Reading Instructions *

Richard Alterman and **Tamitha Carpenter**
Computer Science Department and Center for Complex Systems
Brandeis University, Waltham, Massachusetts 02254
tamitha@chaos.cs.brandeis.edu (617) 736-2718

Abstract

This paper describes a model for reading instructions. The basic framework is that an agent engages in an activity and resorts to using instructions only when "all else fails". That is, by reading the instructions *during* the period of engagement, the meaning of the instructions can be clarified by feedback from the world. This model has been implemented in a computer program, IIMP. IIMP is the instruction reading component of FLOABN (Alterman et al., 1991), an integrated architecture whose domain is reasoning about the usage of mechanical and electronic devices.

Introduction

In general, instructions can convey a variety of information to an agent, ranging from short fixes to a given plan (imagine encountering a detour sign on the road), to the specification of an entirely new plan (e.g., instructions for assembling a piece of furniture). Instructions are difficult to understand in the abstract because they are schematic. Most instructions (written or verbal) omit an enormous number of shared details, with the assumption that an agent sharing general background and an understanding of the situation can fill them in (e.g., LeFevre & Dixon, 1986). Outside of the context of use, the agent can grasp only the general sense of what the instructions mean and the operations they depict. Examples of previous computational approaches to instruction usage include Mannes & Kintsch, 1991; Kieras & Bovair, 1986; and Chapman, 1990.

This paper develops a model of instruction usage where instructions are used after the agent engages in the activity. The basic idea is that instructions are used only when "all else fails". This model has been implemented in a computer program called IIMP. IIMP relies on the fact that, by reading the instructions *during* the period of engagement, the meaning of the instructions can be clarified by feedback from the world. IIMP is the instruction reading component of FLOABN[1], an integrated architecture for an intelligent agent. The core of FLOABN is an adaptive planner which interacts in a situation largely by a process of constructing an interpretation of that situation (Alterman, 1988). When IIMP is invoked, it has access to the interpretation of the current state of the situation of engagement that was constructed by the adaptive planner.

IIMP combines marker passing (e.g., Norvig 1989, Charniak 1986, and Hendler 1986) with summarization and importance techniques (e.g., Lehnert, 1981; Kintsch & van Dijk, 1978; Graesser, 1981; Trabasso & van den Broek, 1985) to make the connection between text and situation. This yields two results. First, it builds a representation of the instructions which the planner can proceduralize and then act upon. Second, IIMP learns the 'purpose' of the instructions and adds this to semantic memory to aid future interpretations in similar situations.

Domain and an Example

This paper explores a model of text understanding in the domain of learning from instructions. This model is developed in the context of FLOABN, an integrated system for reasoning about electronic and mechanical devices. The core of the system is an adaptive planner and plan-learning system, SCAVENGER (Zito-Wolf & Alterman, 1990). When all of SCAVENGER's plan-adapting techniques fail, the second component of FLOABN, IIMP, interprets the instructions provided on or near the device. The capability to use this sort of human intervention greatly reduces the possibility for failure and the amount of problem-solving the system must do as it deals with real-world situations. This is augmented by a third component, SPATR (Goodman, Waterman, & Alterman 1991), which uses a case-based approach to reason about the spatial content of the instructions.

IIMP uses marker passing techniques to build a coherence analysis of the relationship between the ongoing activity and the instructions. IIMP then uses a summary of the instructions to learn the main idea they are trying to express. If this idea is not already

*This work was supported in part by the Defense Advanced Research Projects Agency, administered by the U.S. Air Force Office of Scientific Research under contract #F49620-88-C-0058.

[1] For Lack Of A Better Name. See [Alterman *et al.*, 1991] for further details

1. Please hang up,
2. listen for dial tone,
3. deposit the required coin,
4. dial your call again.

Figure 1: The instructions heard on the pay phone.

in semantic memory, IIMP adds it. By doing so, IIMP will be able to use the current instructions to ease the understanding process for future examples.

