Evaluating BankXX: Heuristic Harvesting of Information for Case-Based Argument

Edwina L. Rissland, David B. Skalak, and M. Timur Friedman
Department of Computer Science
University of Massachusetts
Amherst, Massachusetts 01003
{rissland, skalak, friedman}@cs.umass.edu

Abstract

The BankXX system models the process of perusing and gathering information for argument as a heuristic best-first search for relevant cases, theories, and other domain-specific information. As BankXX searches its heterogeneous and highly interconnected network of domain knowledge, information is incrementally analyzed and amalgamated into a dozen desirable ingredients for argument (called *argument pieces*), such as citations to cases, applications of legal theories, and references to prototypical factual scenarios. At the conclusion of the search, BankXX outputs the set of argument pieces filled with harvested material relevant to the input problem situation.

This research explores the appropriateness of the search paradigm as a framework for harvesting and mining information needed to make legal arguments. In this paper, we discuss how we tackled the problem of evaluation of BankXX from both the case-based reasoning (CBR) and task-performance perspectives. In particular, we discuss how various system parameters—start node, evaluation function, resource limit—affected BankXX from the CBR perspective and how well BankXX performs its assigned task of gathering information useful for legal argumentation by running BankXX on real legal cases and comparing its output with the published court opinions for those cases.

1. Introduction

In this paper, we present our evaluation of how well the BankXX program [Rissland et al., 1993] performs on its assigned task of harvesting information useful for legal argumentation. We run BankXX on real legal cases and compare its output with the published court opinions for those cases. We also evaluate how various internal parameters affect BankXX's performance as a CBR program. Finally we note that although the context of our research is legal argument, we believe that the use of such heuristic retrieval methods is applicable in other areas, such as diagnosis and design, where CBR methods have been classically applied.

This work was supported in part by the Air Force Office of Scientific Research under contract 90-0359.

This workshop submission is extracted from a AAAI-94 conference paper [Rissland et al., 1994]. The full version contains the same experimental results, but also provides a discussion of related work, an introduction to BankXX's retrieval design, and a more extensive bibliography.

2. The BankXX Experiments

In this paper we report on two types of empirical evaluations:

1. comparing the performance of BankXX with itself as a CBR program, by varying parameter settings; and

2. comparing the performance of BankXX with hand-coded arguments found in opinions of actual court cases.

In other experiments, we further explore BankXX's performance.

2.1 Methodology

The methodology for the first set of experiments is straightforward: run BankXX on each of the 54 cases in its case base in a *de novo* manner—that is, excise the case and all its linkages from BankXX's case-domain-graph—and count the number of items filling each of 10 argument pieces.[1] To compare BankXX with written case opinions, we encoded the 54 opinions into "answer" keys comparable in form to those generated by BankXX and applied standard precision and recall measures.

Precision is the ratio of what was mentioned by both the decision and BankXX to that mentioned just by BankXX. **Recall** is the ratio of what was mentioned by both the decision and BankXX to that mentioned just by the decision. We hasten to add that given the small numbers used in these experiments, these measures are very sensitive to small changes. For instance, for a given argument piece, if BankXX retrieves one item that is one of only two items mentioned in the opinion, its precision is 100% and recall is 50%. Should BankXX retrieve an "extra" item not mentioned in the opinion, its precision will drop to 50%; two extra items drop precision to 33%. Its recall will not increase. Since BankXX diligently harvests

[1] There are 10 terms whereas there are 12 argument pieces because the factor analysis argument piece is filled during system initialization, and we do not use the family resemblance prototype argument piece in these experiments.

as much information as it can, it is likely to mention more items than the opinion and be penalized for it in precision and not get credit for it in recall. Thus, one should be careful in reading too much into these traditional metrics. Nonetheless, given their widespread use, we do use them here.

