Analogy, Intelligent IR, and Knowledge Integration for Intelligence Analysis

Larry Birnbaum, Kenneth D. Forbus, Earl Wagner, James Baker and Michael Witbrock*

Northwestern University Computer Science
1890 Maple Ave
Evanston IL 60201
{birnbaum, forbus, ewagner, baker} @cs.northwestern.edu

*Cycorp
3721 Executive Center Dr.
Austin TX 78731
mwitbrock@cyc.com

Abstract

Our project is aimed at integrating and extending inferential, analogical, and intelligent IR technologies to create power tools for intelligence analysts. Our goal is to discover interesting and powerful functional integrations that permit these technologies to exploit each others' strengths to mitigate their weaknesses. From the perspective of knowledge-based AI technology, a key goal of the project is to extend the reach of such systems into the world of unstructured data and text. From the perspective of IR technology, it is to leverage the application of inferential and analogical techniques to structured representations in order to achieve significant new functionality.

Background

Intelligence analysts must sift through massive amounts of data, using perspective gained from history and experience to pull together from disparate sources the best coherent picture of what is happening. Intelligent information technology has the potential to create new software tools that could aid analysts in a number of critical and mutually reinforcing ways:

Analysts make heavy use of precedents and analogies. This sometimes leads to vital "trans-logical" leaps. Because of fundamental human cognitive limitations, it also sometimes leads to false analogies, where the matches are too superficial, and to missed opportunities, where the matches are too obscure for unaided human reasoning to uncover. A suite of knowledge-based software power tools could help analysts recognize deeper or less obvious analogies, could help them apply these analogies to the current situation, and could help them reject superficially plausible but useless analogies and precedents more rapidly.

Analysts make heavy use of scenario generation, both to interpret reported data and to project plausible future events. Again because of fundamental human cognitive limitations, the first one or two plausible interpretations (or predictions) are likely to "lock" us into states that make it difficult to generate other likely interpretations (Heuer, 1999). Knowledge-based software tools can mechanically generate (and evaluate for consistency and plausibility) additional interpretations and scenarios, and present these to analysts to help break this logjam.

Analysts are responsible for being aware of and implicitly knowing the entire contents of immense corpora of data, whose volume far exceeds individual human cognitive capabilities. What the media today calls "connecting the dots"—and blames intelligence analysts for failing adequately to do—is essentially a process of deducing logical consequences of several needles spread across multiple haystacks. Knowledge-based software tools can augment information-retrieval tools for finding relevant pieces of information, and moreover for semantically joining or otherwise interrelating them to produce conclusions of interest.

Current technology is capable of providing some of this functionality, but in a limited and piecemeal manner. Knowledge-based systems offer fine-grained and logically coherent inferences and hypotheses—deduction and induction—but only when a sufficiently large fraction of all relevant information is represented precisely (e.g., in formal logic, if-then rules, etc.). Analogical reasoning systems offer the prospect of "thinking outside the box"—but again depend upon structured representations. IR (Information Retrieval) systems can handle the quantity and diversity of unstructured information that exists in the world, but cannot generate new inferences or hypotheses: The lack of structured representations makes it difficult to express and apply the sorts of "transformational" rules that underlie this kind of generative behavior.

Our project is aimed at integrating and extending these three technologies to create power tools for intelligence analysts. Our goal is to discover interesting and powerful functional integrations that permit these technologies to exploit each others' strengths to mitigate their weaknesses. From the perspective of knowledge-based AI technology, a key goal of the project is to extend the reach of such systems into the world of unstructured data and text. From

