Playing Detective:
Using AI for Sensemaking in Investigative Analysis

Ashok K. Goel, Summer Adams, Neil Cutshaw and Neha Sugandh

Artificial Intelligence Laboratory & Southeastern Regional Visual Analytics Center
School of Interactive Computing, Georgia Institute of Technology
Atlanta, GA 30332, USA

Abstract: The sensemaking task in investigative analysis generates models that connect entities and events in an input stream of data. We describe two knowledge systems for aiding sensemaking in investigative analysis. The Spade system uses crime schemas to generate an explanatory hypothesis and past cases to validate the hypothesis. The STAB system represents crime schemas as hierarchical scripts with goals and states. It generates multiple explanatory hypotheses for an input data stream containing interleaved sequences of events, recognizes intent in a specific event sequence, and calculates confidence values for the generated hypotheses. We view STAB and Spade as automated cognitive assistants to human analysts: they may support sensemaking in investigative analysis by generating and managing multiple competing hypotheses.

Keywords: Investigative Analysis, Sensemaking, Case-Based Reasoning, Plan Recognition, Hierarchical Scripts, Intelligence Analysis.

This is a Georgia Tech GVU Technical Report (# GIT-GVU-09-03). For additional information, please contact the first author at goel@cc.gatech.edu.

For obvious reasons, intelligence analysis is receiving increasing attention in artificial intelligence (AI) (e.g., Adams & Goel 2007; Jarvis, Lunt & Myers 2004; Klein Moon and Hoffman, 2006; Murdock, Aha & Breslow 2003; Sanfilippo et. al. 2007; Welty et. al. 2005; Whitaker et. al. 2004). Intelligence analysis and other related forms of information analysis, such as investigative analysis, share many common components. One unifying element in the various types of information analysis is the task of sensemaking: generation of a model of a situation that connect entities and events in an input stream of data about the situation (sometimes colloquially called the "connect the dots" problem). The input to the sensemaking task in different types of information analysis is characterized by the same kinds of features: the amount of data in the input stream 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 as rare or novel actions, and the amount of useful evidence typically is a small fraction of the vast amount of data (the colloquial "needle in the haystack" problem). The desired output of the sensemaking task in different types of information analysis too has the same kinds of features: models that explain the connections among the entities and events, specify the intent of the various actors, make verifiable predictions, and have confidence values associated with them.

Psychological studies of sensemaking in intelligence analysis (Heuer 1999) indicate the three main errors made by human analysts in hypothesis generation: (1) Due to limitations of human memory, 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. A technological challenge for AI is to develop techniques and tools that can help analysts overcome these cognitive limitations.

In this article, we briefly describe two knowledge systems for aiding sensemaking in investigative analysis. Spade is a proof-of-concept system that uses semantic knowledge in the form of crime schemas to generate an explanatory hypothesis and episodic knowledge in the form of past cases to validate a generated hypothesis. The STAB (for STory ABduction) system generates multiple explanatory hypotheses for an input data stream containing interleaved sequences of events, recognizes intent in a specific event sequence, and calculates confidence values for the generated hypotheses. We view STAB and Spade as cognitive assistants to human analysts: they may potentially support sensemaking in investigative analysis by generating and managing multiple competing hypotheses.

Spade: Connecting the Dots in a Case Study of Political Assassination

Between December 2000 and April 2001, CNN posted about a dozen news stories on its website reporting developments in the investigation of the murder of Mike Jones (names of all entities have been altered for privacy reasons). Each of the news stories was in English (with no images) and contained a few to several sentences. Only some of the stories contained new facts in the case. These news stories formed the data for the Spade project. (In December 2000, Mr. Jones, the Sheriff-elect of Cross County, was shot 11 times in the driveway of his home. Earlier that year, Mr. Jones had defeated Gary Durant, the incumbent Sheriff, in a close and bitter race. In April 2001, Mr. Durant was indicted for conspiring to murder Mr. Jones. Mr. Durant and two of his deputies were later convicted of the crime.)

Figure 1 illustrates Spade's computational architecture. The evidence files contain sentences from the CNN news stories (see below). The semantic memory contains schemas for a small set of murder

