A CBR Approach to Asymmetric Plan Detection

Daniel Fu
Stottler Henke
951 Mariners Island Blvd, Suite 360
San Mateo, CA 94404
650-931-2700
fu@stottlerhenke.com

Emilio Remolina
Stottler Henke
951 Mariners Island Blvd, Suite 360
San Mateo, CA 94404
650-931-2700
remolina@stottlerhenke.com

Jim Eilbert
CHI Systems, Inc.
716 N Bethlehem Pike, Suite 300
Lower Gwynedd, PA 19002
215-542-1400
jeilbert@chiinc.com

ABSTRACT
We describe an approach to the problem of detecting the execution of mission plans by the unconventional side in asymmetric warfare. This problem is characterized by actors who go to great lengths to avoid detection, while most of their actions are seemingly innocuous unless placed in a broader context. The problem is to find threatening patterns of action in a data collection characterized as massive, relational, incomplete, noisy, and corrupt. In this paper we describe Sibyl: a subsystem embodying a case-based reasoning approach to automated plan detection. Sibyl features a "spanning case base" that covers the space of theoretical scenarios. It uses each case in a state-space search algorithm by adapting case elements to the data. A simulator for Russian organized crime was used to generate case and test data. We describe Sibyl's algorithm and experimental results used in this approach.

Categories and Subject Descriptors
I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems.

General Terms
Algorithms, Performance, Experimentation

Keywords
Case-based reasoning, link discovery, plan recognition, relational databases

1. INTRODUCTION
In recent years, law enforcement and government agencies have displayed a growing interest in the prospect of detecting the activity of clandestine organizations. Terrorist organizations and organized crime are two such examples where members evade detection wherever possible so as to avoid mitigation of their efforts. Other than overtly illegal acts, most actions taken by members on behalf of the organization appear harmless. Actions such as phone calls, bank transactions, and fertilizer purchases are in and of themselves innocuous, yet when linked together the activities could constitute a threat. If law enforcement or government agencies are empowered with tools that recognize potential threats such as the construction of a bomb, they could potentially preempt a harmful plan before it comes to fruition.

The clandestine organization and the government agencies who oppose them are an example of asymmetric forces. Asymmetric warfare has seen increased attention in recent years, and comes in contrast to traditional notions of armed conflict involving force-on-force scenarios where each opposing side can be characterized according to doctrine, command and control structure, force size, weapon assets, etc. Opposing asymmetric forces have differing organization, ideology, support, and goals.

The government agencies that oppose clandestine organizations typically have three operational components: (1) recognition and collection of data, (2) data analysis and hypothesis formation, and (3) operational planning and execution. Data analysis and hypothesis formation is our focus here, particularly the discovery of a clandestine organization's plans. In this case, we worked in the domain of Russian organized crime. As part of the DARPA Evidence Extraction and Link Discovery (EELD) program, we used a simulator that takes as input a domain theory of how Russian mafias operate and proceeds to generate test data. Hierarchical task networks were used in part to describe the domain theory. Several simulation parameters are available to adjust quantity of data, noise, observability, corruption, and complexity.

2. PROBLEM DESCRIPTION
Other than the fact that clandestine organizations try to evade detection, there exist three significant obstacles any approach to detection will confront:

1. Massive data: The size of data is substantial. It breaks down into *primary* and *secondary* pieces of evidence. Primary evidence comes from news sources or intelligence agencies as relevant information. The size of primary evidence is eclipsed by secondary evidence, which is latent data such as phone numbers, street addresses, phone calls, and bank transactions.
2. Noise: Almost all secondary data is irrelevant, yet the parts that are relevant are absolutely necessary to recognize an asymmetric plan.
3. Incomplete information: Much of what we would consider to be relevant data is missing. As it is, successful mitigation of clandestine activity requires plan recognition before complete realization.

Despite these obstacles, one important regularity we identify is that the organization's behavior is ultimately goal-driven. The behavior is structured, occurs over a long duration (months to years), and involves several people. Thus, the behavior is the logical execution of a plan which motivates our approach.