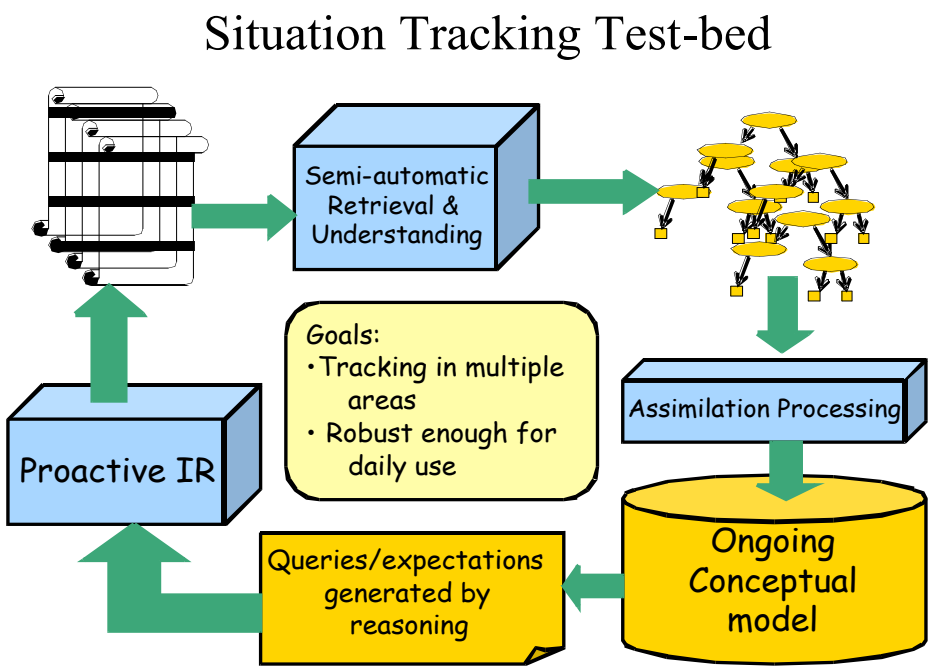

Figure 1: Functional Architecture of Situation Tracking Test-bed

the perspective of IR technology, it is to leverage the application of inferential and analogical techniques to structured representations in order to achieve significant new functionality.

Situation tracking

As a key initial part of the effort described above, we are developing tools to aid intelligence analysts in understanding situations, not just as unstructured sets of documents on the same topic, but also as structured scenarios unfolding over time. Situation tracking in this sense encompasses a meaningful cross-section of the issues and challenges we wish to address. Incoming data feeds must be processed enough to provide useful information both for people and for extending a formally represented model of the situation. Reasoning based on the model, including properties and motivations of sources, can help estimate plausibility and determine how new information should be assimilated. Finally, relational patterns extracted from topic models via inference and analogical processing can be used to generate targeted information retrieval queries to look for "the other shoe dropping."

The overall architecture of the situation tracker is shown in Figure 1. The data feeds are text stripped from web sources, found either by users or as a result of proactive searches by the system itself. The system has some capacity to analyze and represent the content of these texts on its own. We are also developing an interface for interactively translating texts into formal representations, built on Cycorp's knowledge capture tools, including their natural language systems, which rely heavily on clarification dialogues and follow-up questions to facilitate building representations. Finally, the results of these interactions will be formally represented as transformations from the source text to formal representations, and accumulated as part of the system's knowledge.

As new information is discovered and analyzed, it must be assimilated into the system's ongoing conceptual understanding of the system. This involves a combination of first-principles reasoning (for constraint and consistency checking) and analogical processing (to see if it is very unusual in some way compared to previous inputs or to expectations previously generated by the model). Part of the model will be a running set of predictions. These predictions will be checked against incoming information, and indeed will be used to proactively seek relevant information that might bear on them.

Script-based situation tracking

As part of the test-bed we have constructed a prototype situation tracker based on the use of scripts (Schank and Abelson, 1977) as situation models that mediate between and integrate inferential and analogical mechanisms on the one hand, and information retrieval mechanisms on the other. Scripts are explicit, structured representations of stereotypical situations as they unfold over time, e.g., kidnappings. They provide the full inferential power of logical representations, including variables and variable bindings, reasoning over time, and distinguishing among alternative pathways and outcomes. They also serve as a backbone or "glue" supporting a tight link between the IR-based natural language technology and knowledge-based mechanisms, including both Cyc and analogical reasoning. Thus the full power of these mechanisms can be brought to bear on information retrieval, while, at the same time, the IR mechanisms can be used to find information relevant to