Creating the "answers" needed for precision-recall comparisons was done by reading the court's opinion and encoding each case and theory actually cited in the opinion. One problem inherent in encoding written opinions with the set of original argument pieces is how to identify elements fitting each argument piece, since some have technical BankXX meanings (e.g., best case) or make fine distinctions hard for human readers to discern (e.g., applicable versus nearly applicable legal theory, best versus merely supporting cases). In BankXX, these distinctions are made in a principled way with computational definitions. To compensate for such difficulties, the argument pieces were aggregated into four larger-grained argument pieces that were easy to apply.[2] These were then used in hand-coding court opinions and as the basis of BankXX versus actual court performance comparisons. The four simplified argument pieces are: (1) **Cited-Supporting-Cases**,[3] (2) **Cited-Contrary-Cases**,[4] (3) **Cited-Leading-Cases**, and (4) **Cited-Legal-Theories**.[5]

With these aggregated argument pieces, hand-coding was straightforward and involved little subjective judgment. Any case cited in the opinion is listed as a **cited-supporting** case or a **cited-contrary** case depending on how its outcome compares with decision in the opinion.[6] If a cited case is also one that is frequently cited by written opinions in general,[7] it is also listed as a **cited-leading** case. If an opinion explicitly articulates a theory of its own, reiterates or applies the theory of another case, or appeals to a general domain theory (e.g., a "totality of the facts and circumstances" theory of good faith), then that theory is encoded as a **cited-legal-theory**.

[2]Note five argument pieces are not used in the aggregated argument pieces: supporting-citations, factor-analysis, overlapping-cases, factual-prototype-category, family-resemblance-prototype.

[3]Defined for BankXX as the union of supporting-cases and best-supporting-cases.

[4]Defined for BankXX as the union of contrary-cases and best-contrary-cases.

[5]Defined for BankXX as the union of applicable-legal-theories and nearly-applicable-legal-theories.

[6]Complications, such as the fact that a same side case may have been cited (with a so-called *But see* citation signal) in order to differ with its rationale while still agreeing with its outcome, are overlooked.

[7]A frequency analysis was done on a corpus of cases of approximately 800 cases gathered with a WestLaw retrieval. We then checked the citation frequency of each of BankXX's cases in this larger corpus. The five most frequently cited cases were used to define cited-leading-case category applied to written opinions. By contrast, for BankXX leading-cases is defined with respect to frequency of citation within BankXX's own corpus.

Output from these BankXX-court comparison runs can be viewed in various ways. **Figure 1** displays graphically the finest-grained analysis. It shows results for retrieval of objects for the aggregated **cited-leading-cases** argument piece for each of the 54 cases. Each bar compares performance of BankXX with the court opinion on one case.

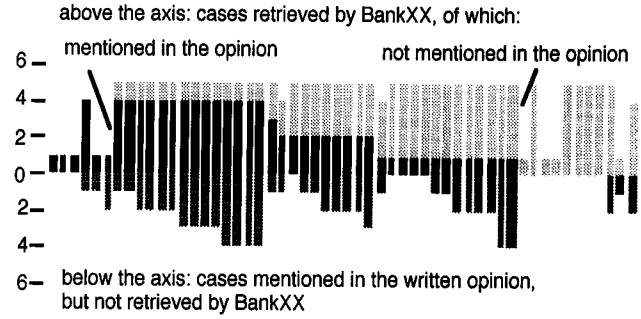

Figure 1: Comparison of retrieved *cited-leading-cases* using the argument piece evaluation function. Performance on each of the cases, in order from highest to lowest precision.

The vertical axis indicates the number of items retrieved. Everything above the zero represents items retrieved by BankXX with the black part of a bar representing those retrieved by BankXX and mentioned in the written opinion and the lightly shaded part of the bar representing items retrieved by BankXX that were not mentioned in the opinion. The darkly shaded part of the bar extending below zero represents items mentioned in the opinion that were not retrieved by BankXX. Graphically, **precision** is the proportion of black out of the total bar above the zero; **recall** is the proportion of black out of the combined black and darkly shaded parts of the bar.

