, knowledge of the domain, and understanding of the task at hand. This represen-

*This research was supported while the 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.

tation is exploited by the case-based reasoner to perform the desired type of reasoning.

SPIRE uses its CBR module to analyze the situation and select a small number of most relevant cases. These cases come from the reasoner's CKB. In standard CBR fashion, SPIRE determines the similarity of each known case to the new problem, sorts the relevant known cases according to their degree of on-pointness, 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 "jump-start" INQUERY's relevance feedback module. This set of "best" cases is called the *relevance feedback case-knowledge-base* or RF-CKB. The original text of the cases in the RF-CKB (i.e., the opinions) are passed to the IR engine. The IR engine then treats these documents as though they had been marked relevant by a user.

Using a modified form of relevance feedback, (we start with an empty query and generate one, rather than modify or expand an existing query,) the IR system generates a query by selecting and weighting terms or pairs of terms from within the RF-CKB. This query is then run against the larger corpus of texts, with the result that documents (court opinions) are retrieved and ranked. Figure 1 gives an overview of this process. (More details on this process can be found in (Daniels & Rissland 1995).)

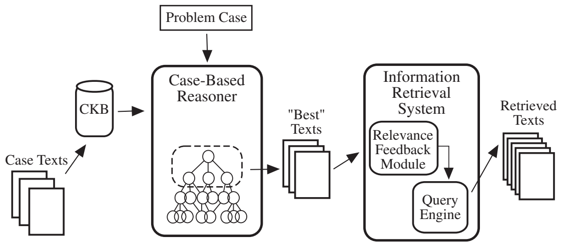

Figure 1: SPIRE's retrieval process for novel documents.

In the second stage, SPIRE locates germane passages within each of the texts retrieved in stage one. There is no real limit on how many of the texts can be examined in stage two, however, we generally examined only the top ten. If we were able to set a threshold that would distinguish between texts that were relevant and those that were not, we would process additional documents until that threshold was reached.

To now locate the text segments that discuss a particular feature, SPIRE once again uses a hybrid CBR-IR approach but this time the task is to locate passages (within a document) rather than documents (within a collection). To locate these passages, SPIRE generates queries that express the information need associated with the particular feature. To do this, SPIRE uses information from past discussions of a feature.

For each case feature of interest, SPIRE has a case-base of textual excerpts, called the *excerpt-ckb*. Each piece of text in the excerpt-ckb comes from an episode of information location/extraction performed on a past case. Using the excerpt-ckb SPIRE generates a feature-level query to be run on each document. Figure 2 gives an overview of the passage retrieval process.

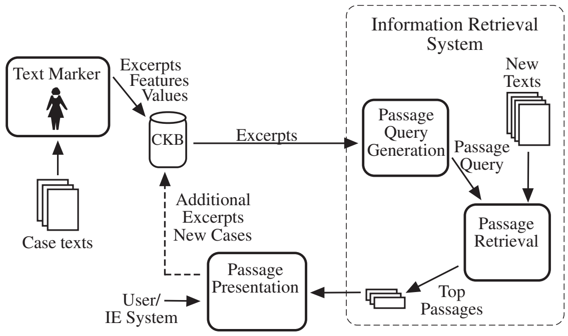

Figure 2: SPIRE's passage retrieval subsystem.

SPIRE presents the passage 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). The IR engine then retrieves the top-ranked passages for presentation to the user (or possibly to an information extraction system). (More details on this process and experimental results can be found in (Daniels 1997).)

Thus, the 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 example excerpts in the opinions corresponding to a small number of the cases in SPIRE's case-base.[1] Typically this will only be a few documents, on the order of 10–15.

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

[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.

may 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. These new excerpts may be used to form later passage queries. This way, SPIRE may aid in the acquisition of additional knowledge about the context of each feature. Once the new texts have been converted and added to the CKB, the CBR component may reason about them relative to the current problem or fact situation and the original task.

Example

To better illustrate the approach we used in SPIRE, we run through the following scenario based on a case from Chapter 13 of United States personal bankruptcy law (11 U.S.C. §1301-1330). In this case 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 the 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 case opinions.

The next step is to examine these newly retrieved case opinions for specific facts, in particular, the length of other 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. Below are sample excerpts from the *duration* excerpt-ckb:

- "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."

The top-rated newly retrieved document for this problem case is the *Sellers*[2] case opinion, 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 query 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 query.

[2] In re Sellers, 33 B.R. 854, 857 (Bankr. D. Colo. 1983)