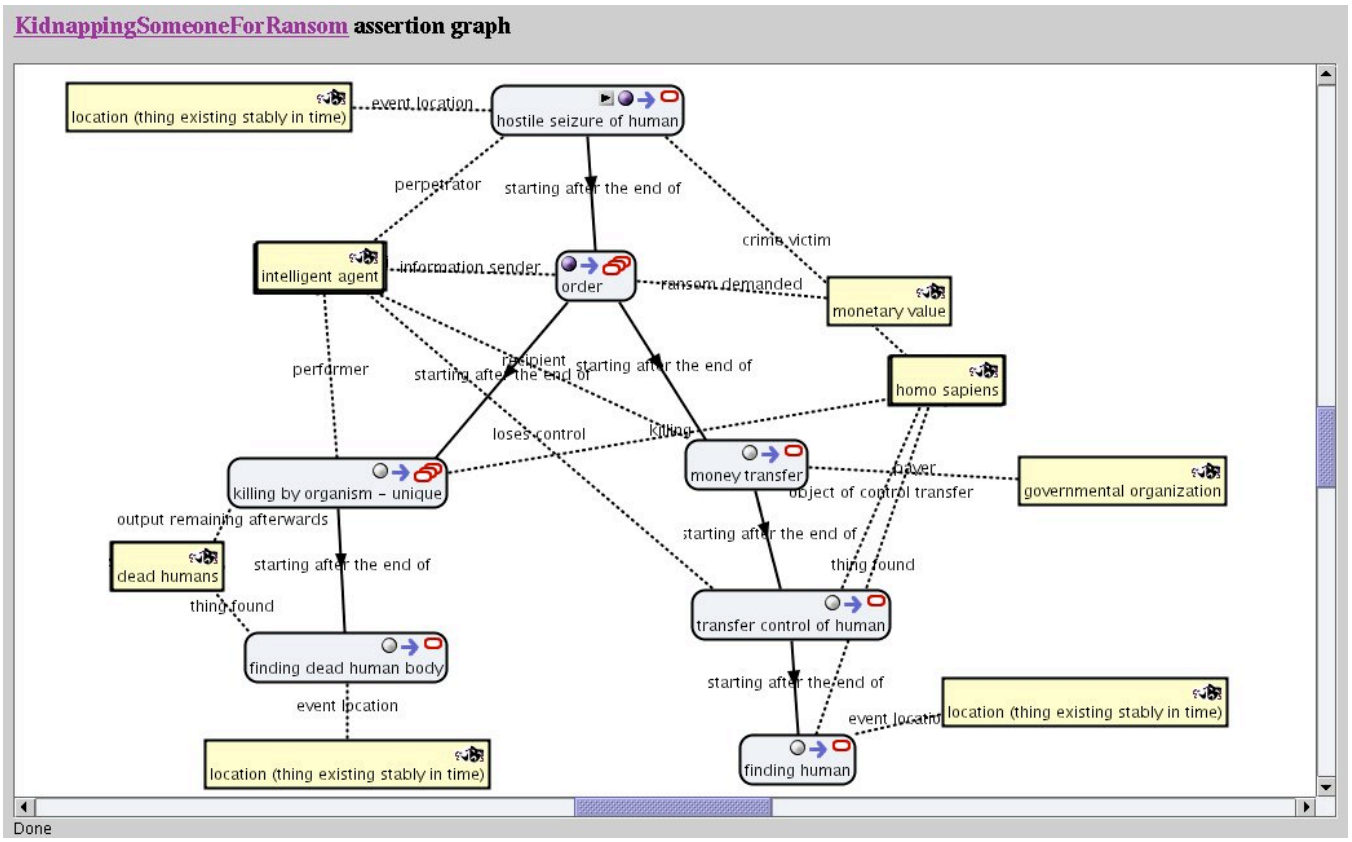

Figure 2: Kidnapping script in CycL

ongoing instances of represented situations. The representation of the kidnapping script employed by the system, in CycL, is shown in Figure 2.

In the current prototype, a user selects and partially specifies a situation to track, e.g., the kidnapping of Nicholas Berg, by selecting the script (kidnapping) and then specifying one or more of its roles (in this case the victim). The assignment of Berg as the victim is propagated to constituent scenes of the kidnapping script, which consist of logical sentences describing the actions that make up a kidnapping, e.g., that the victim is seized by the kidnappers, then taken to a location and held prisoner there, etc. Each scene of the script, as well as the script as a whole, is linked to script- and scene-specific retrieval and recognition rules, parameterized by the appropriate script role variables. These identify and pull out stories related to the situation being tracked, and organize them under the appropriate scene in the script, i.e., determine whether they relate primarily to the initial abduction, to holding the victim prisoner, to announcing the act or demanding ransom, to the release, ransom, or death of the victim, and finally to the escape or capture of the kidnappers. The overall architecture of this component is shown in Figure 3.

