

Six Problems for Story Understanders

Peter Norvig

Division of Computer Science
Department of EECS
University of California, Berkeley
Berkeley, CA 94720

ABSTRACT

Story understanding programs have been classified as script-based processors, goal-based processors or multi-level processors. Each program introduces a new knowledge structure and invents a mechanism to make inferences and manage memory for that knowledge structure. This can lead to a proliferation of incomplete, incompatible processing mechanisms. The alternative presented here is to concentrate on the processing mechanism. It is suggested that a single inferencing scheme can deal with all knowledge structures in a uniform manner. Six basic problems that such a processor must address are presented and discussed.

Introduction

Much work in story understanding has been devoted to describing new knowledge structures such as procedures [Winograd72], scripts [Schank77], plans and goals [Wilensky78], affects [Dyer82], and plot units [Lehnert82]. One is tempted to say that the field of story understanding has "advanced" from script-based processing to goal-based processing to multi-level processing. Each new knowledge structure brings with it a new program embodying a new processing algorithm. Unfortunately, these new knowledge structures are often introduced before the problems of processing the old ones are worked out. This preoccupation with knowledge structures can sometimes lead to programs with impoverished, redundant, or inconsistent processing mechanisms.

For example, Wilensky introduced the "explanation-driven understanding" algorithm for dealing with actions, plans, and goals. He also had a separate algorithm for detecting the "point" of a story. Similarly, Lehnert proposes a system with two processing mechanisms side by side: one for making low-level (sentence based) inferences, and another for high-level (plot based) inferences. In both cases, each of the two processing mechanisms makes many of the same types of memory fetches and inferences. It would seem more economical to have one mechanism serve both jobs. Besides being more economical, a unified processing mechanism would be less *ad hoc*, and would force the system builder to consider more carefully the difficult problems of a complete memory and inference processor.

A program following this unified processing approach has been written, and is undergoing further development. The program is called FAUSTUS (Frame Activated Unified Story Understanding System). As an example of its capabilities, FAUSTUS can take the input story:

Frank hated his job at the factory. He wanted a job where he wouldn't have to work so hard. He envied his friends who went to college, and didn't have to work. So Frank quit building cars and enrolled at the local University. However, as a student, he was soon working harder than he ever had in his life.

and produce the summary:

Frank enrolled in college. He thought being a student would be easy. Ironically, he ended up working harder than ever.

This text makes use of knowledge at various levels of complexity: objects, people, institutions, affects, actions, expectations, and so on. In FAUSTUS each of these knowledge structures is represented as a *frame*, and each is handled by the same basic set of frame manipulation processes.

FAUSTUS itself does not deal with English text. Instead, it calls on the PHRAN parser [Wilensky80] and the PHRED generator [Jacobs83] to translate between English and the internal frame representation. FAUSTUS is described in more detail in [Norvig83].

Six Problems

In this paper I will present six basic problems that must be addressed by any story understander. For each problem a few solutions are discussed, including the solution implemented in FAUSTUS. The emphasis is on solutions that have pervasive effects: that cut across several problems, and several levels of frame complexity.

1. Finding Candidate Frames

According to Rumelhart [Rumelhart75], "the process of understanding a passage consists in finding a schema which will account for it." While I believe there's more to understanding than that, the problem of frame-finding is an important one, and will be the first of six discussed here. Consider Charniak's [Charniak82] example:

As Jack walked down the aisle he picked up a can of tuna fish and put it in his basket.

This work supported by National Science Foundation grant IST-8007045.

The problem is how to find the supermarket-shopping frame, even though it was never explicitly mentioned. The solution implemented in FAUSTUS relies on spreading activation and hierarchical composition. The basic rule is that all parts of all frames can *potentially* be used to find a frame (that is, we are not restricted to a small set of "triggers" for each frame), but in practice the following system is used: (1) if the input matches one unique frame, then instantiate that frame. (2) If a few frames are matched, consider each one and try to make a choice among them. (3) If the input matches a large number of frames, spread "activation energy" to each frame, and check to see if the total energy exceeds a predefined threshold necessary for instantiation.