In summary, we ran BankXX on each of the 54 cases in its case base in *de novo* fashion with each of two evaluation functions, and compared retrieval on each argument piece: approximately 1500 data points.[8]

2.2 BankXX as a CBR program

This section describes three experiments we performed to answer questions about BankXX as a case-based retrieval system:

1. How important is the initial query in determining the eventual outcome of retrieval?

2. How much knowledge must the case retrieval function have in order to be effective?

3. When can search terminate and the retrieval result be satisfactory?

As a baseline, BankXX was run with the *Estus* case, 695 F.2d. 311 (8th Cir. 1982), as start node, the argument piece evaluation function, and search limited to closing 30 nodes. We addressed the three questions above in search terms by examining the effects of:

[8]Given 10 argument pieces used in the general CBR experiments and 4 in the BankXX-Court comparisons, there are (2x10 + 2x4) x 54 data points.

1. varying the start node,
2. changing the evaluation function, and
3. trying different limits on the number of nodes that could be closed.

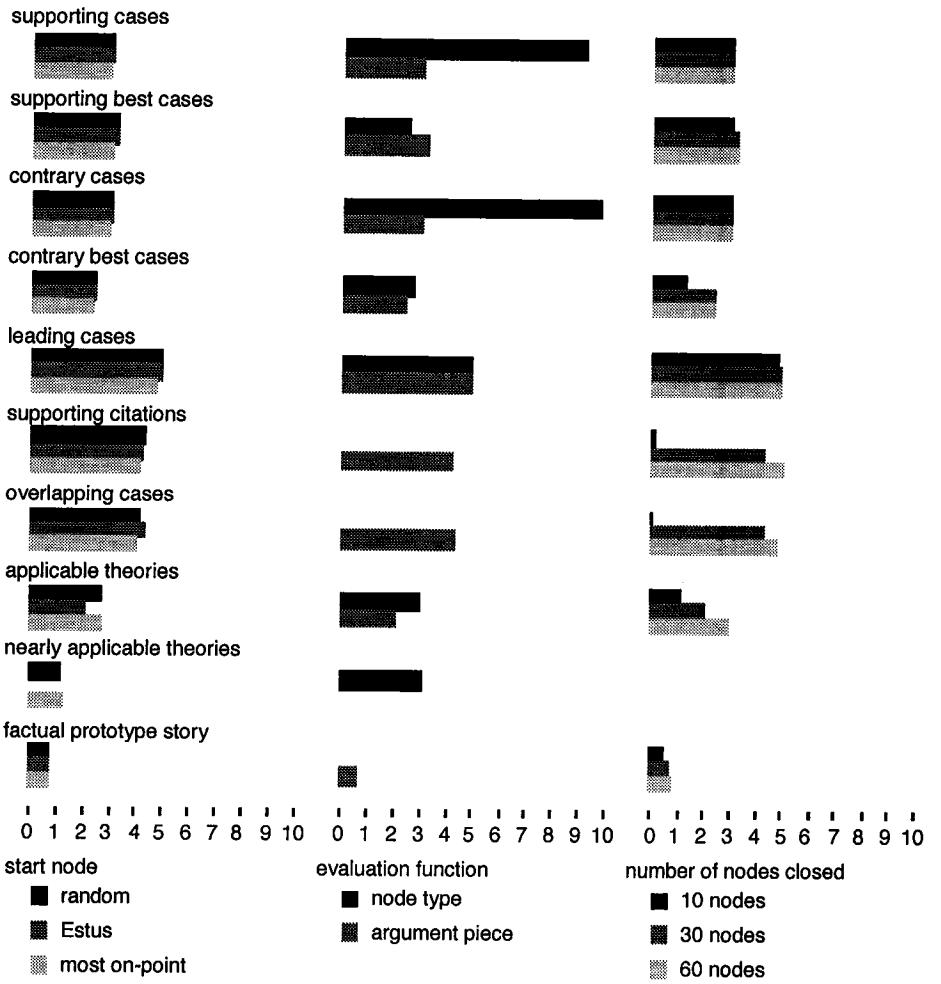

Figure 2: Average number of objects filling each argument piece as the start node is varied (left), the evaluation function is varied (middle), and the number of nodes closed is varied (right).