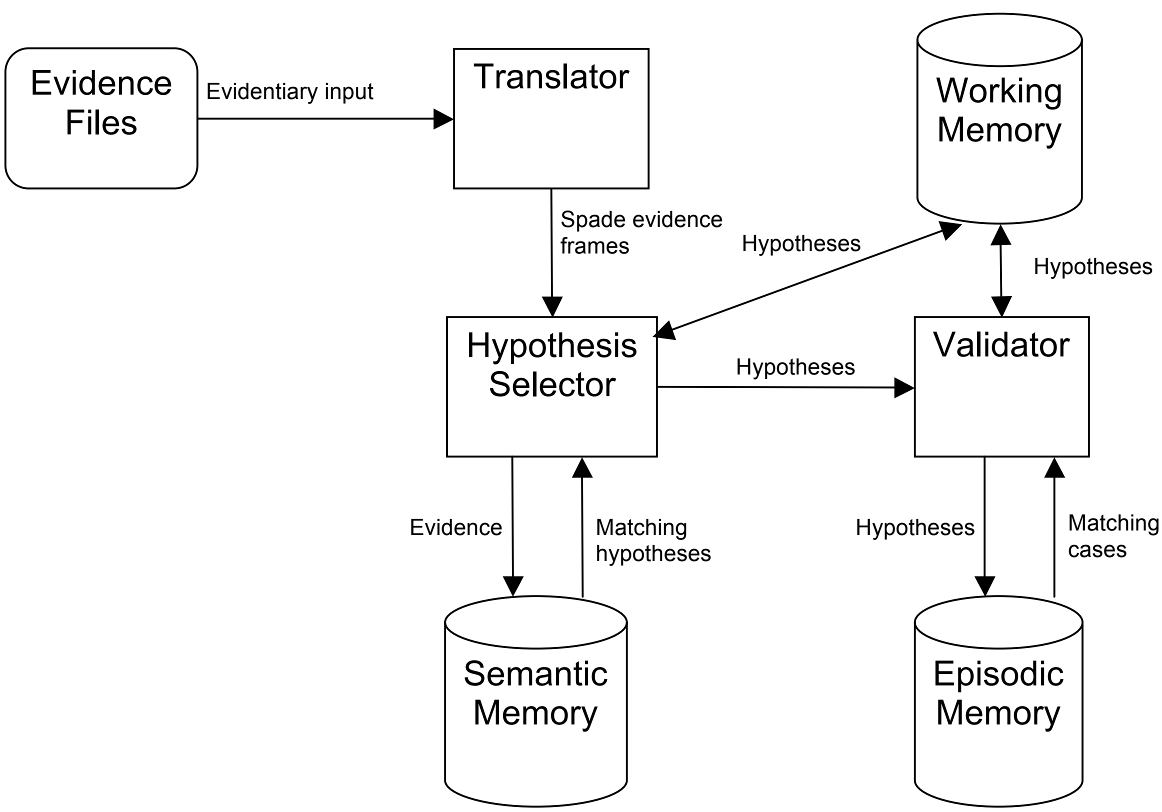

Figure 1: Architecture of Spade

patterns (e.g., crime of passion, organized crime, etc.). The episodic memory contains a small set of past cases of political assassinations. The working memory, initially empty, contains the current explanatory hypotheses. The translator in Spade's architecture extracts information from the evidentiary input. The hypothesis selector selects schemas of murder patterns stored in the semantic memory that match the information extracted by the translator. And finally, once the hypothesis selector has filled the values in the schema and selected a specific hypothesis as its explanation for the evidentiary input, the validator probes the episodic memory to determine if similar cases have occurred in past. If it can find a similar case, Spade views the case as validation of the explanatory hypothesis.

Since our focus in the Spade project was on hypothesis generation in sense making, and not on entity extraction through natural language processing, we manually converted long, complex sentences in the CNN news stories into smaller, simpler sentences. Furthermore, while we retained the chronological order of the data in the news stories, we included only the sentences that contained new data about the investigation. Table 1 illustrates a small sample of the input evidence.

Spade Input
Someone shot MikeJones in his front yard.
MikeJones won an election.
GaryDurant lost that election.
TravisWilson worked for GaryDurant.
MikeJones' occupation was sheriff.
Multiple weapons were used.
MikeJones was shot 11 times.

Table 1: Sample of Evidentiary Input

The translator uses the Link Grammar Parser (http://www.link.cs.cmu.edu/) to transform the input sentences into a murder schema. In particular, the translator extracts entities from the input sentences, places them as fillers in the appropriate slots in the schema, and puts the input murder schema in the working memory. For example, the input sentences "Someone shot MikeJones in his front-yard" and "MikeJones' occupation was sheriff" result in the following input murder schema:

Location: front-yard
Weapon: gun
FrameID: murder
Victim:
  Name: MikeJones
  FrameID: MikeJones
  Occupation: sheriff

Once the input murder schema in the working memory begins to have some entries, the hypothesis selector compares the input murder schema with the schemas for various murder patterns stored in the semantic memory. Based on matches to slots of the input murder schema, the hypothesis selector selects stored murder schemas as candidate hypotheses for the input situation. The selected hypotheses are ranked in order of their confidence values. Each slot in the stored murder schemas has a pre-determined weight. The confidence value of a murder schema is computed by adding the weights of the matched slots.

Each hypothesis is then analyzed by the validator. The validator probes the episodic memory and retrieves cases that are similar to a given hypothesis. The cases in the episodic memory are organized in a discrimination tree. The bottom of Table 2 shows excerpts of a past case, first as text and then as a schema. Retrieved cases are linked to the explanatory hypotheses in the working memory; hypotheses that have no matching cases are discarded.