3. SCOPE
The CBR system we developed, called Sibyl, resides within a bigger system called SCOPE which stands for Socio-Culturally Oriented Planning Environment [2]. The system seeks to improve upon the human analysis process by automatically linking

evidence from a number of sources into graphs, and formulating hypotheses correlating these graphs to underlying plans.

The relevant knowledge/data bases available to a SCOPE model (or an analyst) include:

- A set of known facts about the current activity, mainly about breaks in the terrorist organization's secrecy, and the relations among those facts;
- A catalog of organizations and general information about each of them;
- A set of mission plan templates (MPTs), crafted by intelligence analysts, that capture the invariance in the planning process associated with a particular domain;
- A database of cases; and
- A historical and theoretical knowledge about how organizations train, acquire financing, communicate, plan, and operate, as well as information concerning religious, ethnic, and cultural factors that may impact their operations.

SCOPE provides mechanisms for reasoning about and combining these different sources of information. The architecture used in SCOPE is a synthesis of cognitive modeling and CBR technologies. The fundamental objects passed between the SCOPE modules are hypotheses about the organization's plan. One SCOPE module is based on a cognitive model of an intelligence analyst conducting situational logic [4], which is built using the iGEN toolset, a blackboard-based approach to cognition [5]. This module acts as SCOPE's primary controller. It also encodes the information in MPTs within a set of cognitive tasks, and has the meta-cognitive ability to spawn and track "what if" hypotheses about plausible mission plans. The cognitive model module reasons about how plausible hypotheses about plan components fit together, given the organizational and cultural constraints. It will also manage the active hypotheses related to MPTs taking into account the uncertainty in the evidence and sensitivity of the hypotheses.

The Sibyl module matches current evidence to plans in its case base, generating plausible hypotheses about the current plan. By combining and exchanging of hypotheses between the iGEN and Sibyl modules, the SCOPE system takes advantage of their complementary strengths and weaknesses while generating hypotheses on mission plan execution. Sibyl needs a bigger portion of the complete evidence graph, but is not sensitive to misconceptions an analyst may have as embodied in an MPT. iGEN can function with an evidence graph that instantiates a much smaller portion of a mission plan than Sibyl; however it is quite sensitive to pattern description errors that may get into an MPT.

Now that we have discussed Sibyl in its broader context, we now focus discussion on the Sibyl domain and algorithm.

Workshop on Link Analysis for Detecting Complex Behavior, KDD-2003, August 27, 2003, Washington, D.C.

4. APPLICATION DESCRIPTION

Because the traditional CBR approach of computing a feature vector from the input data does not suggest an obvious representation for our problem, our strategy was to form a *spanning case base* covering the full range of possible plans. Using a domain theory and simulator, we generate nearly all of the mission plans for activity possible in the domain. Detecting plans in the evidence data amounts to a search through the case memory for the case that is consistent with a subset of data. We match the entire case base against evidence. Hence, the CBR phase of adaptation is paramount while retrieval is secondary; in fact, there is no indexing. This approach is the core technology basis for Sibyl.

The immediate consequence of employing a spanning case base is a massive case base. To make our approach practical, we used the Cyc ontology to reduce the case size by abstracting event types. For example, sending an email could be equivalent to a phone conversation. By abstracting case elements, it was possible to condense millions of cases into hundreds.

Having reduced the case base size, we focused on the creation of fast mapping techniques. We match a stored case against the evidence taking into account that (i) actors in the case are not the same as in the evidence (e.g., people in the case are different from people in the evidence), (ii) events in a case can be fulfilled by different events in the evidence (e.g., a meeting and a phone call can have the same purpose), and (iii) not all the relationships in the case have to be known in the evidence (i.e., evidence is incomplete).

For the rest of this section we describe the simulator and how it was used to generate the case base. This is followed by a description of our representation and algorithm.

