Anomaly Detection Strategies For Schema-Based Story Understanding

David B. Leake
Department of Computer Science, Yale University

ABSTRACT

Schema-based story understanding allows systems to process routine stories efficiently. However, a system that blindly applies active schemas may fail to recognize and understand novel events. To deal effectively with novelty, a story understander needs to be able to recognize when new information conflicts with its model of a situation. Thus it needs to be able to do anomaly detection.

Anomaly detection is the process that identifies when new information is inconsistent with current beliefs and expectations. Checking for all possible inconsistencies would be an explosive inference problem: it would require comparing all the ramifications of a new fact to all the ramifications of the facts in memory. We argue that this inference problem can be controlled by selective consistency checking: An initial set of inexpensive tests can be applied to detect potential problems, and more thorough tests used only when a likely problem is found.

We describe a set of stereotype-based *basic believability checks*, designed to identify potential problems with minimal inference, and *fine-grained tests* that can be used to diagnose the problems that basic believability checks detect. These tests are implemented in the story understanding program ACCEPTER.

INTRODUCTION

An important issue in story understanding is how to infer the connections between events. To control inferences, many systems rely on schema-based approaches (*e.g.*, [Schank and Abelson, 1977], [Charniak, 1977], [Cullingford, 1978], [DeJong, 1979], and [Lebowitz, 1980]). As long as an appropriate schema is available, a schema-based system needs only to follow the schema's guidance in order to generate the appropriate inferences.

However, no system dealing with real-world situations can have pre-stored knowledge about every possible eventuality. Even if the system's set of schemas *were* complete, the system would often have to select which schema to apply on the basis of partial information— it still would not be assured of activating the appropriate schema. Yet schema-based systems seldom have any capability to recognize when they've gone astray: they cannot detect contradiction of an active schema.

In order to correct erroneous expectations, systems need to decide whether new information is consistent with active expectations and beliefs. Inconsistencies signal the need to revise beliefs and expectations. Unfortunately, complete consistency checking is unfeasible. The only way to catch every conflict is to compare all the ramifications of new information to all the ramifications of the active knowledge, which would be an explosive inference task. For anomaly detection to be feasible, we need ways to limit the effort expended, even though such limits will necessarily mean that some inconsistencies remain unnoticed.

In this paper I argue for a multi-phase approach to anomaly detection. The first phase checks whether a fact is expected or already known; if so, the fact is accepted. Otherwise, the fact is compared with stereotypes; conflicts are considered anomalous. These comparisons with stereotypes constitute *basic believability checks*, which are efficient enough to be applied to each input, but are also likely to detect a large proportion of problems. When stereotype conflicts are found, *fine grained* checks are used to decide the severity of the problem, and to characterize the difficulty in more detail.

The following sections discuss a set of basic believability checks, and fine-grained tests that can be applied when the basic checks detect problems. Both sets of checks are implemented in ACCEPTER, a story understanding program that detects anomalous events in stories and evaluates candidate explanations for them ([Leake, 1988a], [Leake, 1988b], [Kass and Leake, 1988]).

ACCEPTER

ACCEPTER is a story understanding program that detects anomalies and evaluates candidate explanations for them. Its domain is incidents of death and destruction; the stories it processes include the premature death of the racehorse Swale, the death of basketball star Len Bias, and the explosion of the space shuttle Challenger.

ACCEPTER understands routine events in terms of expectations given by prestored schemas; its understanding process is loosely modeled on that of SAM [Cullingford, 1978]. However, it supplements that understanding process with the following three-step anomaly detection procedure:

1. **Compare input to expectations and prior beliefs.** If they match, no further checks are needed; conflicts are anomalous.
2. **Do basic believability checks.** These checks are coarse-grained tests that compare aspects of an action against standard patterns. The basic checks identify potential problems, while requiring minimal inference.
3. **If standard patterns are not consistent with an event, use fine-grained tests.** More detailed tests can focus on the aspects of the event that conflicted with the pattern, in order to diagnose the problem further.