For this example, we allow SPIRE to use a simple method to generate queries: it 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. Part of this query for *duration* is 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,...)
```

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

Bag of Words		
Rank	Psg Strt	Belief
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

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

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

The third ranked passage is 2650, displayed in Figure 4. (We include enough text to cover passage 2660, as it ranked fourth.) These passages address the *duration* of a repayment plan in a related case that the judge is summarizing.

In stage two, SPIRE has located passages in an individual document relevant to the *duration* feature without requiring a user to pose a query. SPIRE can do this for any feature for which there is an excerpt-ckb, and on any individual document.

Discussion

This approach, as executed by SPIRE, locates relevant documents and then passages, but these passages are consecutive windows and are not based on thematic or discourse units. This method is fine for indexing and retrieval, but presentation to a user may be confusing. Additionally, we have yet to take advantage of information based on the structure of the excerpt during retrieval, nor have we incorporated any domain-specific knowledge into the process. We briefly discuss three approaches to refining the query formulation process that would help

. . . 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.

enhance retrieval accuracy while minimizing user effort: query expansion, learning, and concept recognizers.

Query expansion

We currently only use information gleaned from the excerpts to form our queries. We have not yet explored taking advantage of information found across the entire collection of documents to find terms related to those in the excerpts.

One possible technique is to apply query expansion. We could use a thesaurus to add new terms. Alternatively, we could further refine our expansion operation to take advantage of domain knowledge by building an association thesaurus (Jing & Croft 1994). We have several available databases from which we could build our association thesaurus: the entire collection, or just those documents related to the problem case as found in the CKB. Another possibility would be to build the thesaurus from an intermediate collection, such as that comprised of the documents retrieved by SPIRE with the document-level query. Thesauri built from the second and third collections would be highly domain- and problem-specific.

Learning

Learning techniques have been successfully applied to IR tasks such as routing and filtering (Harman 1994; 1995). Learning techniques could be used to decide which excerpts or even portions of excerpts should be included in a passage query. Unfortunately, there are several drawbacks to this approach. The most prominent is the need for additional training data. We do not want to burden the user with the task of generating large amounts of training data. Further, we do not want to "overtrain" our queries such that they work exceptionally well on the training texts, yet are too tailored to do well over other documents.

We have not yet learned when it is appropriate to add new excerpts to the case-base. Our results have shown the need to keep information about terms that appear multiple times in the excerpt-ckb (Daniels 1997). However, we need to be careful that we do not add too many copies of closely related excerpts such that some terms dominate and overwhelm the retrievals. We want to ensure that we have coverage of the most typical means of expressing the feature, yet allow for the retrieval of possible exceptions.

Concept recognizers

Pattern matching techniques that can rely on regular expressions or an inclusive listing of the set members could be quite beneficial for some of the features. For instance, if we were to use a concept recognizer that assigned credit to passages containing a date, rather than giving credit for each individual term, retrievals would likely benefit because of fewer spurious matches.

Concept recognizers that might prove useful are monetary amounts, dates, and time periods. Specific dollar amounts are present in many of our features and actual values are frequently in the text. Example features with monetary amounts are *monthly payments* (toward either a bankruptcy plan or a loan), *monthly income*, *monthly expenses*, *amount of the debt*, etc. A monetary concept recognizer would simply scan for instances of a money symbol followed by a numeric value, or search for the terms associated with currency such as "dollar", "pound", etc., along with a value.

The concept of time periods, such as weekly, monthly, annually, and yearly, likewise show up in our domains with some regularity. Besides the descriptors just mentioned, the phrases "per week", "per month", and "per year" would additionally be a part of the time period concept.

In a related vein, posing queries that request matches to an open-ended concept could provide some assistance in better locating text that contains the value of a feature. Being able to request a match that includes any instance of a specified concept would be better than trying to enumerate the myriad ways of giving a value in the query. Requesting a match on any instance of the concept rather than on each individual term would be a big gain.

Conclusion

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.

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 the excerpt-ckb queries to retrieve many relevant passages. The next step is to move from the passages back into case-frames.

References

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

Callan, J. P.; Croft, W. B.; and Harding, S. M. 1992. The INQUERY Retrieval System. In Tjoa, A. M., and Ramos, I., eds., *Database and Expert Systems Applications: Proceedings of the International Conference in Valencia, Spain*, 78–83. Valencia, Spain: Springer Verlag, NY.

Daniels, J. J., and Rissland, E. L. 1995. 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*, 238–245. Seattle, WA: ACM.

Daniels, J. J. 1997. *Retreival of Passages for Information Reduction.* Ph.D. Dissertation, University of Massachusetts, Amherst, Amherst, MA.

Golding, A. R., and Rosenbloom, P. S. 1991. Improving Rule-Based Systems Through Case-Based Reasoning. In *Proceedings, Ninth International Conference on Artificial Intelligence*, volume 1, 22–27. Anaheim, CA: AAAI.

Harman, D. K., ed. 1994. *The Second Text REtrieval Conference (TREC-2).* National Institute of Standards and Technology, Gaithersburg, MD. Special Publication 500-215.

Harman, D. K., ed. 1995. *The Third Text REtrieval Conference (TREC-3).* National Institute of Standards and Technology, Gaithersburg, MD. Special Publication 500-225.

Jing, Y., and Croft, W. B. 1994. An Association Thesaurus for Information Retrieval. In *Intelligent Multimedia Information Retrieval Systems and Management, RIAO '94*, 146–160.

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

Lehnert, W. G. 1987b. Case-Based Reasoning as a Paradigm for Heuristic Search. Technical report, University of Massachusetts at Amherst, Amherst, MA.

Veloso, M. M. 1992. *Learning by Analogical Reasoning in General Problem Solving.* Ph.D. Dissertation, Carnegie Mellon University, Pittsburgh, PA.