A STAB at Making Sense of VAST Data

Summer Adams & Ashok K. Goel

Artificial Intelligence Laboratory & Southeastern Regional Visual Analytics Center
School of Interactive Computing, Georgia Institute of Technology
Atlanta, GA 30332, USA
{summer@cc.gatech.edu, goel@cc.gatech.edu}

Abstract

We view sensemaking in threat analysis as abducing stories that explain the current data and make verifiable predictions about future data. We have developed a preliminary system, called STAB, that abduces multiple stories from the VAST-2006 dataset. STAB uses the TMKL knowledge representation language to represent skeletal story plots as plans with goals and states, and to organize the plans in goal-plan-subgoal abstraction hierarchies. STAB abduces competing story hypotheses by retrieving and instantiating plans matching the current evidence. Given the VAST data incrementally, STAB generates multiple story hypotheses, calculates their belief values, and generates predictions about future data.

Background, Motivation and Goals

Making sense of vast amounts of data in intelligence analysis (Heuer 1999; Krizan 1999; Thomas & Cook 2005) generally involves the tasks of recognizing and characterizing a threat based on some initial evidence about an event or activity, generating multiple explanatory hypotheses based on the evidence, collecting and assimilating additional data, evaluating the multiple explanatory hypotheses, and selecting the most plausible hypothesis. The sensemaking task is complex because of the constantly evolving, and often unreliable and conflicting, nature of data. The evolving nature of data implies a need for ongoing monitoring and continual generation and evaluation of hypotheses so that new evidence can be accounted for as it arrives and the most confident explanation can be produced at any given time.

Pirolli & Card (2005) describe an information-processing model of threat analysis based on a cognitive task analysis of human analysts as they did their jobs. They have identified two main, overlapping loops in the analyst's problem solving, a foraging loop and a sensemaking loop. The foraging loop involves finding the right data sources; searching and filtering the information; and extracting the information. The sensemaking loop involves iterative development of a conceptualization (a *hypothesis*) from a stored *schema* that best fits the evidence, and the presentation of the knowledge product that results from this conceptualization. As Bodnar (2005) notes, the information processing in this model is both bottom-up (from data to hypotheses) and top-down (verifiable predictions made by the instantiated hypotheses). This model, however, does not identify the content and the structure of the schemas in the sensemaking loop, or describe the process by which specific schemas are conceptualized as hypotheses. The goal of our work is to help answer these questions in a manner consistent with the current cognitive accounts of intelligence analysis (Heuer 1999; Krizan 1999; Pirolli & Card 2005).

We view the task of sensemaking in threat analysis as that of abducing stories that explains the current data and makes verifiable predictions about future data. We adopt the general conceptual framework for abductive reasoning described in (Bylander et. al., 1991; Goel et. al. 1995; Josephson & Josephson 1995) and use it for story abduction. We have developed a preliminary system, called STAB (for STory ABduction), that abduces multiple, competing stories from the VAST 2006 dataset[1]. Since real intelligence data is not available in the public domain, the National Visual Analytics Center (NVAC) at the Pacific Northwest National Laboratory (PNNL) generated the VAST dataset as a substitute. This synthetic dataset pertains to illegal and unethical activities, as well as normal and typical activities, in a fictitious town in the United States.

STAB abduces multiple competing story hypotheses by retrieving and instantiating skeletal story plots that match the current evidence. STAB uses the TMKL knowledge representation language (Murdock & Goel 2003, 2007) to represent a skeletal story plot as a plan with goals and states, and to organize the plan in a goal-plan-subgoal abstraction hierarchy. This knowledge representation captures both intent and causality at multiple levels of abstraction. Given the VAST data incrementally, STAB generates multiple story hypotheses, calculates their belief values, and generates predictions about future data.

---

[1] See http://conferences.computer.org/vast/vast2006/.

We view STAB as an automated assistant that may act as an external memory for human analysts. The external memory may support threat analysis by (a) focusing the attention of the analyst on specific hypotheses, and (b) keeping track of multiple, competing hypotheses.