We ran BankXX *de novo* on all 54 cases in the case base to obtain averages for the number of objects filling each argument piece.[9]

2.2.1 Initial Query Formulation. Using the argument piece evaluation function and stopping search after closing 30 nodes, three different nodes were used as start nodes: a random case, the *Estus* case, and a most on-point case. The random case provides a base line. *Estus* is well known in this area of bankruptcy law—almost any research materials consulted by an attorney will soon lead to it—and therefore it may be considered a realistic and useful starting point. A most on-point case is another starting point likely to be relevant.

The results showed that the choice of start node, which is the initial query to the case base, made little difference to retrieval. As the left hand side of **Figure 2** shows, the average number of objects found for each argument piece is about the same for each of the three start nodes. We examined search paths through the case-domain graph to understand why. It turns out that no matter where search starts in this case-domain graph of 150 nodes, it soon leads to a highly interconnected region which contains many useful cases and theories. For example *Estus* and *Flygare* (another well known case) and the theories promulgated by these cases are part of this area of the graph. Informally speaking, it doesn't matter where search starts because in this domain all roads lead to *Estus*.

We conclude that in browsing a case base where there is a sense of location and a sufficiently rich indexing fabric,

[9]N.B., numbers of nodes closed, opened, and filling an argument piece are not the same. In general, many more nodes are opened than closed, and the total number of items filling the set of argument pieces exceeds the number of closed nodes (see Figure 2).

the initial probe to case-memory may not matter in a multiple-probe situation.

2.2.2 Case Retrieval Function. Next, we compared the effects of varying the evaluation function while keeping the 30 closed node limit and always starting at the *Estus* node. The node-type evaluation function finds more contrary cases and same side cases, but does so at the expense of failing to fill other argument pieces. See the middle of Figure 2. The node-type function uses only the type for each node and does not limit the number of objects retrieved for any argument piece. Considering its lack of knowledge, it does surprisingly well.

To understand how a knowledge-poor function can produce satisfactory results, one can consider search as just the first of a two-stage retrieval process for filling the argument pieces. The second stage applies the argument piece predicates to the retrieved objects to determine if they fulfill the requirements of the argument piece.

We conclude that in a two-phase retrieval, a knowledge-poor function to generate candidates in the first phase may be sufficient, as long as the performance criteria in the second phase are sufficiently rich. The efficacy of the classic generate-and-test or "many-are-called/few-are-chosen" (MAC/FAC) approach has been observed in other research as well [Gentner & Forbus, 1991].

2.2.3 Termination of Search of Case Memory. There is no objective standard for when one has completed research or completed an argument. Thus BankXX has two termination parameters that may be set by the user: limiting the time ("billable seconds") used and the number of nodes closed. In these experiments BankXX was set to terminate after it had closed 10, 30, and 60 nodes.

With the argument piece evaluation function and *Estus* as the start node, 10 nodes was too few to fill up many of the argument pieces. As a rough guide, 30 nodes seemed an appropriate compromise between more exhaustive search and too scanty an examination of the domain-graph. Incremental benefits of more search decreased after about 30 nodes. See the right hand side of Figure 2.

From the CBR perspective, we conclude that the decreased marginal utility of finding more cases causes there to be a point at which additional search of the case base is not effective. This conclusion echoes the results of Veloso and Carbonell [1991] as to the optimal amount of time to search a case base in a hybrid planner.