Spade outputs information about all explanatory hypotheses after processing all evidentiary input. A user may also access the hypotheses by querying the system at any time. Table 2 below shows Spade's output in response to the query command after processing some evidence regarding Mike Jones' murder. Note that the output contains both a confidence value for the hypothesis (the hypothesis score) and the similar case.

```
Working memory contains 1 hypotheses.
Top 1 hypotheses:
Evidence gathered so far:
Suspects: FrameID:  GaryDurant
Events:
  Winner: FrameID:    MikeJones
    Occupation: sheriff
  Loser: FrameID:  GaryDurant
    FrameID:  election
-----------------------------------
Hypothesis -- Political Gain
Hypothesis Score : 2
LossSurroundingVictim:          Power
ResultOfMurder:                 Political_Power_Shift
PotentialLoss:          Enemy
Victim:
  Occupation: sheriff
  FrameID:    MikeJones
PotentialGain:      Power
-----------------------------------
Similar case
Events:
  Winner:        Tim Dunn
  FrameID:       election
  Loser:         Charlie Lewis
Weapons:         gun
Victim:
  FrameID:       Tim Dunn
  Occupation:    legislator
FrameID:         murder frame
LocationOfMurder: home
Murderer:
  Motives:       Eliminate Political Opponent
  Occupation:    assessor
  FrameID:       Charlie Lewis
```

Table 2: Partial output from Spade

Functional Role of Case-Based Reasoning

Spade's knowledge and reasoning is similar to the "data/frame" computational theory of sensemaking (Klein, Moon and Hoffman 2006). However, since Spade is an operational program, it makes much more precise commitments of knowledge and reasoning.

Spade's use of past cases to justify explanatory hypotheses is similar to that of the AHEAD system (Murdock, Aha & Breslow 2003). AHEAD uses past cases to build arguments for and against an explanatory hypothesis given by a user. An interesting issue here pertains to the functional role of past cases in sensemaking in investigative analysis. In investigative analysis, past criminal cases are available in abundance. However, use of past cases for hypothesis generation would require adaptation knowledge in the form of domain models, case-independent rules or detailed explanations of reasoning used in the past cases. It is not evident that this adaptation knowledge is easily available or applicable in investigative analysis. Our work on Spade suggests that instead of hypothesis generation, past cases may be more useful for *post-hoc* validation of a hypothesis generated by some other method of reasoning.

STAB: Finding the Needle in a Haystack of Political Blackmail and Other Crimes

In early 2006, the Pacific Northwest National Laboratories released a synthetic dataset called VAST-2006 (http://www.cs.umd.edu/hcil/VASTcontest06/). This synthetic dataset pertains to illegal and unethical activities, as well as normal and typical activities, in a fictitious town in the United States. It contains over a thousand news stories written in English, and a score of tables, maps and photographs. Figure 2 illustrates an example news story from the VAST dataset. We manually screened the dataset for 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 manually extracted events and entities pertaining to illegal/unethical activities. These events/entities form the input to STAB. We also hand crafted representations for events in terms of the knowledge states it produces. In addition, we examined the maps, photos and tables that are part of the VAST dataset and similarly extracted and represented the relevant information about various entities. Table 3 illustrates a sample of inputs to STAB along with the resulting knowledge state created by an input event.

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 2: Example news story from the VAST-2006 dataset.

Sample STAB Inputs	Resulting State
stolen(money $40 Highway-Tire-Store)	Has-object
cured-disease(Boynton-Labs Philip-Boynton prion-disease)	Is-rich-and-famous
named-after(lab Philip-Boynton Dean-USC)	Expert-involved
was-founded(Boynton-Labs)	Is-open
have-developed(Boynton-Labs prion-disease)	Exists-new-disease
announced-investigation(USFDA Boynton-Labs)	Is-investigating
Injected-cow(Boynton-Labs prion-disease)	Cow-is-infected
treatment-cow(Boynton-Labs prion-disease)	Cow-is-cured

Table 3: Sample STAB inputs

Intent Recognition

In general, the desired outcomes of an investigative case are (1) Models that causally relate entities and sequences of events into coherent stories. (2) 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. (3) Confidence values for the explanations. (4) Explanations that can make verifiable predictions.