A Model of the VAST Dataset

We begin with an introduction to the VAST-2006 dataset. The VAST dataset contains over a thousand news stories written in English, and a score of tables, maps and photographs. Figure 1 shows an example news story from the VAST dataset. We analyzed this dataset and screened the stories that indicated an illegal or unethical activity, which left about a hundred news stories out of the more than a thousand originally in the dataset. We then analyzed the remaining stories and manually extracted all the events that pertained to illegal/unethical activities; Table 1 shows a sample of such events that form the input to STAB. We also hand crafted representations for each of these events in terms of the knowledge states it produces; Table 2 shows the representations for a sample of events input to STAB. In addition, we examined the maps, photos and tables that are part of the VAST dataset and similarly extracted and represented the relevant input information.

Torch scandal?

Story by: John Panni
Date Published to Web: 4/30/2004

Political wags in Alderwood are excitedly discussing the impact of steamy photos taken of Mayoral democratic candidate John Torch with an unidentified young brunette woman late one evening at a Tri-Cities Starbucks. Torch, married with 4 children, has not commented on the incriminating pictures. Hawk Press has obtained copies of these pictures, but following company policy, will not publish them.

Incumbent mayor Rex Luther characterized the scandal as "unfortunate". "Moral values are key to anyone wishing to assume a position of leadership and responsibility," he added.

Sources have identified the woman as an employee of Boynton Laboratories. Laurel Sulfate, spokeswoman for the laboratory, was unavailable for comment, currently vacationing in Switzerland. An assistant to Sulfate stated that she "will look into the matter upon her return."

Webmaster
 Copyright
Hawk Press Inc.

Figure 1. Example news story from the VAST dataset.

TABLE 1
SAMPLE INPUTS FOR STAB

Sample STAB Inputs
stolen(money $40 Highway-Tire-Store)
cured-disease(Boynton-Labs   Philip-Boynton   prion-disease)
named-after(lab Philip-Boynton Dean-USC)
was-founded(Boynton-Labs)
have-developed(Boynton-Labs prion-disease)
announced-investigation(USFDA Boynton-Labs)
discontinued-investigation(USFDA Boynton-Labs)

Injected-mouse(Boynton-Labs prion-disease)
Injected-cow(Boynton-Labs prion-disease)
treatment-mouse(Boynton-Labs prion-disease)
treatment-cow(Boynton-Labs prion-disease)

TABLE 2
KNOWLEDGE STATE PRODUCED BY
THE SAMPLE INPUT EVENTS

Actions	Resulting State
Stolen	Has-object
Broken	Is-broken
Cured-disease	Is-rich-and-famous
Named-after	Expert-involved
Was-founded	Is-open
Have-developed	Exists-new-disease
Announced-investigation	Is-investigating
Discontinued-investigation	Cancel-investigation
Injected-cow	Cow-is-infected
Treatment-cow	Cow-is-cured

Abductive Reasoning, Plan Recognition, and Story Understanding

Abduction is generally characterized as inference to the best explanation for a given set of data. The general task of abduction takes as input a set of data, and has the goal of giving as output the best explanation for the dataset. Abductive reasoning includes generation, criticism and possible acceptance of candidate explanatory hypotheses. Factors that make one explanation better than another include explanatory coverage, plausibility, and, in case of composite explanations, internal consistency and parsimony. AI research has led to a number of computational models of abduction starting with Pople's (1977) early work on medical diagnosis. These models range from normative models based on minimal set covering algorithms, e.g., (Reggia, Nau & Wang, 1983), logical reasoning from first principles, e.g., (Reiter 1987), and probabilistic inference in Bayesian networks, e.g., (Pearl 1987); to functional models based on task decomposition, e.g., (Josephson & Josephson 1994).

The input to the abduction task in threat analysis is characterized by the following features (e.g., the VAST dataset):
- The amount of data is huge.
- Data comes from multiple sources and in multiple forms.
- Data from various sources may be unreliable and conflicting.
- Data arrives incrementally and is constantly evolving.

- Data may pertain to multiple actors, where the actions of the various actors need not be coordinated.
- The actors may try to hide data about their actions, and may even introduce spurious data to hide their actions.
- Data may pertain to novel actors as well rare actions.
- The amount of useful evidence is a small fraction of the vast amount of data (the proverbial "needle in the haystack" problem).