2.3 BankXX as an Argument Program

Using standard precision and recall measures, we compared the performance of BankXX with written judicial opinions. All 54 cases were run *de novo* with *Estus* as start node and a limit of 30 closed nodes. Results were averaged over the 54 cases.

2.3.1 Precision/Recall Performance and the Evaluation Functions. We were somewhat surprised to find that in general the knowledge-poor node-level evaluation function usually exhibited higher recall and precision than the knowledge-richer argument piece function. For instance, all the average recall values for the node-type function lie above the corresponding values for the argument piece function. Averaged over the four simplified argument pieces, the node-type evaluation function gave higher recall (0.55), than the argument piece evaluation function (0.44). We conclude that BankXX's overall **recall** performance seems to depend more on the choice of evaluation function than the choice of argument piece. The node-type evaluation function may give higher recall simply because it retrieves more items than the argument piece function. See the middle of Figure 2. The argument piece evaluation function is more selective but pays a price for that in recall.

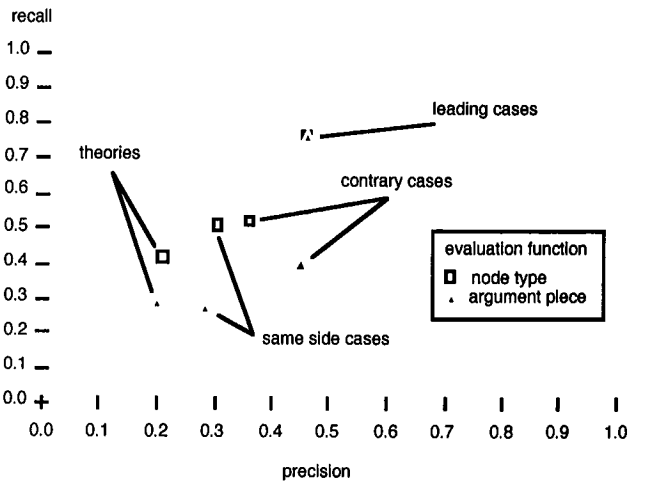

Figure 3: Average precision and recall (over all 54 cases) for the four aggregated argument pieces.

On the other hand, there seems not to be much difference in overall **precision** performance between the two evaluation functions. Each argument piece performs at about the same precision for each function. As we did in Section 2.2.2, we ascribe this to BankXX's two-stage approach: the lack of precision inherent in the node-type function is ameliorated by the precise filling of the argument pieces. Finally, we note that we did not observe the classical trade-offs between precision and recall. This might be because BankXX is not be functioning at a frontier where such phenomena occur or we need to vary other parameters to see them. In these studies, we only varied two, the evaluation function and the argument piece.

2.3.2 Recall/Precision and Argument Pieces. We observed differences in retrieval precision for the different argument pieces (see **Figure 3**). For both evaluation functions, highest precision was found for **cited-leading-cases** (0.46), followed by **cited-contrary-cases, cited-supporting-cases,** then **cited-legal-theories** (0.21). The results for recall were similar for the argument piece function. For the node-type function there was a flattening of performance differences among recall for the three argument pieces involving cases; all three did well.

We interpret the better precision on **cited-leading-cases** as follows. Since the same small group of leading cases are cited repeatedly in the opinions (that's what makes them leading cases), the probability that a given leading case is mentioned is higher than that for an arbitrary contrary or supporting case or legal theory. Thus if BankXX mentions a leading case it is likely to be in the opinion as well and hence BankXX's good precision marks on this argument piece.

For the other argument pieces, there is a wide range in the amount of information mentioned in the opinions. Thus if BankXX retrieves information not found in the opinions—which is likely to happen given BankXX's diligence in going after information—this lowers BankXX's precision. In particular, BankXX's low precision and recall scores on **cited-legal-theories** may be due to the high number of legal theories (18) relative to the number of cases (54), and the similarity of many theories. The program receives no credit for retrieving a useful but uncited theory in the absence of a metric to measure the similarity of the retrieved theory to the one actually applied by a court.