4.1 Simulator

The domain theory for the simulator consists of task descriptions that specify how to populate a world with people, relationships between people, mafias, companies, geographical regions, etc. More importantly, it specifies how the world works; e.g., how a murder event comes to fruition, starting with (say) a mafia gang war. One low-level task might be an exchange of money. This can happen through a bank wire or a cash exchange. The goal of exchanging money between a middleman and hitman could be accomplished by either method, chosen at random according to specified simulation parameters. Likewise, many other tasks can be accomplished through multiple methods.

Note the simulator is a black box. Because we had neither access to simulator source code, nor the inclination to re-create it, we used the simulator to help us generate a case base because it could also report the ground truth; i.e., what really happened in the world as opposed to reported events. We ran the simulator several thousand times with differing random seeds until we converged on a nearly complete case base, discarding duplicate cases.

According to the domain theory, there could be over three million potential cases. Through abstraction, we narrowed cases down to 1,500.

4.2 Case and Evidence Representation

A case describes how a particular event (e.g., contract-kill, phone call, wire transfer) took place. Events are described in terms of subevents and properties associated with the event. Every object in the representation (e.g., events, event property values) has a type, and types are organized by the Cyc ontology. Events are linked by subevent relationships and by common actors (e.g., the same person making a bank deposit and a phone call). Events have associated spatio-temporal properties: where and when they occurred. The value of these properties admits various degrees of uncertainty (e.g., a murder event happened somewhere in Europe on May 2000). We can think of a case as a directed graph where nodes represent objects in the case (e.g., events, people, telephone numbers, bank accounts) and edges represent relationships between objects.

Figure 1 shows a partial example of a case involving a murderForHire case (UID6166). A hit contractor (UID5312) made a phone call (UID6136) to a middleman (UID5317) to arrange a murder. Then the contractor paid (UID6141) by doing a wire transfer (UID6139) from his account (UID5306) to the middleman's bank account (UID5294). The middleman eventually hires the perpetrator (UID5321), who observed (UID6151) the victim (UID5160) before performing the murder (UID6153).

The evidence is a database of reported events. In general this database is incomplete. For example, a murder event can be reported where the killer is not known. A PlanningToDo-Something event can be reported without reporting its subevents: whether the persons planning to do something met or talked on the phone.

4.3 Search Algorithm

For each case, we use a best-first search algorithm to match the case against the evidence. A search state is a tuple $<c,r,m>$ where c is the case, r is a list of edges in c that need to be considered for a match, and m is a set of pairs, each pair consisting of a graph node in c and a node in the evidence. Let f_m be a one-to-one function for node pairs in m such that $f_m(a) = a'$ where a is a node in c and a' is a node in evidence. The set of pairs has an associated weight indicating node similarities. A heuristic evaluation function F assesses the quality of the mapping m by assigning m a real number, where higher values of m are better.

The initial states of the search have the form $<c,e_c,\emptyset>$ where e_c is a list of all the edges appearing in case c. The initial order in e_c is important for the performance of the algorithm as Sibyl processes edges sequentially (we later present our heuristic to order e_c). The basic best-first search algorithm we use is as follows:

1. Let initial states $H = \{<c_0,e_{c_0},\emptyset>, <c_1,e_{c_1},\emptyset>, <c_2,e_{c_2},\emptyset>, \ldots, <c_n,e_{c_n},\emptyset>\}$ where n is the number of cases in the case base.
2. Identify best hypothesis $h = <c,r,m>$ from H.
3. If r is empty, h is the best hypothesis. Stop.
4. Generate successors S from h.
5. Let $H = H - \{h\} \cup S$.
6. Go to 2.