The desired output of the abduction task in threat analysis include the following:
- Explanations that causally relate specific sequences of actions into plans and stories. Since the data pertains to actions of multiple, uncoordinated actors, the output may contain multiple unconnected stories.
- Explanations that specify intent of the various actors in the stories. Ideally, the intent should be specified for specific subsequences of actions in addition to complete sequences.
- Belief values for the explanations that can help focus and prioritize subsequent processing of data.
- Explanations that can make verifiable predictions.

Given the structure and complexity of the abduction task in threat analysis, we have adopted the general functional model of abductive reasoning described in (Bylander et. al., 1991; Goel et. al. 1995; Josephson & Josephson 1994). This functional mechanism for abduction has previously been used in a number of domains, such as medical data interpretation and diagnosis, scientific theory formation, speech recognition, and diagram interpretation (Josephson & Josephson 1994). This model characterizes the best composite explanation for a set of data based on three criteria (Bylander et. al, 1991 and Goel et. al (1995) provide more formal specification):

1. *Coverage:* One composite explanation is better than another if explains more of the observed data.
2. *Belief:* One composite explanation is better than another if it has a higher belief value.
3. *Parsimony:* One composite explanation is better than another if it is a subset of the other.

The model then characterizes the abduction task as finding a composite explanation for the given set of data such that it satisfies four conditions:

I. The composite explanation explains as much of the observed data as possible in the available time.
II. The composite explanation is internally consistent.
III. The composite explanation is parsimonious, i.e., no proper subset of the composite explanation can explain as much of the data.
IV. The composite explanation is a confident explanation; an explanation is confident if its belief value is significantly higher than that of alternative explanations.

Plan Recognition: Schmidt, Sridharan & Goodson (1978) identified plan recognition as an AI problem, and proposed a *hypothesize-revise* mechanism for addressing it. Cohen, Perrault & Allen (1982) distinguished between two kinds of plan recognition situations: *keyhole,* in which the observed actor acts as if unobserved, and *intended*, in which the observed actor performs actions to aid in the plan recognition. In threat analysis, we can distinguish additional types of plan recognition situations: *covert,* in which the observed actor tries to hide its actions, and *multiple actor*, in which the observed data contains actions of multiple actors. In multiple actor plan recognition, while the actions of some actors may be coordinated towards a shared goal, other actors may have their own independent goals. Further, the evidence of the actions of various actors in general is partially-ordered and interleaved.

Charniak & McDermott (1985) identified plan recognition as an instance of the general abduction task. As with abduction, AI research since then has led to several normative models of plan recognition, including minimal set covering algorithms and logical reasoning from first principles, e.g., (Kautz 1991), and probabilistic inference in Bayesian networks, e.g. (Charniak & Goldman 1993). More recently, Goldman, Geib & Miller (1999) describe a model that also takes causal requirements of plan execution into account.

At present, it is unclear whether or how these normative models of plan recognition might be used for threat analysis. For example, plan recognition in threat analysis takes partially-observed sequences of actions as its input, and desires a specification of actors' goals as part of the output. In contrast, the model of Kautz (1991) requires as input a fully-observed sequence of actions, and does not explicitly represent goals. Charniak & Goldman's (1993) model requires conditional probability distributions between atomic events. While these conditional probability distributions may be available for domains in which large corpus of historical data is available, or constructed for domains which can be easily simulated (e.g., interactive games), it is unclear how we can obtain such conditional probability distributions for threat analysis.

The VAST dataset, for example, contains many routine goals, plans and actions, such as purse snatching, burglaries, and vandalism. For these, we could try to obtain conditional probability distributions from investigative
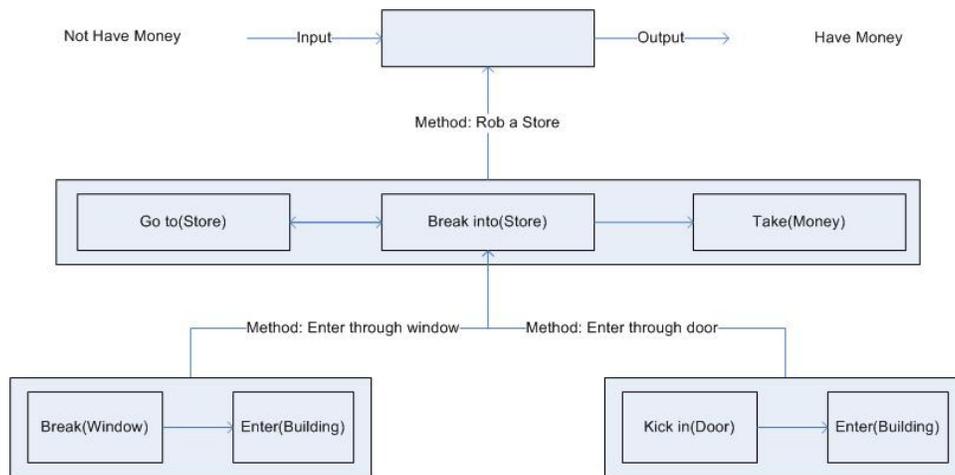
Figure 2. The content and structure of a plan in STAB.