Also associated with each scene, as well as the script overall, are extraction rules that try to fill other roles in the script, e.g., location, perpetrators, times, reasons, method of execution, etc. There is some overlap between the scene identification rules and the extraction rules. For example, the rule for identifying a hostage release scene consists of a set of patterns for recognizing descriptions of that event. If the hostage is already known, then these patterns will be parameterized appropriately by the script, leading to highly specific patterns including, e.g., the name or names of the victims. On the other hand, if the victim isn't known, then one of these patterns, upon matching, might provide this information to the script by binding that role variable. As each role is filled, this further parameterizes other recognition/retrieval and extraction rules, increasing their specificity and effectiveness. The output is a fully fleshed-out script, with roles extracted, and stories categorized by scene, presented to the user as shown in Figure 4.

From the perspective of IR, the use of a script in situation tracking, and the propagation of constraints and information among the script's scenes, enables chaining, i.e., the identification and retrieval of stories that are only inferentially related to what was originally specified by the user. For example, if the user specifies a kidnapping situation to track by specifying the kidnapper (e.g., Zarqawi), once a story that mentions a particular victim (e.g., Berg) is found and analyzed, that role will also be bound in an instance of the script. This in turn means that new queries can be launched looking for, e.g., stories describing the scene in which that victim (Berg) was first seized, and these can be retrieved and properly placed in the evolving situation model even if the kidnapper was not

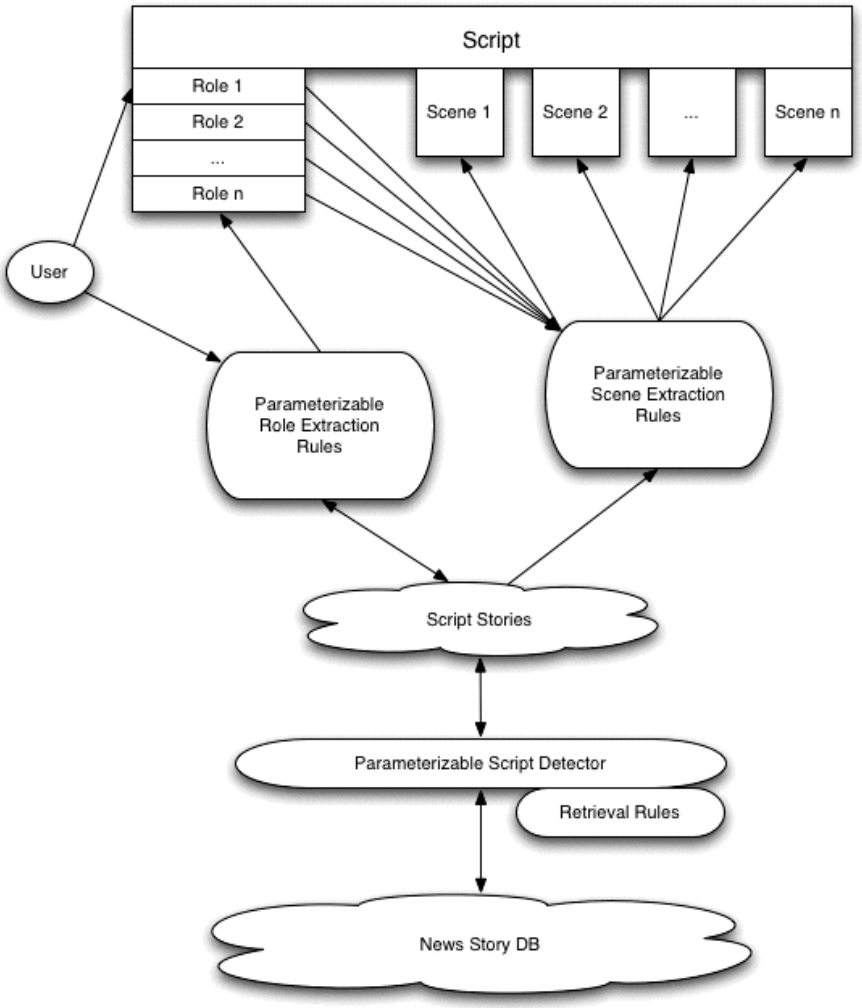

Figure 3: Architecture of script-based situation tracking component

mentioned in those stories, for example because his identity was not yet public knowledge. Moreover in this way additional information, e.g., the location of the victim's seizure, might be extracted. In other words, the inferential capacity of the script (or any other explicit representation of the situation) can be used to automatically parameterize queries that may find related stories even when those stories could not be retrieved given the information originally provided.