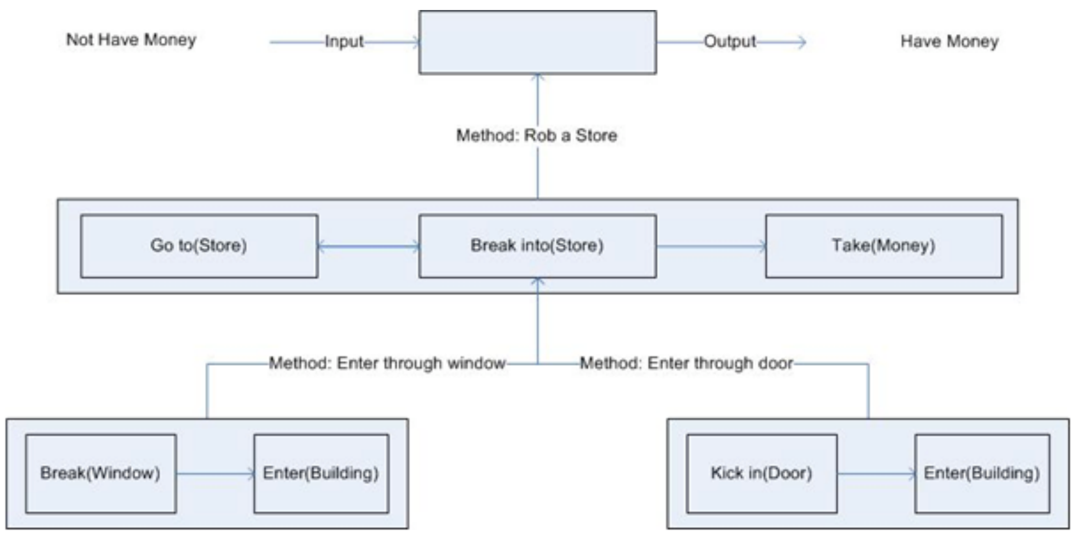

Figure 3: The content and structure of a script in STAB.

STAB contains a library of *hierarchical scripts* relevant to the VAST domain. Unlike traditional scripts (Schank & Abelson 1977), STAB's hierarchical scripts explicitly represent both state and goal at multiple levels of abstraction. While representation of the state caused by an event is useful for inferring causality, representation of goals of sequences of events is useful for inferring intention.

We found that seven hierarchical scripts appear to cover all the illegal/unethical activities in the VAST-2006 dataset. We handcrafted this library of scripts into STAB. Figure 3 illustrates a simple script in STAB's library, which is composed of several smaller scripts. The main script (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 script has the goal of Have Money, given the initial state of Not Have Money (top of figure). Each of the steps in this script 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 4 illustrates a more complex script of political conspiracy in which a political figure may get an opponent out of an electoral race either by exposing dirt on him (political blackmail) or having him assassinated.

STAB's hierarchical scripts are represented in the TMKL knowledge representation language (Murdock & Goel 2001). 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 script captures both *intent* and *causality* at multiple levels of abstraction.

Figure 5 shows STAB's high-level computational architecture. 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.

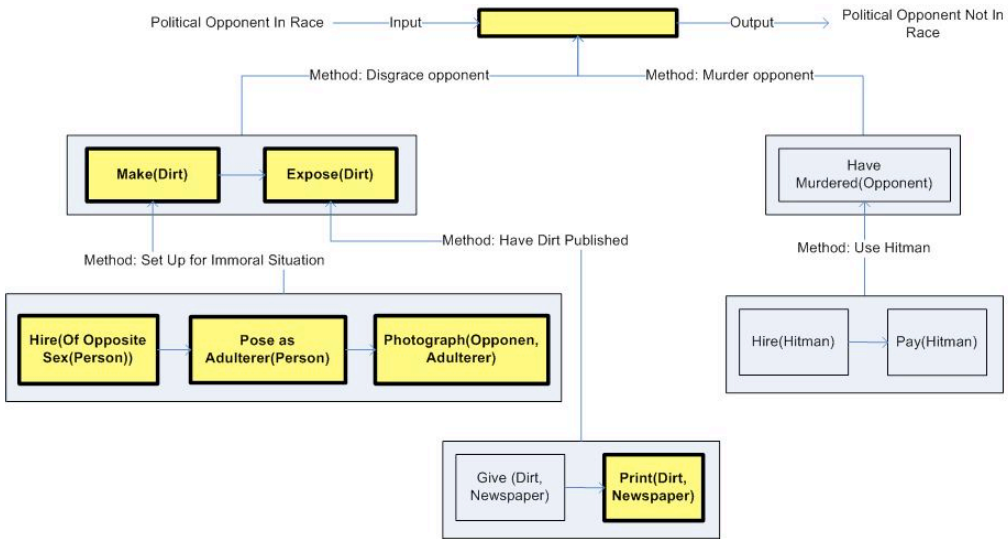

Figure 4: The plan for a political conspiracy intended to remove an opponent from an electoral race. Activated nodes are denoted by a thick outline around yellow boxes and with bold text.

If the new input event results in the retrieval of a new script, then the script is similarly stored in the working memory. If the newly retrieved script is already in the working memory, then additional task nodes that match the new input are also tagged but only one script instance is kept. Figures 6 & 7 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 method that provide the intentional contexts for the leaf node (Break into(Store) & Rob(Store)) are also activated.