agencies such as the local police. However, the VAST dataset also contains plans for removing a political opponent from an election by digging dirt on the opponent, and suspicions that a scientific laboratory may have released mad cow disease bacteria into a community; the conditional probabilities for such events are harder to find. Of course, we could handcraft and fine tune the conditional probabilities in STAB so that we get the right results for the VAST dataset, but that seems contrived and is unlikely to generalize. Therefore, at present we use heuristic methods for plan recognition in STAB, with the hope these methods will reveal the deep structure of the problem of threat analysis.

Story Understanding: From the early work of Charniak (1977) and Cullingford (1981) to the recent work of Mueller (2004), there has been substantial research on story understanding. There also has been significant work on story generation, e.g., (Meehan 1981). STAB's task, however, is neither story understanding nor story generation. Traditional story understanding programs (such as SAM (Cullingford 1981)) take one story as input and give an understanding of that one story as output. Similarly story generation programs produce one story at a time. In contrast, STAB takes fragments from multiple news stories in the VAST dataset as input, and composes interpretations of the fragments into one or more stories as its output.

A skeletal story represented as a plan in STAB is like a *script* (Schank & Abelson 1976). However, while a script specifies a sequence of actions, a story in STAB also specifies the goals of the action (sub-)sequences and the knowledge state(s) produced by each action. These knowledge states play an important role in the retrieval of stories relevant to a given event. Further, the goals of the action sequences in the stories specify their intent.

Knowledge Representation

STAB contains a library of generic, skeletal scripts relevant to the VAST domain. We found that seven major scripts appear to cover all the illegal/unethical activities in the VAST dataset. We handcrafted this library of scripts into STAB. Figure 2 illustrates a simple script in STAB's library, which is composed of several smaller scripts. The main pattern (in the middle of the figure) is to Rob a Store, which has several steps to it: Go to Store, Break into Store, Take Money. This pattern has the goal of Have Money, given the initial state of Not Have Money (top of figure). Each of the steps in this pattern can (potentially) be done using multiple methods. For example, the step of Break into Store can be done by Entering through a Window or Entering through a Door (bottom of figure). Each of these methods in turn is a process consisting of multiple steps. Figure 3 illustrates a more complex pattern of political conspiracy in which a political figure may get an opponent out of an electoral race by exposing dirt on him or having him assassinated.

The story patterns are represented in the TMKL knowledge representation language (Murdock & Goel 2003, 2007). A task in TMKL represents a goal of an agent, and is specified by the knowledge states it takes as input, the knowledge states it gives as output, and relations (if any) between the input and output states. A task may be accomplished by multiple methods. A method specifies the decomposition of a task into multiple subtasks as well as the causal ordering of the subtasks for accomplishing the task, and is represented as a finite state machine. Thus, the TMKL representation of a plan pattern captures both *intent* and *causality* at multiple levels of abstraction.
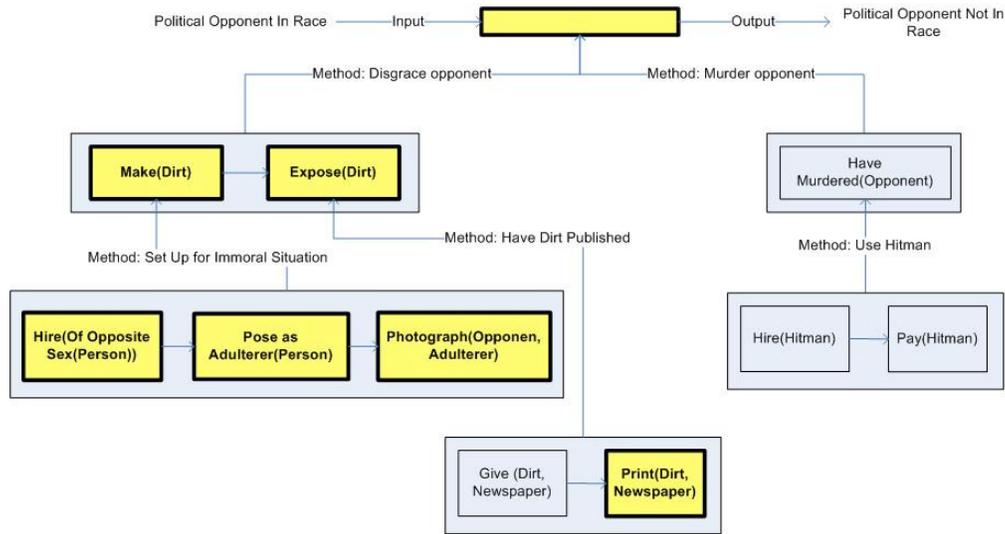
Figure 3: The plan for a political conspiracy intended to remove an opponent from an electoral race.
Activated nodes are denoted by a thick outline around yellow boxes and with bold text.