2.3.3 Precision-Recall Measures - Limitations. Again, let us note that the answers derived from actual opinions are not necessarily the best possible nor the only answers. Each opinion is the product of an individual judge and clerks. Some will cite many cases in support of their argument. Others will cite few. Some will mention only the legal theory of their particular judicial circuit. Others will look to other circuits as well. We found that earlier decisions, those written when the good faith issue was first being addressed under the new law, tended to look further afield and compared more different approaches. Once a number of appeals courts had set standards for analyzing good faith, opinions tended to look more exclusively to appeals cases in their own circuit for guidance.

Further, the way we have applied precision-recall measures—using the court's opinion as the "right" answer—is but one way to examine performance. Another would involve comparing BankXX with other programs. Without such comparisons, it is hard to judge BankXX's performance.

Lastly, these measures are problematic for a program like BankXX which seeks to harvest as much information as its resource limits allow. If BankXX retrieves information not found in the opinions—which is likely to happen given its biases—this lowers BankXX's precision and does not help its recall, even though BankXX might be doing a superb job of legal analysis. Benchmarks better measuring retrieval *accuracy*[10] are needed in our experiments—and CBR or AI and Law, in general.

[10]In engineering, accuracy is different from precision, which only notes to what decimal point one measures.

3. Conclusions

The general conclusion that we draw from BankXX is that the process of gathering information for an argument can be usefully modeled as heuristic search. In particular, the retrieval of cases and other knowledge can fruitfully be done with a combination of knowledge-based indexing and heuristic search. Using heuristic search as the mechanism to traverse memory permits relevancy assessment and case retrieval to be repeated iteratively in order to locate the nodes in the case graph that provide the underpinnings of an argument.

4. References

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

Branting, L. K. (1991). Integrating Rules and Precedents for Classification and Explanation: Automating Legal Analysis. Ph.D. Thesis, Technical Report AI90-146, AI Laboratory, University of Texas, Austin, Texas.

Gentner, D. & Forbus, K. D. (1991). MAC/FAC: A Model of Similarity-based Retrieval. *Proceedings of the 13th Annual Conference of the Cognitive Science Society*, 504-509. Chicago, IL. Lawrence Erlbaum, Hillsdale, NJ.

Kolodner, J. L. (1983). Maintaining Organization in a Dynamic Long-Term Memory. *Cognitive Science*, 7(4), 243-280.

Kolodner, J. L. (1993). *Case-Based Reasoning*. San Mateo, California: Morgan Kaufmann.

Martin, C. E. (1990). Direct Memory Access Parsing. Ph.D. Thesis, Yale University, New Haven, CT.

Rissland, E.L., Skalak, D.B. & Friedman, M. T. (1993). Case Retrieval through Multiple Indexing and Heuristic Search. *Proceedings, 13th International Joint Conference on AI*, 902-908. San Mateo, CA: Morgan Kaufmann.

Rissland, E. L., Skalak, D. B. & Friedman, M. T. (1994). Heuristic Harvesting of Information for Case-Based Argument. *Proceedings of the Twelfth National Conference on Artificial Intelligence (to appear)*, . Seattle, WA. AAAI Press/MIT Press.

Rosch, E. & Mervis, C. B. (1975). Family Resemblances: Studies in the Internal Structure of Categories. *Cognitive Psychology*, 7, 573-605.

Turner, R. (1988). Organizing and Using Schematic Knowledge for Medical Diagnosis. *Proceedings, Case-Based Reasoning Workshop 1988*, 435-446. Clearwater Beach, FL. Morgan Kaufmann.

Veloso, M. M. & Carbonell, J. G. (1991). Variable-Precision Case Retrieval in Analogical Problem Solving. *Proceedings, Third Case-Based Reasoning Workshop*, May 1991, 93-106. Washington, D.C. Morgan Kaufmann, San Mateo, CA.