The sections below examine phases 2 and 3 of this process. Comparison of facts to expectations is described in [Leake, 1988a].

BASIC BELIEVABILITY CHECKS

The idea underlying ACCEPTER's routine verification level is analogous to the idea of basic level categories [Rosch *et al.*, 1976]: that there is a level at which anomaly detection maximizes the amount of return per unit of effort. I call tests at this level *basic believability checks*. Although finer-grained checks might detect additional anomalies, they give proportionately less return, since they need to check specialized aspects of the situation that are irrelevant to many situations.

In order to detect problems efficiently, ACCEPTER relies on comparisons with stereotyped patterns. Such patterns are usually only viewed as ways to characterize routine objects or situations. (For example, knowledge structures such as *Memory Organization Packets* (MOPs) can organize standard event sequences in memory [Schank, 1982], [Kolodner, 1984].) However, seeing how well inputs agree with stereotypes is also useful for monitoring the expectations guiding understanding. Conflicts with stereotypes suggest that the wrong knowledge structures are being used to understand an event; similarity to stereotypes suggests that the active knowledge structures are reasonable.[1]

Using patterns for anomaly detection raises two questions: *what types of patterns are important to check?* and *how are the relevant patterns accessed and applied?*. We answer these questions in context of basic verification in ACCEPTER.

Which Patterns to Check

ACCEPTER finds potential anomalies by comparing events to four kinds of stereotypes:

Event sequence patterns: ACCEPTER uses MOPs to represent stereotyped expectations for

[1]This is similar to the *representativeness* heuristic discussed in [Kahneman *et al.*, 1982]. People are likely to accept statements of category membership, if observable features of the object match the stereotypes for category members.

the events that occur in a given context. For example, the MOP for dining in a restaurant includes knowledge about the normal sequence of events in a restaurant meal (the customer enters, is seated, orders, eats, etc.), and about the normal temporal separation between these events. Conflicts such as premature events (*e.g.*, being given food without being seated) are anomalous, and may show that the restaurant MOP does not apply. For example, the customer may be picking up a take-out order, rather than eating there.

Normative role-filler types: For each role in its MOPs, ACCEPTER maintains information on the types of objects that usually fill the role. For example, the roles in the restaurant MOP include the diner, the waiter, and the customer's order. Normally, the customer and waiter are both human, and the order is for food. (These types are only normative: the waiter might actually be a robot, or the diner could be a pampered pet.) ACCEPTER checks each role-filler in a new action to see if it is a novel type of filler. If not, an anomaly is noted.

Again, explaining the anomaly may show that the wrong MOP is being applied. For example, if someone enters a fast food restaurant, we normally assume that he'll eat there. But if he buys a pack of cigarettes instead of food, we'll retract our assumption that he entered for a meal.

Class limitations: Objects of a subclass may have features or functional limitations that are unusual, compared to most members of the class to which they belong. For example, if our stereotyped view of expensive sports cars is that they are fast and handle well, but we find that cars of brand X have bad handling, this deficiency makes it anomalous for the car to be used in a situation where handling is very important (such as a prestigious race). Noticing the limitation problem may make us change our expectations. For example, if we expected the car to win because it had a good driver, we might want to change the assumption in view of the car he's using.

Decision patterns: ACCEPTER represents knowledge about the types of actions that an actor favors, and avoids. For example, fraternity members seek out wild parties; athletes in training are supposed to avoid them. If an actor participates in an unusual action— for example, an athlete goes to a party the night before a big game— his behavior is anomalous, and related expectations may need to be changed.

How the Patterns are Organized and Accessed in Memory

ACCEPTER's memory is organized in an abstraction net. For example, abstractions of the MOP for running in a race include athletic competition and exercise; abstractions of racehorse include horses (which have the abstraction of living things) and valuable objects. Patterns are indexed under nodes in this hierarchy.

The scenes of a MOP are stored under the memory node associated with that MOP, as are normative filler types for the MOP's roles. (Both can be inherited from abstractions, if no information is indexed under the specific MOP.) Decision patterns are indexed under actors they involve. When ACCEPTER retrieves patterns to apply, it attempts to retrieve the most specific relevant patterns.