Although we originally developed TMKL to capture an agent's self-model of its own knowledge, reasoning and architecture, it has since been used in several other applications. TIELT (Molineaux & Aha, 2005), an environment for evaluating learning in computer games, uses TMKL as its agent description language. The AHEAD system (Murdock, Aha & Breslow 2003) uses TMKL for representing hypotheses about asymmetric threats. In particular, AHEAD uses past cases of such threats to generate arguments for and against a given hypothesis. The explicit specification of the goals of subsequences of actions allows AHEAD to retrieve past cases relevant to specific subsequences.

TMKL is more expressive than Hierarchical Task Networks (HTNs) (Erol, Hendler & Nau 1994), but HTN's implicitly provide support for features explicitly represented in TMKL. In a recent experiment, Hoang, Lee-Urban & Munoz-Avila (2005) encoded the same game-playing agent in both TMKL and HTN. They found that "TMKL provides constructs for looping, conditional execution, assignment functions with return values, and other features not found in HTN." They also found that since HTN implicitly provides support for the same features, "translation from TMKL to HTN is always possible." Thus, while TMKL shares the good computational properties of HTNs, it makes reasoning easier and more perspicuous.

Computational Process

Figure 4 shows the high-level architecture of STAB. First, The Evidence Collector collects the input events in an Evidence File in chronological order. Next, the Story

Matcher takes one input event at a time and uses its resulting knowledge state of the event with the task nodes in the TMKL representations of the scripts stored in the Story Library. The Story Matcher tags the matching tasks and passes the matching plans to a Working Memory. Then, the Story Matcher inspects the next input event in the Evidence File and repeats the above process.

If the new input event results in the retrieval of a new script, then the corresponding plan is similarly stored in the Working Memory. If the newly retrieved plan is already in the Working Memory, then additional task nodes that match the new input are also tagged but only one plan instance is kept.

Figures 5 & 6 illustrate the two script plans, Rob a Store and Commit Vandalism, respectively, whose task nodes match the input event Break(Window). The matching task nodes are shown with a thick outline around yellow boxes and with bold text. Note that when a leaf task node in a plan (e.g., Break(Window) in the Rob a Store plot) is activated, then the higher-level task nodes in the plan that provide the intentional contexts for the leaf node (Break into(Store) & Rob(Store)) are also activated.