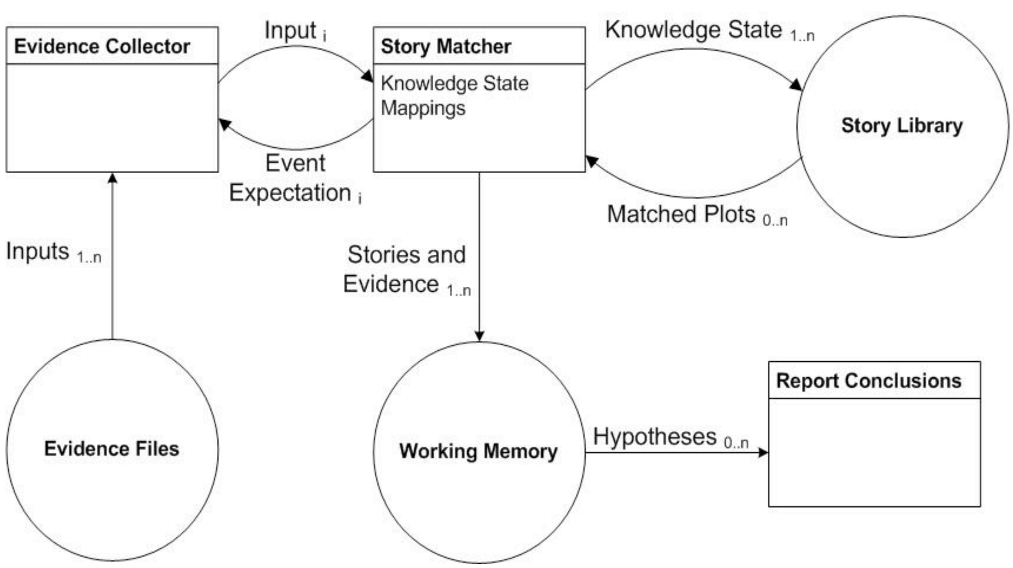

Figure 5: High-Level Architecture of STAB.

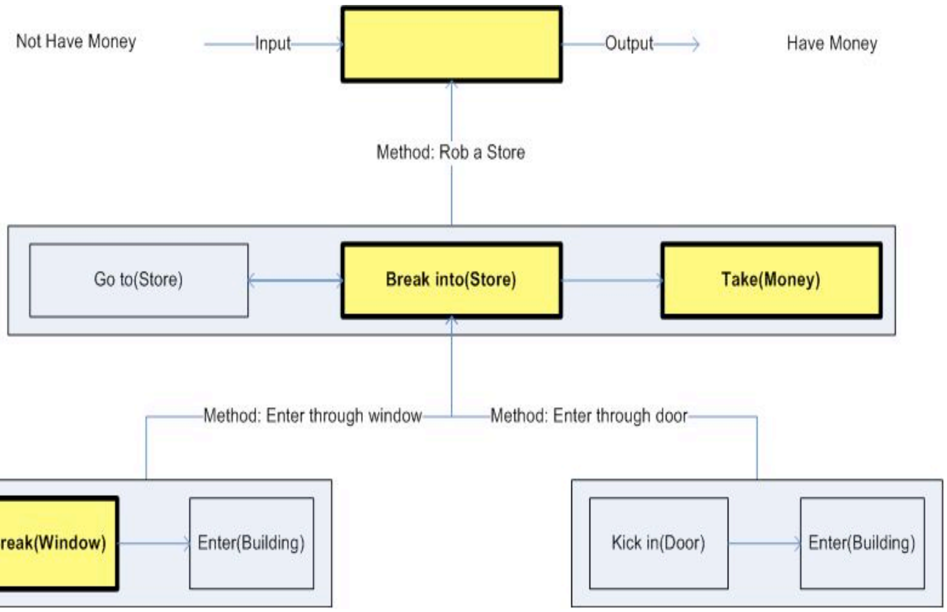

Figure 6: The activated nodes in the Rob a Store plan.

Jarvis, Myers & Lunt's (2004) CAPRe system for intent recognition uses Hierarchical Task Networks (HTNs) (Erol, Hendler & Nau 1994) for knowledge representation. TMKL is more expressive than HTN in part because TMKL enables explicit representation of subgoals and multiple plans for achieving a goal. When Hoang, Lee-Urban and Munoz-Avila (2005) designed a 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."

Confidence Values

STAB calculates confidence values for multiple competing hypotheses based on two criteria (Goel et. al. 1995): *Coverage:* An explanation is better than others if explains more of the observed data, and *Parsimony:* One composite explanation is better than another if it is a subset of the other. STAB stores the multiple competing hypotheses (Rob a Store and Commit Vandalism) in its working memory and assigns confidence values to them. The confidence value of a hypothesis depends on the proportion of the task nodes in its script that are matched by the input evidence (higher the proportion, higher is the confidence value) and the level of abstraction of the matched task nodes (higher the abstraction level,