The patterns are applied as follows. Given an action that is hypothesized to fit within a MOP or plan, ACCEPTER matches the action against the packaging structure's expectations, in order to identify problems such as premature events, delayed events, or missing events. Restrictions on role-filler types are retrieved from the MOP or its abstractions, and the system compares hypothesized role-fillers to them. More specifically:

Applying normative-filler information: For each role in the input, ACCEPTER retrieves role-filler patterns for that role. These patterns are stored in memory under the action, indexed by the role. If ACCEPTER fails to find a pattern under the current action, it does a breadth-first search up the hierarchy of abstractions for the action, checking each to see if a relevant pattern is indexed under it.

Applying class limitations: Normative types for a role-filler are also used to guide the search for limitations of particular objects that might interfere with their ability to fill a role. Guiding the search for limitations is important, since any object can have many abstractions, and can have limitations compared to any of its abstractions. To restrict its consideration to features that are relevant in the current context, ACCEPTER only considers limitations of the filler *compared to the normative role-filler*. For example, if a hypothesized car theft is being evaluated, the normative role-filler type for the object of the theft is *valuable object*. Compared to other valuable objects, a brand X sports car might have no limitations, so the theft would have no basic-level problems. However, if a hypothesized automobile race were being checked for anomalies, and *sports cars* were the normative filler type for vehicles involved, a different limitation— that cars of brand X had bad handling, compared to other sports cars— might be retrieved. This limitation would be flagged as something to check with fine-grained tests— would the bad handling make it unable to perform in the race?

Applying decision patterns: For each actor involved in an action, the system tries to retrieve actor decision patterns that are relevant to the actor's involvement. These patterns may be indexed under the actor (*e.g.*, we know that John loves eating at MacDonalds), or under the actor's abstractions (*e.g.*, John is a taxi driver, and we know that taxi drivers often eat at MacDonalds). It is sometimes necessary to abstract both the actor and the action involved: we might know that taxi drivers often eat at fast-food restaurants. These patterns can show that a decision is believable. (When retrieving decision patterns, ACCEPTER does a breadth-first search over abstractions of the actor and action, to try to find the most specific pattern that is relevant to an actor's decision.)

FINE-GRAINED CHECKS

Basic-level checks give little information about the reasons for problems: they simply identify that something is unusual. When ACCEPTER encounters a potential problem, it diagnoses the problem more specifically by applying fine-grained checks. These checks give a more specific problem characterization, which points to specific information that an explanation must address.

ACCEPTER uses three types of fine-grained checks. They are *basic action decomposition*, which evaluates a role-filler's ability to perform in a role, and two motivational checks: *examination of direct effects*, which is used to check whether an action is consistent with actor goals, and *plan choice checks*, which see if the actor's plan choice conflicts with our model of his plan preferences. Thus when an action is unusual for an actor, ACCEPTER checks three things: whether the actor probably could have performed the action, whether the action was consistent with his goals, and whether the action was consistent with his planning style. Although these checks are more expensive to apply than pattern-based checks, the inferencing they involve is still limited.

Basic Action Decomposition

The primary purpose of basic-action decomposition is identifying causal problems that might result from an unusual action, making it even more anomalous. Basic action decomposition takes a composite action, decomposes it into its constituent parts, and checks any restrictions associated with those parts that are relevant to the unusual role-filler.

For example, anyone who participates in jogging is usually human. Either a fish, or a monkey, would violate this stereotype, so a hypothesis that either one was jogging would be anomalous. However, there is a difference in the seriousness of the problem. There's no reason why a monkey couldn't perform the actions of a jogger; basic-action decomposition shows that the only problem would be that a monkey wouldn't have the normal goal of physical fitness that drives joggers. Consequently, an explanation would have to show an alternative goal being served— perhaps the monkey was owned by someone who gave him a banana every time he jogged a quarter mile. Basic action decomposition shows more severe problems for the fish: for example, since it doesn't have legs, it couldn't even perform the actions involved.