STAB stores the multiple competing plans (Rob a Store and Commit Vandalism) in its Working Memory and assigns belief values to them. The belief value of a plan hypothesis depends on the proportion of the task nodes in a plan that are matched by the input evidence (higher the proportion, higher is the belief value) and the level of abstraction of the matched task nodes (higher the abstraction level, more is the weight of the node). Equation (1) represents the formula for calculating belief values where level is the depth of the task within the hierarchy of
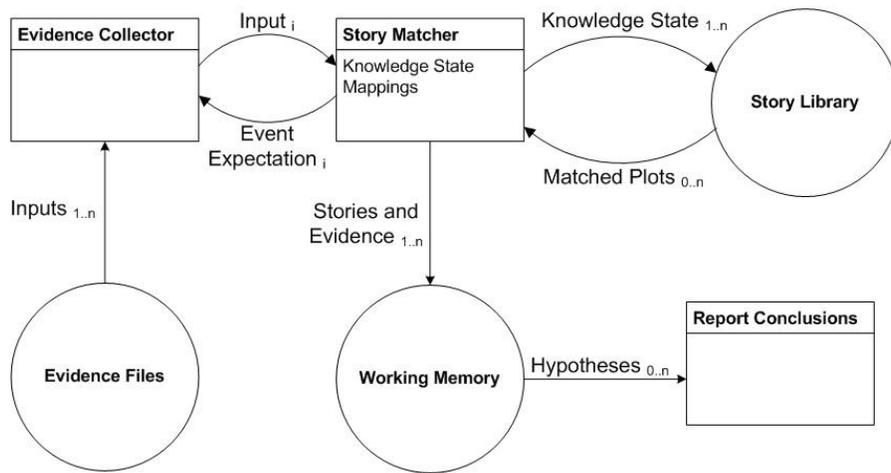
Figure 4: High-Level Architecture of STAB.

the plan and n is the maximum depth of the task hierarchy for the plan. As an example, the belief value for the Commit Vandalism plan (Fig. 6) is (100% / 1) + (50% / 2) = 1.25. Note that only the sub-tree of the method with activated tasks is used in the belief calculation. Similarly, the belief value of the Rob a Store before the Take(Money) node is activated equals 1.33.

$$\sum_{level=1}^{n} \frac{\text{\# activated tasks at level/total tasks at level}}{level} \quad (1)$$

The plan hypotheses in the Working Memory generate expectations. Thus, the Rob a Store hypothesis generates expectations about the events Go to (Store), Enter (Building), and Take (Money), while the Commit Vandalism hypothesis generates expectation about only Kick In (Door). As additional data arrives as input in the Evidence File, STAB matches the data with the expectations generated by the candidate hypotheses. If, for example, the new data contains evidence about Take

(Money), then this node too in the Rob a Store story is tagged, and Equation 1 is used to update the belief value of the hypothesis to 1.50. If the new data contains evidence that contradicts an expectation generated by a hypothesis, then the hypothesis is considered as refuted, and its belief value is reduced to 0.

At the end, STAB generates a report which displays all current plan hypotheses (including refuted hypotheses, if any), the belief value of each hypothesis, and the evidence for and against each hypothesis. Since STAB continually monitors the Evidence File and updates its Working Memory, the user may at any point query STAB to inspect the current hypotheses and the related evidence.

Evaluation

Evaluation of STAB is challenging for two reasons. Firstly, since there is little intelligence data available in the public domain in a suitable form, we cannot directly test
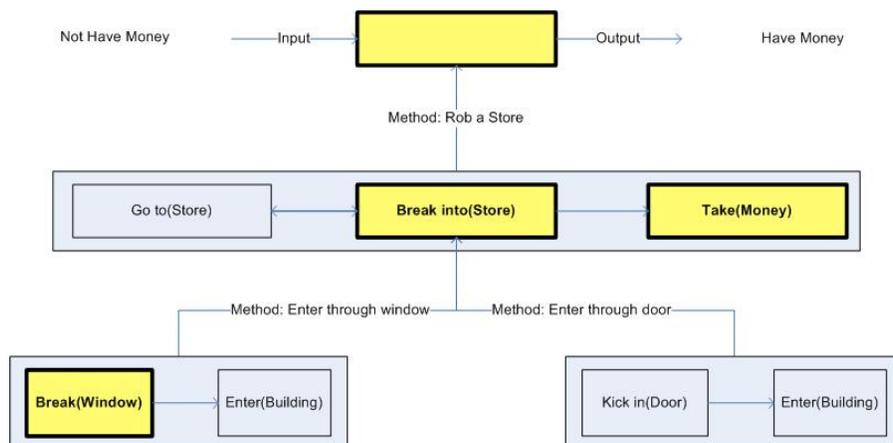

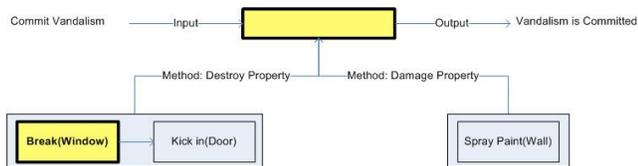Figure 5: The activated nodes in the Rob a Store plan.