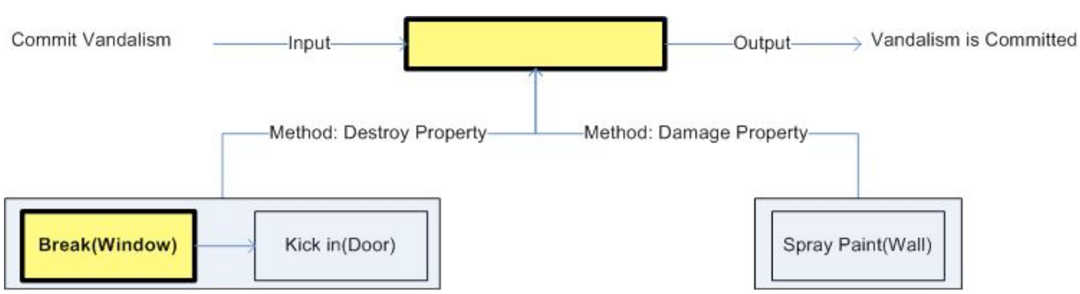

Figure 7: The activated nodes in the Commit Vandalism plan.

more is the weight of the node). Equation (1) represents the formula for calculating confidence values where level is the depth of the task within the hierarchy of the script and *n* is the maximum depth of the task hierarchy for the script. As an example, the belief value for the Commit Vandalism plan (Fig. 7) is (100% / 1) + (50% / 2) = 1.25. Note that only the sub-tree of the method with activated tasks is used in the confidence calculation. Similarly, the belief value of the Rob a Store before the Take(Money) node is activated equals 1.33.

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

The 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 confidence 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 confidence value is reduced to 0.

At the end, STAB generates a report which displays all current hypotheses (including refuted hypotheses, if any), the confidence 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.

Interface

STAB's graphical user interface provides the ability of viewing its scripts in a graphical manner. Analysts can navigate through the scripts and focus on the different levels of abstraction present in the scripts.

The interface shows a list of all the inputs, so that the analyst can select a subset of the inputs and look at the scripts activated by these inputs. For an activated script, the interface also shows which tasks have supporting evidence and which tasks have refuting evidence. Figure 8 shows one such activated script. The interface also enables the analyst to search through the input list for the presence of specific entities.

Evaluation

Our evaluation of STAB has taken two forms. Firstly, we have evaluated STAB for the new VAST-2007 dataset recently released by the Pacific Northwest National Laboratories (http://www.cs.umd.edu/hcil/VASTcontest07/). 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 scripts (of the seven stored in its library), three with high confidence values and the other three with relatively low confidence values. Our own analysis of VAST-2007 dataset (done manually) suggests that STAB generates the right explanations for the VAST-2007 dataset.

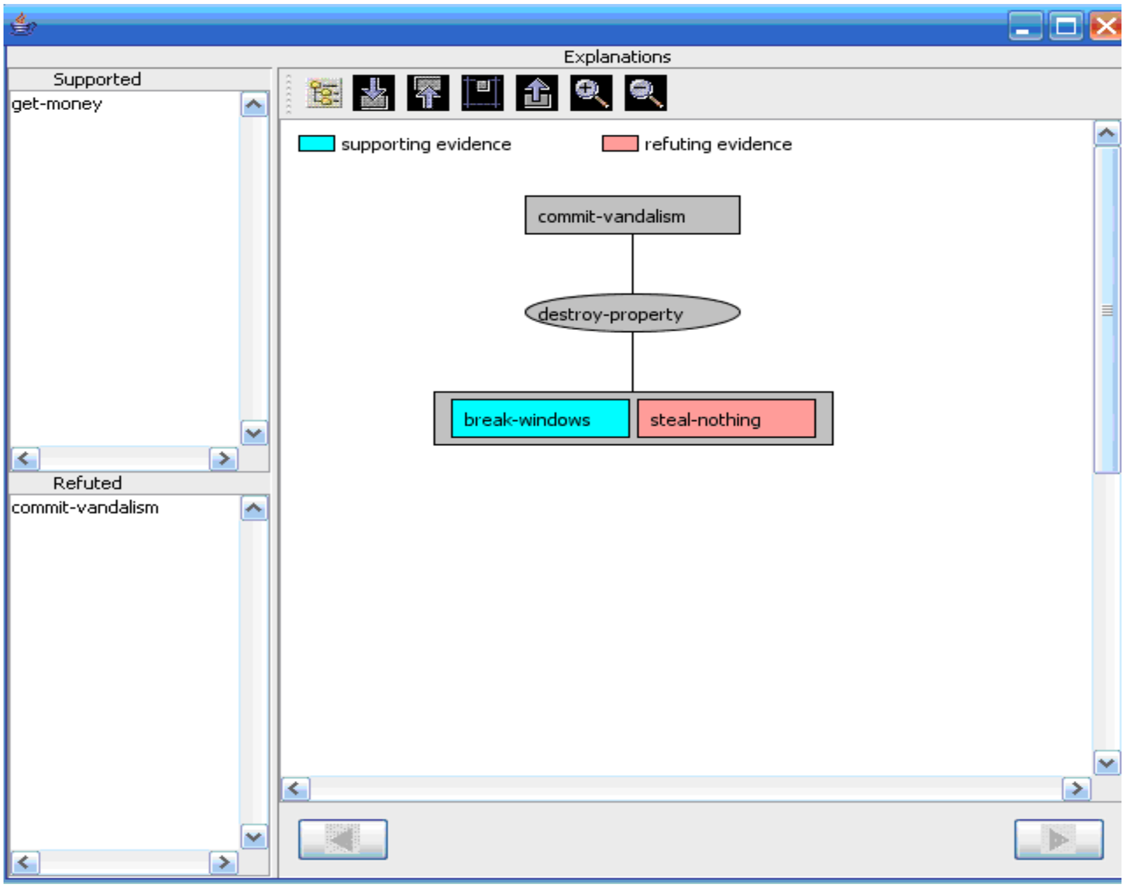

Figure 8: An activated script with supporting and refuting evidence.