Finally, we are currently extending this component of the situation tracker to handle multiple interpretations of the same story. Depending upon the interpretation selected, different recognition/retrieval rules would be invoked and parameterized, and so different texts retrieved and linked to the initial story. We can view the original story as a "point" and the various interpretations of the story as multiple curves intersecting with the point. Then examining an interpretation involves following a curve to view other stories, or points.

Analogical support for analysis

Our next step will be to integrate situation representations constructed by the tracker to analogical mechanisms for assimilation and reasoning support. Building on our current models of the basic components of analogical processing (matching in SME, similarity-based retrieval in MAC/FAC, and generalization in SEQL), we are developing a theory of analogical reasoning based on the retrieval of relevant precedents, support for working out analogies in arguments, and understanding task-specific constraints (see Figure 5).

One aim of this part of the project is to address the "whodunit" problem: given a library of incidents with known perpetrators, and a new incident with an unknown perpetrator, output a small set of hypotheses as to its identity, along with supporting explanations. Our data source will by Cycorp's terrorist KB, which includes about 8,000 terrorist incidents, approximately 800 organizations, and 1500 individuals.

We have developed two alternative analogy-based algorithms for this task. Method one is a version of nearest neighbor or closest exemplar using MAC/FAC to generate possibilities, iterating until enough have been developed (Forbus, Gentner and Law, 1995). The other uses SEQL to develop generalizations of the incidents and then uses the closest generalization(s) to provide hypotheses (Kuehne et. al. 2000). We are currently experimenting with both.

Next steps and conclusion

Earlier in the paper, we mentioned that we are developing an interface for interactively translating texts into formal representations, built on Cycorp's natural language knowledge capture tools, which rely heavily on clarification dialogues and follow-up questions to facilitate building representations. Our intent is to formally represent he results of these interactions as transformations from the source text to formal (situation) representations,