Figure 6: The activated nodes in the Commit Vandalism plan.

STAB. Secondly, given the high cost of an analyst's time, evaluating STAB with a human intelligence analyst can be justified and made possible only after STAB is validated as a proof of concept system.

Thus, our evaluation of STAB as a proof of concept system has taken two alternative threads. Firstly, we have demonstrated STAB to an expert who has done significant work with intelligence analysts. This expert found STAB's knowledge representations and information processing as "plausible." However, the expert also raised serious concerns about the usability of STAB's preliminary graphical interface. Secondly, we have begun evaluating STAB for the new VAST-2007 dataset[2] recently released by PNNL. As with the VAST-2006 dataset, we handcrafted representations of events in the VAST-2007 dataset corresponding to illegal/unethical activities. When these events were given to STAB as input, we found that STAB invoked six plans (of the seven stored in its library), three plans with high belief values and the other three with relatively low belief values. Our preliminary analysis suggests that STAB thus makes the right plan hypotheses for the VAST-2007 dataset.

Summary, Conclusions, and Future Work

Psychological studies of intelligence analysis (Heuer 1999) indicate the three main errors made by human analysts in hypothesis generation: (1) Due to limitations of human memory, intelligence analysts may have difficulty keeping track of multiple explanations for a set of data over a long period of time. (2) Analysts may quickly decide on a single hypothesis for the data set and stick to it even as new data arrives. (3) Analysts may look for data that supports the hypothesis on which they are fixated, and not necessarily the data that may refute the hypothesis. STAB attempts to address these limitations. Firstly, there are no limitations on the size of STAB's library or its working memory. On the contrary, STAB offers a non-volatile memory of both generic scripts and specific plan hypotheses. Secondly, for each new additional input event, STAB examines all the scripts whose task nodes match the input. Thus, it is not fixated on any particular hypothesis. Thirdly, STAB explicitly looks not only for evidence that may confirm the expectations generated by a hypothesis but also for evidence that may contradict the expectations.

STAB provides preliminary, but verifiable, answers to two questions arising from Pirroli & Card's model of sensemaking: what is the content of the schemas, and what is process for developing a hypothesis? According to STAB, the schemas are scripts, where the scripts pertain the activities of interest to an analyst, e.g., illegal/unethical activities in the VAST domain. The scripts are represented as plans with goals and states. The TMKL language provides a scheme for representing intent and causality of sequences of events at multiple levels of abstraction. Scripts are retrieved by matching input events with the task nodes in the script representations. Belief value of a plan hypothesis depends on the proportion and level of the matched task nodes. The retrieved stories generate expectations about data, and the confirmation of these expectations results in updates to the belief values.

In its present state of development, STAB has many limitations. Firstly, the input events to STAB are extracted from new stories and represented by hand. We are building a mechanism for automatic extraction of input events from news stories. Secondly, STAB uses a simple, heuristic method for calculating belief values. We are developing a more robust method for calculating belief values based on the principles of coverage and parsimony. Thirdly, STAB does not propagate the values of variables between the nodes in a plan pattern. We are augmenting TMKL to automate variable propagation. Finally, we are conducting usability tests with STAB's interface, and constructing a more usable graphical interface.

Acknowledgements

References

E. Bodnar. (2005) Making Sense of Massive Data by Hypothesis Testing. In *Proceedings of 2005 International Conference on Intelligence Analysis*, May 2005.

T. Bylander, D. Allemang, M. Tanner, J. Josephson. (1991) The Computational Complexity of Abduction. *Artificial Intelligence,* 49(1-3): 25-60, 1991.

---

[2] http://www.cs.umd.edu/hcil/VASTcontest07/

E. Charniak. (1977) Ms. Malaprop, A Language Comprehension Program. In *Proc. Fifth International Conference on Artificial Intelligence (IJCAI-77).*

E. Charniak and R. Goldman. (1993) A Bayesian Model of Plan Recognition. *Artificial Intelligence,* 64(1): 53-79, 1993.

E. Charniak and D. McDermott. (1985) *Introduction to Artificial Intelligence*. Reading MA: Addison-Wesley, 1985.

P. Cohen, C. Perrault & J. Allen. (1982) Beyond Question Answering. In *Strategies for Natural Language Processing,* W. Lehnert & M. Ringle (eds)*,* Erlbaum, 1982.