Secondly, we have demonstrated STAB to an expert in evaluation of computational tools for intelligence analysis. This expert found STAB's knowledge representations and computational process as "plausible." However, the expert also raised concerns about the usability of STAB as a cognitive assistant to human analysts because of the limited interaction its graphical interface provides. In particular, the expert suggested enhancing potential interaction between human analysts and STAB to include scenarios in which a human may enter a new hypothesis (in the form of a script) into system and ask it to find evidence for and against the hypothesis.

Hypothesis	Activated/Refuted	Confidence Value
Robbery	Activated	2.00
Vandalism	Refuted	
Mad-cow Disease	Activated	1.33

Table 4: Hypotheses activated/refuted using VAST 2007 dataset and confidence in activated hypotheses

Current Work

One of the characteristics of intelligence and investigative analyses is that typically only a small fraction of the data contains evidence relevant to the final explanation. Although the VAST datasets are not necessarily representative of intelligence data, it is instructive to analyze them for the relative proportions of relevant and irrelevant data. Let N be the number of stories in the VAST-2006 dataset, and N_e be the number of evidentiary items (events, entities) in the N stories. Let M be in the number of stories relevant to STAB and N_e be the number of evidentiary items (events, entities) in the N stories. Let M be in the number of stories relevant to STAB and M_e be the number of relevant evidentiary items in the M stories. Our analysis of the VAST-2006 dataset shows that it contains about $N \approx 1200$ news stories, each describing about ten events and related entities (see Figure 2 for a sample story). Thus, the total number of evidentiary items N_e in this dataset is of the order of 10,000 ($N_e \approx 10{,}000$). However, our analysis revealed only about $M \approx 100$ news stories in the dataset related to illegal and unethical activities. Further, the number of new evidentiary items in the 100 stories is only about 1 per story on average so that $M_e \approx 100$. The dataset also contains some maps, photographs and tables that also describe various entities, but this tends to increase N_e (and not M_e). Thus, we estimate that the number of evidentiary items relevant to illegal or unethical activities in the VAST-2006 dataset is less than 1% of all the entities and events in the dataset. Thus, a (second) technological challenge for AI in sensemaking is to find the relevant information in the stream of (largely) irrelevant data.

As we mentioned above, we manually extracted these 100 odd evidentiary items for input into STAB. In current work, we are partially automating this process in a way that filters out data items that are irrelevant to STAB's sensemaking. The basic idea is that the tasks in STAB's hierarchical scripts can be used as "seed events" to focus the search for relevant events (and related entities) in the input data. Thus, for the new VAST-2007 dataset, each news story in is searched for a match with at least one of the seed events. At present this is done by simple string matching. Any news story which has no mapping with the seed events is filtered out. The remaining stories are used to manually generate inputs to STAB. Table 5 illustrates a few sample seed events and a snippet of a story matching the seed event "inject."

<table>
<tr><td>Seed Events</td><td rowspan="7">Seed: inject

Story:
For the study, Dr. Boynton and his colleagues produced prion protein fragments in bacteria, folded them into larger protein structures called amyloid fibrils, and then injected them into the brains of susceptible mice. The mice began exhibiting symptoms of disease in their central nervous systems.………….</td></tr>
<tr><td>steal</td></tr>
<tr><td>break</td></tr>
<tr><td>kick</td></tr>
<tr><td>develop</td></tr>
<tr><td>inject</td></tr>
<tr><td>treat</td></tr>
</table>

Table 5: Sample seed events and a story obtained using a seed event

Of course the above filtering process may exclude some events which are relevant but which have no mappings into the tasks in STAB's scripts (false negavtives). Further some of the events thus included in M_e may actually be irrelevant (false positives). Our hypothesis is that STAB's scripts will enable it eliminate the false positives in later processing: a false positive event will either not activate any of the scripts or activate one with very small confidence. Further, the activated scripts will generate expectations for other relevant events and hence lead to new search for any false positives eliminated in the first round.

We have developed a module which takes the VAST-2007 news stories in unstructured text and generates structured inputs in a form acceptable to the story matcher. Following our earlier work on Spade, this module utilizes the Link Grammar Parser to obtain entities present in a sentence and syntactic roles of the entities like subjects, verbs, objects, etc. The parser also gives links between words representing various syntactic relationships, for example, a link AN connects noun modifiers to nouns. Using rules on these links, the sentences are converted into the needed structured form.