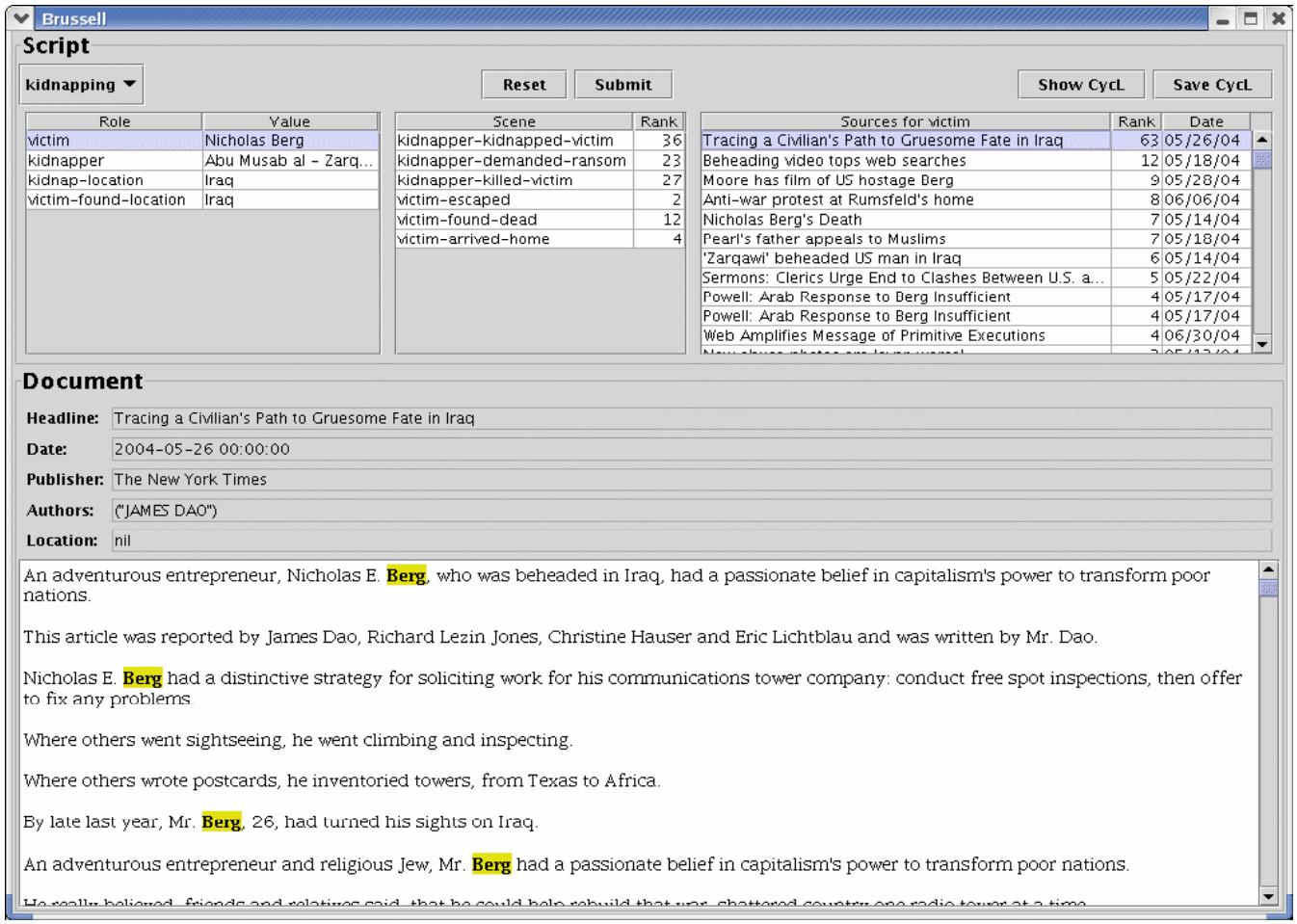

Figure 4: Prototype Script-Based Situation-Tracker interface

and accumulate them as part of the system's knowledge. We see these transformations being used in two interesting ways:

In processing subsequent inputs, these transformations will be applied via analogical processing to suggest how that new document should be understood, in a manner analogous to example-based machine translation (Brown 1996; Somers 2001). Our hypothesis is that this will enable the rate of knowledge entry to be sped up over time, with experience in a situation.

These transformations will provide an input to our intelligent information retrieval system, which will essentially "invert" them to help generate better queries for finding texts that are closely related to something that is formally represented, in particular predictions based on the situation representation. Our hypothesis is that this will enable us to automatically retrieve documents that provide relevant evidence regarding the predictions made by the system.

We are also engaged in a redesign and reimplementation of the script-based component of the situation tracker. At the highest level, we need to do a better job managing multiple instances of multiple scripts, including the ability to merge and split script instances as new information becomes available. Second, we need to better support abstraction in the interface between situation models (scripts) and natural language retrieval or extraction rules, so that the latter can be expressed in terms of entities such as "person descriptions" that can vary flexibly depending upon the available information (name, nationality, gender, occupation, etc.).

In many ways this component of the system resembles script-based text "skimmers" of 25 years ago, in particular Frump (DeJong 1982) and IPP (Lebowitz 1980), and it's worth reflecting on what has changed since then, and what hasn't. The biggest change has been the tremendous development of information retrieval, and to a lesser extent, of textual information extraction (Baeza-Yates and Ribeiro-Neto, 1999). These improvements provide a robust technology substrate that wasn't available two and a half decades ago. On the other hand, the successes of these systems came at a price of decreased semantic sophistication, and, taken together, their strengths and