In addition to showing the severity of a problem, basic-action decomposition helps focus explanation: if causal problems are found, explanation should focus on how they might have been overcome. For example, we might have the stereotype that people who fly first class are usually businessmen. If a non-businessman flies first class, basic believability checks detect the stereotype conflict. However, the characterization "a non-businessman flying first class" isn't very helpful in finding an explanation. If we look at the detailed requirements for flying first class, we can identify more specific problems, which can then be addressed by an explanation. For example, one specific requirement is paying for an expensive ticket. If we find that the person flying didn't have much money, we can describe the anomaly more specifically as "how could he do something he couldn't afford?" This characterization gives more guidance to the explanation process. For example, by looking for ways he could have recently obtained money, we might find that he'd just won a lottery, or received a substantial raise.

Examination of Direct Effects

Examination of direct effects tests whether the unusual action is consistent with an actor's goals. If an action doesn't fit an actor's usual behavior patterns, we can ask whether the action has bad effects. If so, explanation should show how other goals took precedence over the normal reasons for avoiding the action.

However, we cannot hope to detect all possible bad effects of an action: for any action, there would be infinitely-many effects to check. Nevertheless, simply checking an action's direct effects can give an indication of whether the action undermines the actor's goals. For example, direct effects of buying a car include having possession of the car, and having less money.[2] If we know that someone is avoiding buying a car, and he buys it anyway, we can compare his goals to the direct effects of buying the car. If he's trying to save money for college, we can reformulate the anomaly as spending money, which conflicts with his goal to save money. Our explanation would then focus on the tradeoff between having the car or having savings.

ACCEPTER identifies goal problems by checking actions that conflict with their actor's behavior patterns, to see if the actions' direct effects undermine the actor's goals (either specifically known for the actor, or inherited from abstractions). For example, when the system evaluates the explanation that the racehorse Swale was poisoned by his owner, basic checks show that the poisoning is something his owner would have been expected to avoid, since it's illegal. This prompts ACCEPTER to check the direct effects of the action, to see if they account for the poisoning. The only direct effect it finds is that the owner's property is destroyed, which conflicts with the businessman's theme goal of increasing wealth, so the poisoning is anomalous.

Plan Choice Checks

Plan choice checks compare an unusual plan with the types of plans that are typical for the actor, in order to see if the plan is consistent with his planning patterns. In general, it could be very difficult to decide if a plan matches the planner's planning patterns, simply because it is hard to predict a plan's possible ramifications. However, actors often use standard plans, for which relative advantages and disadvantages are known in ad-

[2] ACCEPTER's characterization of direct effects is quite arbitrary: it considers direct effects to be the effects that are reached by short inference chains (in the current implementation, chains of length 4 or less).

vance.

When we use standard plans, or observe others using them, we gather comparative information about alternative ways of accomplishing a goal. For example, we learn the likely cost of the plan, whether the plan is likely to fail, and how efficient it is compared to plans we have used before. We can use this knowledge to decide between plans. For example, suppose bus service is a cheap but unreliable way to travel, and taxis are more expensive, but more efficient and reliable as well. If we lack money and punctuality is unimportant, we'll choose the bus; otherwise we'll prefer taxis.

Knowledge of plan characteristics also helps us predict the actions of others. If we know the goal orderings of different actors, we can anticipate their priorities, and expect their plans to reflect their priorities— both in the goals for which they plan [Carbonell, 1979], and in the plans that they select for a given goal. For example, an impatient executive puts a low priority on money, but a high priority on saving time. From this, we might expect him to fly on the Concorde when he goes to Europe.