To put it another way, if the input is unambiguous, interpret it that way. If there are a small number of possible meanings, try to decide among them, and if there are a large number, don't even try to make a choice, unless one of the choices has already been strongly indicated. Currently, "a small number" is defined as four or less. This triage system is not restricted to the problem of finding candidate frames, but is also used in choosing the right frame, relating new input to existing frames, and in going from an abstract to a more concrete frame. It is a general answer to the problem of "choosing (or not choosing) from several possibilities."

The concept of spreading activation is not used to actually determine what frames to use, or to make inferences, but only to suggest possible frames. Decisions are made by a more discrete process.

To get back to the example, suppose "walked down the aisle" matches the supermarket-shopping, department-store-shopping, hardware-store-shopping, church-wedding, lecture-listening, concert-going, ballgame-watching, airplane-trip, bus-trip and train-trip frames. Each would have their activation energy incremented. Similarly, "pick up can of tuna fish" and "put can of tuna fish in basket" would each match several different frames, including the supermarket-shopping frame, and the net result would be that only the supermarket-shopping frame would acquire sufficient activation energy to pass the threshold and become instantiated.

This approach was implemented in FAUSTUS and tested on the text above. Although the approach worked, it is still suspect: it relies on the proper choice of numbers for the activation energy and threshold to make the example come out right. We would like an alternative solution that is not so dependent on the choice of numbers.

One such solution is to arrange the frames hierarchically, to cut down the number of frames matched by any input. In the current implementation, "Walked down the aisle" matches only four frames: store-shopping, church-wedding, public-event-watching and mass-transit-trip. Thus, we can afford to consider each of these frames carefully, rather than just spreading activation energy to them. When we come to "pick up can of tuna fish," we notice it matches the "get item for purchase" piece of the store-shopping frame, and thereby instantiate the frame. The problem of getting from store-shopping to supermarket-shopping is discussed in section 5.

As a final remark, we note that the hierarchical approach can find the store-shopping frame for the following text, while the spreading activation algorithm would not be able to:

As Jack walked down the aisle he picked an object off the shelf.

2. Choosing the Right Frame

It's easy to choose among candidate frames when there are contradictions to rule out all but one candidate. The hard part is weighing the merits of several frames, none of which are obviously incorrect. Consider these two sentences (originally presented by Paul Kay):

- (a) The florist sold a pair of boots to the Balerina.
- (b) The cobbler sold a pair of boots to the Alpinist.

In (a) we want to choose a default prototypical selling frame. However, in (b) we would be remiss if we didn't choose the commercial-transaction frame; we should make the connection that the boots are from the cobbler's shop, that the selling is done there, and that the Alpinist will probably use them in his avocation. This interpretation is preferred because it is richer than the interpretation for (a); it ties together more information. FAUSTUS tries to find interpretations that account for all of the input, but it does not have any more sophisticated decision rules.

A delayed way to make a choice is through *attrition*, the process that drops frames out of active consideration over time. Older frames with lower initial activation are the first to go. Thus, if we are faced with a choice of five candidate frames, and are unable to choose between them, eventually the candidate frames will start dropping out. Finally we will be left with only one frame; the one with the highest initial activation. This is choice by default. There is some evidence that such choices are reinforced over time. Consider the text:

Doctor Smith entered the operating room.
She was the best open-heart surgeon the Medical Center ever had...

When FAUSTUS reads the word "she," the representation for Doctor Smith is altered to record the fact that she is female. This overrides the default, but causes no problems. However, if the text were part of a novel, and twenty pages had elapsed between the time Doctor Smith was introduced and the time the doctor was referred to as "she," why might the reader then be surprised? I submit it is because the candidate frame female-doctor dropped out by attrition, and the male-doctor frame (with its higher *a priori* activation) was chosen by default. At this point it cannot be changed without noticing a contradiction. Paul Kay [Kay81] discusses a similar example.

3. Relating New Input to Existing Contexts

Many inputs should not instantiate new frames, but rather fit in as part of an existing instantiated frame. In [Wilensky78] it was proposed that the understander should first try to interpret the input as elaborating an existing frame, and if that fails to try to find a new frame. In [Wong81] the algorithm is to consider simultaneously any existing frames the input might match and a novel frame created on the spot.