We derive the successors of a state $<c,r,m>$ by considering the first edge e in r. Either m already pairs the two nodes in e, or pairings for those nodes need to be generated and added to m. Each directed edge is a tuple $<s,d,l>$ where s is the source, d is the destination, and l is the label. Here is the procedure for

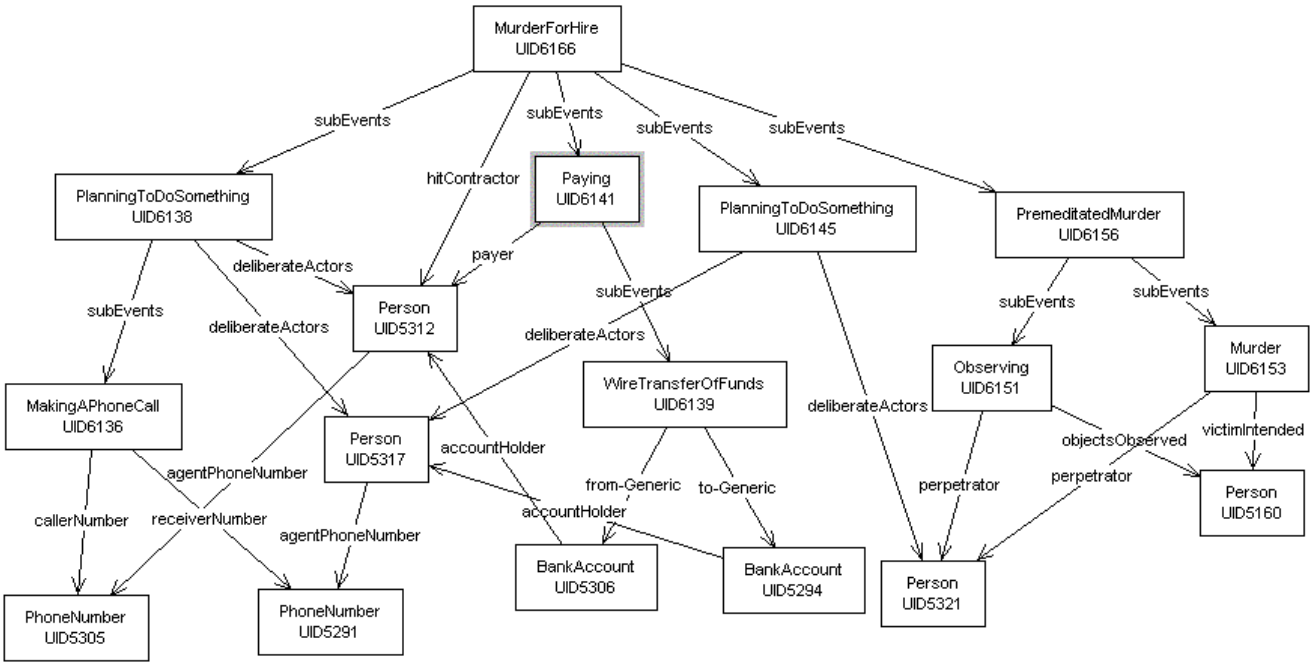

Figure 1: A partial case from the case base.

generating successors.

```
Procedure GENERATE-SUCCESSORS (state <c,r,m>)
  Let the first of r be edge <a,b,l>
  if (a and b are in the domain of f_m)
  {
     if edge <f_m(a),f_m(b),l> is inconsistent with evidence
          return Ø
     else
          return { <c, r – {<a,b,l>}, m> }
  }
  else
  {
     Let r' = r – { <a,b,l> }
     Let s = { <c,r',m> }
     For each edge <a',b',l> in evidence matching <a,b,l>
     {
          Let m' be a new copy of m with
               f_m'(a) = a' and f_m'(b) = b'
          s = s ∪ { <c,r',m'> }
     }
     return s
  }
```

Since the evidence is generally incomplete (e.g., usually the perpetrator in a murder is not known), not all the edges in a case require a counterpart in the evidence. The state $<c,r',m>$ is a possible successor for $<c,r,m>$. In this case, the edge $<a,b,l>$ is not required to hold in the evidence. The heuristic evaluation function will penalize this state but the state will remain in the search queue.

Next we describe the main aspects of the above algorithm: detecting inconsistent mappings, evaluating the goodness of a state, generating match candidates, and pruning the search space.