To reason about plan selection, we need to be able to characterize plans along the dimensions that affect people's plan choices. One way to start is to look at stereotypes about priorities, and to translate them into parameters for characterizing plans. For example, the stereotype of a miser directs the choice of the lowest-cost plans possible. To know which plans a miser is likely to pick, we need to have an estimate of their relative monetary costs. People who are impatient give time a high priority; thus relative speed of a plan is important. Cautious people avoid risk; thus risk must be represented also. Once the dimensions are specified, we can represent both general preferences, and those that only apply in particular domains. For example, someone might avoid risk in relationships, but be indifferent to financial risks. The most important differences between plans can be characterized along four basic dimensions:

- **Reliability** of a plan for accomplishing goal.

  Reliability can be either absolute (*e.g.*, this plan always works) or relative to other plans for the goal (*e.g.*, this treatment has only a 50 percent success rate, but it's still the best way we know of to deal with cancer.)

- **Risks** of using the plan.

  This is a characterization of the possible bad side effects of the plan.

- **Cost** compared to other plans for the same goal.

  This can be characterized along standard dimensions, such as those described in [Wilensky, 1978]: time, consumable functional objects (like money), nonconsumable functional objects (like a stove), and abilities.

- **Yield** compared to other plans for the same goal.

  If the effects of a plan can be measured along a scale, yield can be used to compare the effectiveness of the plan. For example, having a paper route and being a lawyer are both plans for making money, but the yield of the paper route is low, while the yield of law is high. Yield could also be balanced against costs, to determine a plan's efficiency.

Table 1 shows how these dimensions can be used to characterize bank robbery and medical school as plans for getting money.

In order to decide if the type of plan is anomalous, ACCEPTER compares the plan's dimensions to the values that the actor usually favors. It determines the plan parameters by retrieving and applying a procedure for generating the plan dimensions of any instantiation of the plan. (The procedure is indexed under the plan, or one of its abstractions, in ACCEPTER'S memory.) A question for future research is how a system might learn the parameters for a plan, based on observation of use of that plan and alternative ones.

A PROGRAM EXAMPLE

The output below is an edited trace of ACCEPTER applying plan choice checks. The sys-

Bank robbery as a plan for A-wealth

Dimension	Value
Reliability	LOW
Risks	HIGH
Cost	LOW
Yield	HIGH

Medical school as a plan for A-wealth

Dimension	Value
Reliability	NORMAL
Risks	LOW
Cost	HIGH
Yield	HIGH

Table 1: Plan choice dimensions of two plans.

tem has previously processed the explosion of the space shuttle Challenger, and is given as input the conjecture that Russia sabotaged it.

Sabotage doesn't fit standard patterns for Russia, so ACCEPTER does fine-grained checks. It finds no problems with basic-action decomposition, and the effect of the plan— harming the U.S.— is consistent with its picture of Russia's goals. To find if the type of plan matches Russia's policies, it checks the plan parameters of sabotage. It determines that sabotage against the U.S. is risky, because of possible retaliation. Since Russia is not especially prone to risky plans, sabotage seems unlikely.

Checking if "RUSSIA'S SABOTAGE" is believable.

Searching for plan dimension generator for RUSSIA'S SABOTAGE.

... No generator stored under SABOTAGE.

Searching for plan dimension generator under abstractions of SABOTAGE.

... Generator found under abstraction VIOLENT-ACTION.

Applying generator to check RISK of RUSSIA'S SABOTAGE. Description of test:

> "Risk of VIOLENT-ACTION depends on comparative PHYSICAL-STRENGTH of actor and victim."
>
> For countries, PHYSICAL-STRENGTH specifies to MILITARY-STRENGTH. Comparing....
>
> RUSSIA inherits HIGH as its MILITARY-STRENGTH, from abstraction INDUSTRIALIZED-COUNTRY.
>
> USA inherits HIGH as its MILITARY-STRENGTH, from abstraction INDUSTRIALIZED-COUNTRY.
>
> ... STRENGTH is the same.

... Risk is HIGH.

Comparing to planning tendencies for RUSSIA.

PLAN-SELECTION problem: "RUSSIA'S SABOTAGE" CONFLICTS-WITH planning tendencies for RUSSIA, due to HIGH RISK.

When ACCEPTER evaluates the explanation that Challenger was sabotaged by Libya, ACCEPTER considers that explanation more likely, because Libya often uses risky plans.