Orthogonal to those two algorithms is the choice of where to look first. One possibility is to look in all currently activated frames for unfilled slots that match the current input. This may be slow if there are many active frames, and if we have to go through them linearly. The alternative is to look first in the generic knowledge data base (as we did to find candidate frames in the first place) and then if we do find a match, check to see if there are any frames of the proper type currently instantiated.

4. Recovering from an Incorrect Choice

One mark of a good story understander is the ability to back up, replacing an incorrect assumption with an alternative when the story merits it. Granger [Granger81] has worked on a story understander (and a question answerer) with this ability. The problem is really three-fold: recognizing that an error has occurred, locating the error, and correcting it. The hard part is locating the error. It would be easy if each frame had a list of concepts that are *not* part of the frame, but obviously we can not afford to store that kind of knowledge. FAUSTUS makes the third task easier by keeping candidate frames around (in a separate data base) for a short time after they are discarded as inappropriate. This makes the problem of selecting a new frame identical to the problem of selecting an old one, provided the candidate frames have not yet been lost through attrition. Consider the following text:

As Jack walked down the aisle he picked up a can of tuna fish and put it in his basket. As a janitor at the stadium, he had to pick up a lot of trash after a big game like this one.

The reader decides on the supermarket-shopping frame after the first sentence. In the second sentence the location of stadium conflicts with the location supermarket. FAUSTUS then backs up, compares the supermarket-shopping scenario with the stadium-janitor scenario, and determines that only the later scenario accounts for all the inputs.

Notice that the ability to back up requires meta-knowledge of what knowledge was acquired through input, through inference, or through default. As discussed above in problem 2, the distinction between default and input probably becomes lost over time.

5. Finding the Proper Level of Abstraction

Consider the example:

John went to the pool at the country club. He sat in a chair.

Most readers would probably view "chair" as "lounge chair." This process of going from the abstract concept chair to the more concrete concept lounge chair is called *concretion*. How might it be done? Again, spreading activation is a possible algorithm, and again, it is not a very attractive one. We probably want "pool" to activate "lounge chair" to only a very low degree.

Another possible algorithm is exhaustive search; on processing "chair" we could consider in turn office-chair, easy-chair, lounge-chair and a host of others. But this approach is also problematic. The fact that John is at the pool suggests lounge chair, but the fact that, say, he is a manager for the Xenon Corporation suggests office-chair, while the fact that he is tired suggests easy-chair, *etc.* We would need a sophisticated comparison mechanism to make a choice among these possibilities. Of course, this algorithm, like all exhaustive search algorithms, is also computationally expensive.

The third algorithm is to do intelligent search; to use the fact that the type of furnishing in a room (or more generally a setting) is usually determined by the type of room. Then we need only notice that the setting is "at the pool" and the prototypical chairs for a pool are lounge chairs.

The fourth algorithm is to use *a priori* probabilities. This is the last resort when no intelligent search heuristics are available.

The last choice is not to do concretion at all. In fact it is a difficult question to decide when to stop concreting. A possible answer is suggested by Rosch's [Rosch78] notion of *basic level*. Her hypothesis is that there are levels in the hierarchy of frames that are normally used to express the frame in question. Thus it would be normal to say "I almost ran over a dog on my way to work," but abnormal to say "I almost ran over a German Shepard" or "I almost ran over a mammal." So perhaps we should stop trying to concretize when we reach a basic level.

In the preliminary implementation of concretion currently used by FAUSTUS, the rule is to *always* try to concretize all frames to the most specific possible frame.

6. Deciding What to Do Next

In problem 5 above, we came up with the tentative solution that FAUSTUS would *always* try to concretize instantiated frames to more specific versions. However, the notion of *always* is not precise enough; we need a measure of how hard (or how often) to try to concretize, in comparison to the other tasks: invoking frames, choosing among invoked frames, relating the input to existing frames, or just reading the next input. In the current implementation of FAUSTUS, this is done with an agenda system that simulates a multi-processor with priority queuing: it repeatedly finds and runs the task with the highest priority, taking aging into account.