We found that the rules developed for processing the parser output to create the structured inputs left a large room for error (i.e., many false positives and false negatives). There are at least two reasons for these errors. Firstly, it is very difficult to set up a perfect set of rules which deal with all possible syntactic variations of natural language sentences in a correct manner. Secondly, there can be errors in the other language processing stages like that of extracting sentences from documents or the parsing stage. However, we have also found that, for the VAST2007 dataset, the events output by the above module are enough activate appropriate scripts in STAB. Our work is now focusing on making this processing more robust.

Discussion

Our work on STAB and Spade is an attempt to address the first technological challenge for using AI for sensemaking that we mentioned in the introduction: supporting human analysts in overcoming three specific cognitive limitations: (i) limitations on size of memory, (ii) cognitive fixation, and (iii) confirmation bias. Firstly, there are no limitations on the size of STAB's or Spade's knowledge libraries or working memory. On the contrary, STAB offers a non-volatile memory of hierarchical scripts. 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.

To accomplish this, STAB and Spade use knowledge representations and computational techniques that appear especially useful for sensemaking in investigative analysis. In particular, unlike traditional case-based reasoning systems, Spade uses past cases not for generating a hypothesis but for validating a hypothesis generated by a different method of reasoning. Further, unlike traditional scripts, STAB uses hierarchical scripts with explicit representation of goals and states. This enables it to more directly recognize the intent of sequences of actions.

We also noted a second technological challenge for AI in supporting investigative and intelligence analyses: finding relevant information in an input stream of data containing mostly irrelevant information. Our preliminary work on this issue suggests that it may be productive to combine top-down and bottom-up processing: the tasks in STAB's scripts act as seed events for locating relevant information in the input news stories, and the link grammar parser attempts to extract events and entities related to the seed events. A difficulty with STAB's current method for filtering out irrelevant information in its current form is that it appears to lead to the elimination of some relevant information and the inclusion of some irrelevant data.

Finally, it is worth noting a third technological challenge for AI. We view STAB and Spade as cognitive assistants to human analysts. However, the use of cognitive assistants in the practice of investigative or intelligence analysis raises the critical issue of *trust*: human analysts must be able to trust the computational tools (or they will not use the tools). Therefore, an automated agent such as STAB must not only produce accurate results and provide evidentiary support for them, but it also must make its reasoning transparent to the analyst. In future work we will explore how STAB can generate perspicuous explanations of its reasoning.

Acknowledgements

We thank Huzaifa Zafar for his contributions to Spade, Tanvi Bhadbhade for her contributions to STAB, Jean Scholtz for discussions on STAB's evaluation, and Kristin Cook for comments on earlier drafts of this article. We also thank John Stasko, James Foley, Anita Raja and William Ribarsky of the Southeastern Regional Visualization and Analytics Center (SERVAC) for many discussions on sensemaking in intelligence and investigative analyses. Our work on the STAB project was sponsored by the National Visual Analytics Center (NVAC) under the auspices of SERVAC. NVAC is a U.S. Department of Homeland Security Program led by the Pacific Northwest National Laboratories.

References

S. Adams & A. Goel (2007) STAB: Making Sense of VAST Data, In *Proc. IEEE Conference on Intelligence and Security Informatics*, May 2007.

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.

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).*

R. Heuer. (1999) *Psychology of Intelligence Analysis*. CIA Center for the Study of Intelligence.

P. Jarvis, T. Lunt & K. Myers. (2004) Identifying Terrorist Activity with AI Plan Recognition Technology. In *Proc. Innovative Applications of AI (IAAI),* pp. 858-863.

G. Klein, B. Moon, & R. Hoffman. (2006). Making sense of sensemaking II: A macrocognitive model. *IEEE Intelligent Systems*, 21(5), 88-92.

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. (2001) Meta-Case-Based Reasoning: Use of Functional Models to Adapt Case-Based Agents. In *Proc. Third International Conference on Case-Based Reasoning*, Vancouver, Canada, June 2001.

A. Sanfilippo, A. Cowell, S. Tratz, A. Boek, A. Cowell, C. Posse, & L. Pouchard: Content Analysis for Proactive Intelligence: Marshaling Frame Evidence. In *Proc. AAAI 2007*, pp. 919-924.

R. Schank & R. Abelson. *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures.* Hillsdale, NJ: Erlbaum.

C. Welty, J. Murdock, P. Pinheiro da Silva, D. McGuinness, D. Ferrucci, & R. Fikes. Tracking Information Extraction from Intelligence Documents. In *Proceedings of the 2005 International Conference on Intelligence Analysis*, McLean, VA, May, 2005.

E. Whitaker, R. Simpson, L. Burkhart, R. MacTavish & C. Lobb. (2004). Reusing Intelligence Analysts' Search Plans. In *Proc. Annual Meeting of the Human Factors and Ergonomics*, pp. 367-370.