4.4 Inconsistent Mappings

Since the evidence is incomplete, we cannot generally check whether an arbitrary relationship is false (i.e., it does not hold in the evidence) or is missing from the evidence. There are, however, instances where an edge $<f_m(a),f_m(b),l>$ could not exist in the evidence, allowing mapping m to be deemed inconsistent. For example, the victim of a murder is always unique and usually identified, and so it is possible to consider a mapping inconsistent if it posits a second victim for the same murder. In contrast, the attendees of a meeting are not unique and usually unidentified (i.e., a report can indicate that a meeting took place but the report may not indicate all the participants). In these instances, it is not possible to decide whether a mapping is inconsistent.

4.5 State Evaluation Function

Our state value heuristic is a function of the number and similarity of matches. The function is made up of the following parameters associated with a state $<c,r,m>$:

- u is the number of case edges not matched to evidence; i.e., there is no analog in the evidence
- v is the number of case edges matched to evidence
- n (= $u+v+|r|$) is the number of edges in case graph c
- w is the average pair similarity in m
- α and β respectively weigh how much importance is given to the quality of the matches associated with the evidence explored so far and how much importance is given to the search progress so far

The form of the state evaluation function is:

$$F(\langle c,r,m\rangle) = \alpha(v-u) + \beta w \frac{u+v}{n}$$

4.6 Generating Match Candidates

Generating match candidates always occurs in the context of explaining the case edge associated with a state $<c,r,m>$. The candidates for a case node are evidence nodes of the same exact type that preserve a given set of labels to m: A match m preserves a label l if for each case edge $<a,b,l>$ if an edge $<f_m(a),f_m(b),l>$ holds in the evidence, whenever $f_m(a)$ and $f_m(b)$ are defined in the evidence. Nodes in a pairwise mapping must be of the same type, thus there will also exist edges $<a,t_1,isa>$, $<f_m(a),t_1,isa>$, $<b,t_2,isa>$, and $<f_m(b),t_2,isa>$ where t_1 and t_2 are types such as Person or EmailSending, and the isa label denotes the type relationship. The set of labels that must be preserved include subEvent, accountHolder, agentPhoneNumber, and to-generic. Not all the edges in a case must be preserved, since evidence is in general incomplete.

Consider the problem of matching case node UID_1 representing an email sending event as edge $<UID_1, EmailSending, isa>$. Suppose in addition that it is known that UID_1 is a subevent of a planningToDoSomething event UID_2 (edges $<UID_1, UID_2, subEvent>$ and $<UID_2, planningToDoSomething, isa>$). Moreover, it is the case that UID_2 has already been mapped: $f_m(UID_2)$ exists. Since we want to preserve the subEvent relationship, the match candidates for UID_1 will be all evidence nodes x such that edges $<x, f_m(UID_2), subEvent>$ and $<x, EmailSending, isa>$ exist in the evidence.

4.7 Pruning Heuristics

The size of the match candidate set determines the branching factor of the search. The smaller the set the better. As early as possible, it is important to prune search paths leading to inconsistent hypotheses. In addition to preserving a certain link, other pruning heuristics include:

- Mappings are one-to-one relationships: remove from candidates those evidence nodes already in the range of f_m.
- Matching should preserve temporal constraints: if event #1 occurs before event #2 in the case, then f_m(event #1) should occur before f_m(event #2) in the evidence.

4.8 Temporal Reasoning

Each event e has an interval $[lb(e), ub(e)]$ where lb is lower bound, ub is upper bound, delimiting when the event must have occurred. A subevent of e occurs in the time window of his parent: if e_1 is subevent of e_2, then $[lb(e_1), ub(e_1)] \subseteq [lb(e_2), ub(e_2)]$. A mapping m is consistent if it preserves all temporal relationships between events known in a case. If case events e_1 and e_2 have a relationship R where $[lb(e_1), ub(e_1)]$ R $[lb(e_2), ub(e_2)]$, and $f_m(e_1)$ and $f_m(e_2)$ are defined, then the evidence must have the same relationship $[lb(f_m(e_1)), ub(f_m(e_1))]$ R $[lb(f_m(e_2)), ub(f_m(e_2))]$.