One feature of this priority system is that it allows for individual differences in processing. For example, I experimented with raising the priority of the task of reading the next input. The result is a simulation of "skimming" the input text; FAUSTUS makes far fewer frame determinations and inferences of all kinds. Unfortunately, it was not a very good simulation: FAUSTUS's choice of inferences in this mode is rather odd.

Another interesting application for FAUSTUS, not yet implemented, is suggested by Granger's recent work [Granger82] on classifying people's problem solving strategies. He had subjects read texts like the following:

John asked Mary to marry him. She started to cry. Why did she cry?

Subjects would either stick to the hypothesis that Mary would be happy, and answer something like "they were tears of joy," or they would allow the second sentence to supplant the initial hypothesis and answer something like "she decided she couldn't marry John." Interestingly, subjects tended to consistently choose one strategy or the other, over a number of texts (although some subjects used both strategies). This suggests that the tendency towards a particular processing strategy is engrained at a deep level, and is not dependent on the type of knowledge under consideration. FAUSTUS could be fairly easily made to simulate one of these strategies, while it would be more difficult to alter the behaviour of a knowledge-dependent processor in this way.

Acknowledgements

Many of the ideas expressed here were developed jointly with Joe Faletti, Marc Luria, Robert Wilensky, and other members of the Berkley AI Research group. Joe Faletti and I also worked together on a preliminary implementation of the frame manipulation package; this package was shared by FAUSTUS and PANDORA, Faletti's problem solving program.

References

- Charniak82.
Charniak, Eugene, "Context Recognition in Language Comprehension," pp. 435-454 in *Strategies for Natural Language Processing*, ed. Martin H. Ringle, Lawrence Erlbaum Associates, Hillsdale, NJ (1982).
- Dyer82.
Dyer, Michael G., "Affect Processing for Narratives," *Proceedings of the National Conference on Artificial Intelligence*, pp. 265-268 (1982).
- Granger81.
Granger, Richard H., "Directing and Re-Directing Inference Pursuit: Extra-Textual Influences on Text Interpretation," *Proceedings of the Seventh IJCAI*, pp. 354-361 (1981).
- Granger82.
Granger, Richard H. and Jennifer K. Holbrook, "Perseverers, Recencies and Deferrers: New experimental evidence for multiple inference strategies in understanding," *Information and Computer Science*, U.C. Irvine, Irvine, CA (1982).
- Jacobs83.
Jacobs, Paul, "Generation in a Natural Language Interface," *Proceedings of the 8th IJCAI*, (1983).
- Kay81.
Kay, Paul, *Three Properties of the Ideal Reader*, Berkeley Cognitive Science Program (1981).
- Lehnert82.
Lehnert, Wendy G., "Plot Units: A Narrative Summarization Strategy," pp. 375-414 in *Strategies for Natural Language Processing*, ed. Martin H. Ringle, Lawrence Erlbaum Associates, Hillsdale, NJ (1982).
- Norvig83.
Norvig, Peter, "Frame Activated Inferences in a Story Understanding Program," *Proceedings of the 8th IJCAI*, (1983).
- Rosch78.
Rosch, E., "Principles of Categorization," in *Cognition and Categorization*, ed. B.B. Lloyd, Lawrence Erlbaum Associates (1978).
- Rumelhart75.
Rumelhart, David, "Notes on a schema for stories," pp. 211-236 in *Representation and Understanding*, ed. A. Collins, Academic Press, New York (1975).
- Schank77.
Schank, Roger C and Robert P. Abelson, "," in *Scripts, Plans, Goals and Understanding*, Erlbaum, Hillsdale, N.J (1977).
- Wilensky80.
Wilensky, Robert and Yigal Arens, "A Knowledge-based Approach to Natural Language Processing," Memorandum No. UCB/ERL/M80/34, UC Berkeley ERL, Berkeley (1980).
- Wilensky78.
Wilensky, Robert W., *Understanding Goal-based Stories*, Yale University Computer Science Research Report, New Haven, CT. (1978).
- Winograd72.
Winograd, Terry, *Understanding Natural Language*, Academic Press, New York (1972).
- Wong81.
Wong, Douglas, "Language Comprehension in a Problem Solver," *Proceedings of the Seventh IJCAI*, pp. 7-12 (1981).