Exploring analogical reasoning to support analysis

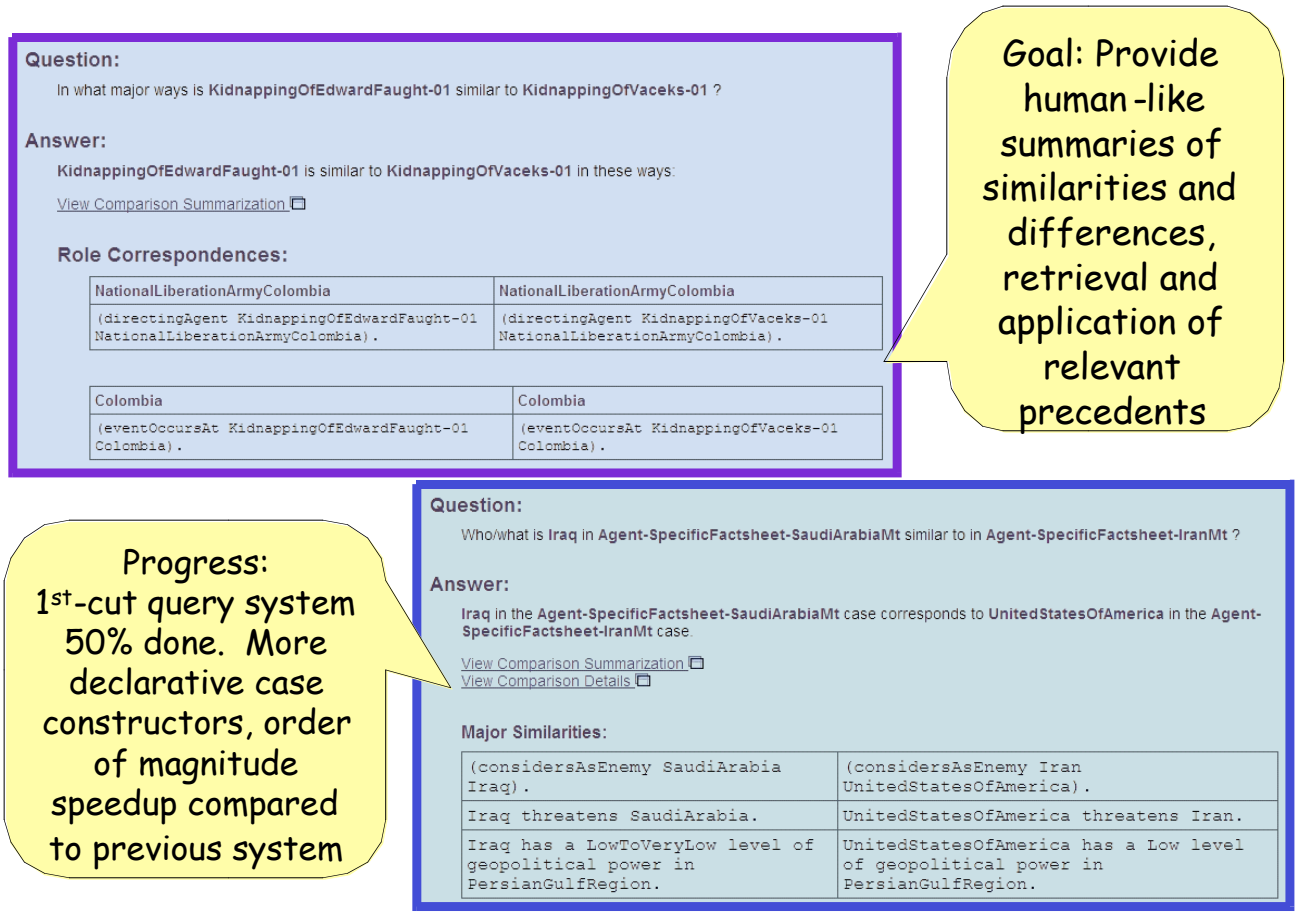

Figure 5: Analogical Reasoning in Support of Analysis

weaknesses provide us a far clearer picture of the role and value of higher-level interpretive mechanisms.

Acknowledgements

This work was supported in part by the National Science Foundation.

References

Baeza-Yates, R. and Ribeiro-Neto, B. 1999 *Modern Information Retrieval*. Addison Wesley.

Brown, R. 1996. "Example-Based Machine Translation in the Pangloss System." In COLING-96: The 16th International Conference on Computational Linguistics, Copenhagen, pp. 169-174.

Forbus, K., Gentner, D. and Law, K. 1995. "MAC/FAC: A Model of Similarity-based Retrieval." *Cognitive Science*, 19(2), April-June, pp. 141-205.

Heuer, R. 1999. *Psychology of Intelligence Analysis*. Washington, DC: Center for the Study of Intelligence, Central Intelligence Agency.

Kuehne, S., Forbus, K., Gentner, D. and Quinn, B. 2000. "SEQL: Category Learning as Progressive Abstraction Using Structure Mapping." In *Proceedings of CogSci 2000*, August 2000.

Lebowitz, M. 1980. "Language and Memory: Generalization as Part of Understanding." In Proceedings of AAAI-80. pp. 324-326.

Schank, R., and Abelson, R. 1977. *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Somers, H. 2001. "EBMT Seen as Case-Based Reasoning." *MT Summit VIII Workshop on Example-Based Machine Translation* Santiago de Compostela, Spain, pp. 56--65.