CONCLUSION

In order to maintain an accurate picture of the world, and to be able to learn from novel situations, story understanders need to be able to detect anomalies. Since anomalies could arise at any point in a story, anomaly detection needs to be applied as a routine part of understanding. Consequently, the anomaly detection process must be efficient.

ACCEPTER reduces the cost of anomaly detection by having two levels of tests for how well new information fits prior beliefs. Comparatively inexpensive tests are applied to all inputs. These basic believability checks will not detect all problems, nor will the problems they identify always be significant, but they detect many of the situations that are likely to need explanation. When basic believability checks detect potential problems, fine-grained tests are used to do more careful analysis.

By first comparing an input to specific beliefs in memory and active expectations, and then verifying in terms of the patterns described above, ACCEPTER can often detect potential problems without doing extensive inference. Only when potential problems are detected does it apply more costly checks.

A topic for future research is how verification should change to reflect the importance of certainty in a given situation: the effort expended on verification should depend on an estimate of its value to the system. This might depend on factors such as the potential consequences of missing an anomaly, or the availability of resources needed for verification.

Acknowledgements

This work was supported in part by the Defense Advanced Research Projects Agency, monitored by the Office of Naval Research under contract N0014-85-K-0108 and by the Air Force Office of Scientific Research under contract F49620-88-C-0058.

References

[Carbonell, 1979] J. Carbonell. *Subjective Understanding: Computer Models of Belief Systems.* PhD thesis, Yale University, 1979. Technical Report 150.

[Charniak, 1977] E. Charniak. Ms. malaprop: A language comprehension program. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence,* Cambridge, Mass., August 1977. IJCAI.

[Cullingford, 1978] R. Cullingford. *Script Application: Computer Understanding of Newspaper Stories.* PhD thesis, Yale University, 1978. Technical Report 116.

[DeJong, 1979] G. DeJong. *Skimming Stories in Real Time: An Experiment in Integrated Understanding.* PhD thesis, Yale University, May 1979. Technical Report 158.

[Kahneman *et al.*, 1982] D. Kahneman, P. Slovic, and A. Tversky. *Judgement under uncertainty: Heuristics and biases.* Cambridge University Press, 1982.

[Kass and Leake, 1988] A. Kass and D. Leake. Case-based reasoning applied to constructing explanations. In J. Kolodner, editor, *Proceedings of the Case-Based Reasoning Workshop,* pages 190–208, Palo Alto, 1988. DARPA, Morgan Kaufmann, Inc.

[Kolodner, 1984] J.L. Kolodner. *Retrieval and Organizational Strategies in Conceptual Memory.* Lawrence Erlbaum Associates, Hillsdale, N.J., 1984.

[Leake, 1988a] D.B. Leake. Evaluating explanations. In *Proceedings of the Seventh National Conference on Artificial Intelligence,* pages 251–255, Minneapolis, MN, August 1988. AAAI, Morgan Kaufman Publishers, Inc.

[Leake, 1988b] D.B. Leake. Using explainer needs to judge operationality. In *Proceedings of the 1988 AAAI Spring Symposium on Explanation-based Learning.* AAAI, 1988.

[Lebowitz, 1980] M. Lebowitz. *Generalization and Memory in an Integrated Understanding System.* PhD thesis, Yale University, October 1980. Technical Report 186.

[Rosch *et al.*, 1976] E. Rosch, C.B. Mervis, W. Gray, D. Johnson, and P. Boyes-Braem. Basic objects in natural categories. *Cognitive Psychology,* 8:382–439, 1976.

[Schank and Abelson, 1977]
R.C. Schank and R. Abelson. *Scripts, Plans, Goals and Understanding.* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1977.

[Schank, 1982] R.C. Schank. *Dynamic Memory: A Theory of Learning in Computers and People.* Cambridge University Press, 1982.

[Wilensky, 1978] R. Wilensky. *Understanding Goal-Based Stories.* PhD thesis, Yale University, 1978. Technical Report 140.