Sibyl uses the above condition to prune the set of match candidates for events in a case. For example, suppose the following:

- *<meeting32, planning35, subEvent>*
- *<meeting33, planning35, subEvent>*
- *ub(meeting32) < lb(meeting33)*
- *ub(meeting33) < ub(planning35)*
- f_m*(planning35) and* f_m*(meeting32)* are defined

A candidate evidence node *x* for meeting33 must satisfy

ub(f_m*(meeting32)) < lb(x)* and *ub(x) < ub(*f_m*(planning35))*

In practice, as the matching process maps additional related events, the temporal constraints become much more stringent.

4.9 Ordering the Case Evidence

Recall that the initial states of the search have the form $<c,e_c,\varnothing>$ where e_c is a list of all the edges comprising case c. During the search, the list is explored sequentially. The initial order of edges is important for the performance of the algorithm. For example, it would be unwise to start the search matching a phone call event, which will have a massive number of possible matches in the evidence, rather than to start the search matching a murder event, which has fewer possible matches and provides more information about the key actors in the case (e.g., the victim or the person following the victim before the murder).

In our current application, the user manually specifies a partial order in which events in a case should be considered, with key events types having highest priority. These events offer tend to constrain the number of viable match candidates. The algorithm "grows" a single connected graph by continually selecting immediate edges based on the user's specification. The order in which edges are added to the graph is the order of e_c.

4.10 Abstracting the Case Representation

So far for the case-matching algorithm, a case node must be matched to an evidence node of the same type. This turns out to be too restrictive as the number of distinct cases would be in the millions. Two cases could be identical except for one single event, perhaps a phone call in one, and an email in the other. To shrink the size of the case base, we abstracted events using the "isa" relationship in Cyc. Edge labels were renamed to be abstracted types. Thus, a phone call and email would be renamed to be a generic "contact" event.

4.11 Matching a Case Base to the Evidence

Earlier in Section 4.3, we discussed the search procedure and method for generating new states. When applied with a case base, we want to let all cases have an opportunity to match against the evidence. We therefore employ a round-robin timeout approach such that during a round Sibyl uses the best state originating from each case in the case base. For each state chosen, either a match is found, or, more frequently, a time limit halts the search for the time being until the next round.

After a round ends, the time limit for the next round is increased and the parameters of the heuristic functions are changed: α (representing the quality of the match) is decreased and β (representing the depth of the search) is increased. The term β/α is proportional to the number of case edges that are allowed to be skipped before backtracking. When no matches have been found, our round-robin policy attributes the situation to a lack of evidence supporting edges in a case. Consequently, the policy increments β/α to increase chances of finding a match.

After a match is found, Sibyl subtracts the matching evidence from the evidence body. The process then repeats until no more cases match and the ratio exceeds an empirically adjusted threshold.

5. Evaluation

We evaluated SCOPE system as part of DARPA's EELD year 2002 evaluation. The evaluation software was available to all participants. A total of fourteen evidence databases were used (see Table 1), each with a Bayesian and task network generated version. Because we imported the task-based simulator data directly into the case base, and because the iGEN portion used the task network for knowledge engineering purposes, we only tested SCOPE on the task network datasets. In Table 1, *Size* refers to the number of valid threats in the evidence. A threat is a "valid" behavior pattern that is present in the evidence. *Observability* refers to how complete the evidence is. *Connectivity* measures the degree up to which the same people/events are part of different threats. *Corruption* refers to how accurate the evidence is; e.g., whether middleman and killer roles are swapped for two people. *Noise* refers to evidence that might be useful, but are not.

Table 1: Test data set characterization.

Data Set	Size	Observability	Corruption	Connectivity	Noise
1	Small	Very high	None	Medium	None
2	Small	Average	None	Medium	Medium
3	Small	Very low	None	Medium	High
4	Medium	Very low	None	Medium	High
5	Medium	Average	None	Medium	Medium
6	Medium	Very high	None	Medium	None
7	Medium	Average	None	High	Medium
8	Medium	Average	None	Low	Medium
9	Medium	Very high	Low	Medium	Medium
10	Medium	Very high	High	Medium	Medium
11	Large	Very low	None	Medium	High
12	Large	Average	None	Medium	Medium
13	Large	Very high	None	Medium	None
14	0	Average	None	Medium	High

For each dataset, the output of our application was fed into a scoring program where the output was assigned a number representing the overall match quality. The scoring program employs a metric incorporating the notion of *social cost*. This cost can be thought of as the amount of effort involved in either investigating a false positive, or ignoring a true positive. An output correctly capturing all threat events with no false positives will have zero social cost while an output either having false positives or not having true positives will have greater than zero cost.