R. Cullingford. (1981) SAM and Micro SAM. In R. Schank, & C. Riesbeck (Eds.), *Inside Computer Understanding*: *Five Programs Plus Miniatures.* Hillsdale, NJ: Erlbaum.

K. Erol, J. Hendler & D. Nau. (1994) HTN Planning: Complexity and Expressivity. In *Proc. Twelfth National Conference on Artificial Intelligence (AAAI-94).*

A. Goel, J. Josephson, O. Fischer & P. Sadayappan. (1995) Practical Abduction: Characterization, Decomposition and Distribution. *Journal of Experimental and Theoretical Artificial Intelligence,* 7:429-450, 1995.

R. Goldman, C. Geib & C. Miller. (1999) A New Model of Plan Recognition. In *Proc. 1999 Conference on Uncertainty in Intelligence, Stockholm*, 1999.

R. J. Heuer. (1999) *Psychology of Intelligence Analysis*. Center for the Study of Intelligence, Central Intelligence Agency, 1999.

H. Hoang, S. Lee-Urban & H. Muñoz-Avila. (2005) Hierarchical Plan Representations for Encoding Strategic Game AI. In *Proc. Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE-05).*

J. Josephson & S. Josephson (editors). (1994) *Abductive Inference: Computation, Philosophy, Technology*. Cambridge Univerity Press, 1994.

H. Kautz. (1991) A Formal Theory of Plan Recognition and Its Implementation. In *Reasoning About Plans,* J. Allen, H. Kautz, R. Pelavin & J. Tenenberg (eds), Morgan Kauffman, pp. 69-126.

L. Krizan, (1999) *Intelligence Essentials for Everyone*, Joint Military Intelligence College, 1999.

J. Meehan. (1981) TALE-SPIN and Micro TALE-SPIN. In R. Schank, & C. Riesbeck (Eds.), *Inside Computer Understanding*: *Five Programs Plus Miniatures.* Hillsdale, NJ: Erlbaum.

M. Molineaux & D. Aha. (2005) TIELT: A Testbed for Gaming Environments. In *Proc. 2005 National Conference on AI (AAAI 2005),* pp. 1690-1691.

E. Mueller. (2004) Understanding Script-Based Stories using Commonsense Reasoning. *Cognitive Systems Research,* 5(4), 307-340.

J. Murdock, D. Aha & L. Breslow. (2003) Assessing Elaborated Hypotheses: An Interpretive Case-Based Reasoning Approach. In *Proc. Fifth International Conference on Case-Based Reasoning.* Trondheim, Norway, June 2003.

J. Murdock & A. Goel. (2003) Localizing Planning with Functional Process Models. In *Proc. Thirteenth International Conference on Automated Planning & Scheduling (ICAPS'03).* Trento, Italy, June 9-13, 2003.

J. Murdock & A. Goel. (2007) Meta-Case-Based Reasoning: Self-Improvement through Self-Understanding. To appear in *Journal of Experimental & Theoretical Artificial Intelligence*, 2007.

J. Pearl. (1987) Distributed Revision of Composite Beliefs. *Artificial Intelligence,* 33:173-215.

P. Pirolli and S. Card. (2005) The Sensemaking Process and Leverage Points for Analyst Technology as Identified Through Cognitive Task Analysis. In *Proceedings of 2005 International Conference on Intelligence Analysis*, May 2005.

H. Pople. (1977) The Formation of Composite Hypotheses in Diagnosis: Problem Solving and Synthetic Reasoning. In *Proc. Fifth International Conference on AI (IJCAI-77),* pp. 1030-1037.

R. Schank & R. Abelson. (1977) *Scripts, Plans, Goals and Understanding.* Erlbaum, 1977.

C. Schmidt, N.S. Sridharan & J. Goodson. (1978) The Plan Recognition Problem: An Intersection of Psychology and Artificial Intelligence. *Artificial Intelligence,* 11(1-2), 45-83.

J. Thomas & K. Cook. (2005) *Illuminating the Path,* National Visual Analytics Center, Pacific Northwest National Laboratory, 2005.