The program scores an output by comparing it to the "answer key" which contains all the threat events. It uses a greedy heuristic to pair threat events for comparison (although users may elect to

exhaustively pair threats). From there, the program uses an edit distance algorithm to compare two graphs. An "edit" is the addition/deletion of a node/edge. A series of edits will transform one graph into a duplicate of the other. The sum of all the edits is the edit distance. The algorithm was specialized to use ontological distance between two nodes/edges as part of the cost of an edit. Examples could be changing simple dollar amounts, dates, or terms in the Cyc ontology. More complex examples could be editing high-level events with differing sub-events.

Relating edit distance to social cost, a completely correct hypothesis has zero edits and zero social cost, while any necessary edits will incur a positive social cost. This metric was normalized such that the output of the scoring program was a real number between 0.0 and 1.0, the smaller the better.

Only the iGEN scores were submitted for formal evaluation. For our own evaluation purposes, Sibyl and iGEN were separately tested. Figure 2 shows the scoring results for our application, broken into Sibyl and iGEN scores. On the normalized social cost metric, Sibyl scored zero on three of the fourteen datasets provided, indicating a perfect mapping, and scored greater than zero and less than 0.05 on four others, indicating a near-perfect mapping. iGEN scored zero on three, and greater than zero and less than 0.05 on five others.

In general, the lower the dataset's observability, or the higher the corruption, the harder it is to detect a plan in the evidence; cf, datasets 3, 4, 10 and 11. Sibyl's pruning heuristics and the order of a case's edges favor high-level events (e.g., planning a murder) over low-level events (e.g., making a phone call). The heuristics will prune valid search paths in the presence of high-level noise. Figure 2 suggests how SCOPE can benefit from threat hypotheses generated using Sibyl (e.g., datasets 3 and 8). When integrated with iGEN, Sibyl will contribute as a hypothesis generation module, and iGEN will manage hypotheses from modules like Sibyl.

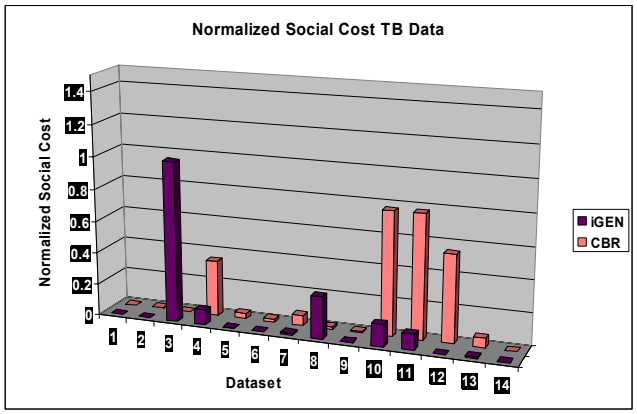

Figure 2: Evaluation results for task-based datasets

In summary, although the evaluation does not allow one to draw any general conclusions regarding relative overall rankings of the performers or technologies, we nevertheless feel our results are promising.

6. Related Work

There has been increasing work in CBR that uses graph-based representations. Perhaps the most closely related is the Caper system [7] which searches for subgraphs within a semantic network. Similar to Sibyl, there is no *a priori* indexing, or construction of a feature vector.

Bergman & Stahl [1] use object-oriented "class hierarchies" to model the similarity and differences between objects. Because this method relies on objects being thought of as distinct entities, it is unclear how it applies to our problem as, for example, a person in a case has several relationships to other persons and events. Indeed, a person's relevance is a product of the person's actions and relationships to other people. It is simply not possible to judge similarity through a myopic lens. The myriad connections among associated people and events must be considered.

Messmer & Bunke [6] detail an algorithm that constructs a decision tree to determine subgraph isomorphism in polynomial time. The approach will not scale with our problem as the tree is exponential in the size of the input in worst case. As well, the matching is a form of exact matching.

Gentner & Forbus [3] describe the MAC/FAC system which is a model of analogical reminding. Matches are made between structurally similar concepts and verified in the SME portion of FAC. Wolverton & Hayes-Roth [8] also explore analogical retrieval, but focus on successive revision of heuristics to guide search.

Though Sibyl shares a graph representation similar to all of the above work, Sibyl's differences with all these approaches is driven by the nature of input data. Because of the novel nature of the data, the search mechanism must work by adapting its cases to the data. This is in contrast to the related systems that handle small amounts of input data to search over a larger case base or semantic network.

7. Conclusion

We have described a CBR approach to plan detection that handles input data characterized as relational, massive, noisy, incomplete, and corrupted. The nature of the data demanded a new perspective on case retrieval and adaptation. Case retrieval, typically emphasized in the literature, was non-existent in Sibyl as cases were never indexed. Indexing would have required some processing on the input to construct a feature vector. This is an untenable task for two reasons. First, any fragment of evidence could be somehow relevant, but to determine its relevance, more evidence must be considered. What is important here is the relationship between fragments. Only together can they form a threatening pattern. Second, considering all input evidence is out of the question.

Case adaptation was our focus. We started with a strong domain theory of mission plan execution, and concentrated on mapping complete cases to the evidence. Because no case is preferred over another initially, the case base needed to be condensed from millions into hundreds. This was achieved by abstraction of isa types using the Cyc ontology. Cases were mapped using a search heuristic that traded off mapping quality with search progress. Pruning heuristics, such as temporal ordering, were used to limit the search space.

The combination of AI search techniques and domain dependent pruning heuristics made our case adaptation algorithm effective for DARPA's EELD year 1 evaluation.

8. Acknowledgements

This research is sponsored by the Defense Advanced Research Projects Agency and managed by Rome Laboratory under contract F30602-01-C-0200. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of the Defense Advanced Research Projects Agency, Rome Laboratory, or the United States Government.

We are thankful to the engineering staff of the AFRL Information Directorate for their comments on an earlier draft of this paper, as well as anonymous reviewers.

9. References

[1] Bergmann, R., and Stahl, A. "Similarity measures for object-oriented case representations," in Proceedings of the 4th European Workshop on Case-Based Reasoning, 1998.

[2] Eilbert, J.L., Carmody, D.M., Fu, D., Santarelli, T., Wischusen, D., and Donmoyer, J. "Reasoning about adversarial intent in asymmetric situations," in AAAI Fall Symposium on Intent Inference, 2002.

[3] Gentner, D., and Forbus, K.D. "MAC/FAC: A model of similarity-based retrieval," in Proceedings of the Cognitive Science Society, 1991.

[4] Heuer, R. The Psychology of Intelligence Analysis. Center for the Study of Intelligence, Central Intelligence Agency, 199. http://www.cia.gov/csi/books/19104/index.html

[5] Le Mentec, J.-C., Zachary, W., Iordanov, V. "Knowledge Tracing of Cognitive Tasks for Model-based Diagnosis," in Proceeding of 9th International Conference on Artificial Intelligence in Education, 1999. Le Mans, France.

[6] Messmer, B.T., and Bunke, H. "Subgraph isomorphism in polynomial time," 1995. Technical Report TR-IAM-95-003.

[7] Sanders, K.E., Kettler, B.P., and Hendler, J.A. "The case for graph-structured representations," in Proceedings of the Second International Conference on Case-based Reasoning, 1997.

[8] Wolverton, M., and Hayes-Roth, B. "Retrieving Semantically Distant Analogies with Knowledge-Directed Spreading Activation," in Proceedings of the National Conference on Artificial Intelligence, 1994.