The centerpiece example in this paper will be a FLOABN simulation of an agent interacting with a pay telephone for the first time. When he approaches the pay phone, the agent has a semi-complete set of procedures for using a home phone and the assumption that the procedures for using the pay phone will be similar. The sequence of steps that occurs is as follows: the agent lifts the receiver, dials a number, and hears two rings followed by a beep. Immediately following the beep, a recording gives the instructions shown in Figure 1. During this sequence of events, SCAVENGER is building a representation of the ongoing activity in semantic memory, which is available to IIMP when SCAVENGER asks for help. Using this representation, the recorded instructions, and the instruction 'For local cash calls deposit ten cents before dialing' from the front of the pay phone, IIMP is able to accomplish two things. First, it builds a representation of the content of the instructions, which SCAVENGER is then able to *proceduralize*[2] and incorporate into its plan. Second, it learns that the main point of the instructions is: "payment on a pay phone involves inserting a coin", which IIMP adds to semantic memory. In subsequent encounters with the pay phone, this concept is used and refined (e.g., refinements include different types of payment on a pay phone).

Three Stages of IIMP: An Overview

IIMP works in three stages. The first stage of IIMP is:

- Use semantic memory to build a coherence representation that characterizes the relationship between the instructions and the ongoing activity.

IIMP constructs a 'copy' of a coherent, internally consistent piece of instantiated semantic memory that represents the coherence of the instructions in relation to the ongoing activity. Each node in the coherence graph (or 'interpretation graph') is either explicitly mentioned in the text of the instruction, a concept encountered during the building of the coherence representation, or a concept developed in the understanding of the ongoing activity as developed by the adaptive planner, SCAVENGER.

The second stage of IIMP is:

- Reason about this graph using summarization technology.

The set of procedures that accomplish this are based on the summarization techniques of the program, SSS (Alterman & Bookman, 1991). IIMP uses these techniques to find the conceptual roots of the interpretation graph built in the first stage, generate a graded list of the relative importance of each node in the graph, and, in stage 3, to produce a base-line summary of the interpretation graph.

The third stage of IIMP is:

- Build a new concept in semantic memory which represents what is learned from the instructions.

This stage uses a modification of Explanation-based Learning (EBL) (e.g., DeJong & Mooney, 1986; Mitchell et al., 1986). Since no *goal concept* is provided with instructions, IIMP uses the information found in stage 2 to find the goal concept of the instructions.

Stage 1: Building a Coherence Graph

Artificial intelligence treats text understanding as a problem of representation; a machine 'understands' a piece of text to the extent to which it can use its representation for such tasks as summarization and question answering. IIMP builds this representation by using the 'marker passing' scheme described in Norvig, 1989. (Other examples of marker passing methods for inferencing on text are described in Charniak 1986 and Hendler 1986.)

IIMP's marker passing technique works from a semantic memory structure, recognizing general classes of inference as follows:

1. For each node **N** in semantic memory representing the ongoing activity and each word in the text of the instructions, do:
   (a) Find the nodes related to **N** via any of a relatively small number of predefined patterns of connectivity (*path shapes*).
   (b) Identify nodes where these paths meet (*collisions*).

Path shapes are defined by the types and order of links between the nodes. Only a few collision types are defined or needed; all undefined collisions are ignored.

Again, consider the pay phone example. This stage begins with a semantic memory containing relevant background knowledge, including knowledge about the current situation of engagement, but with no representation of a pay telephone (since this is the first time IIMP has encountered a pay phone). IIMP passes markers from each significant word in the instructions (i.e., words other than 'please', 'the', and 'and'), as well as from the current plan being adapted: 'telephone plan'. 'Hang up' is found to indicate that the call should be restarted. 'Listen for dial tone' and 'dial' are found to be steps from the original plan.

The instruction 'deposit the required coin' causes a collision at the 'payment' node, as illustrated in Figure 2. The path between 'insert-coin' and 'payment' is labelled as a 'goal of action' path shape. The path between 'telephone plan' and 'payment' is labelled as a 'goal of plan' path shape. The collision defined by these two path shapes is called a 'new step' collision. This indicates that 'insert coin' is valid as a 'new step'

[2] To convert the "understanding" into plan modifications.

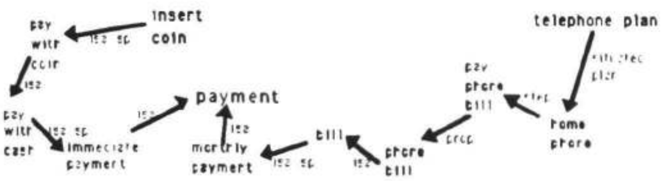

Figure 2: The collision between telephone-plan and insert-coin

for the 'telephone plan', because it satisfies the goal of 'payment', which was satisfied in the 'home phone' situation by paying a monthly bill.

When the marker passing is completed, all the paths that were involved in defined collisions, such as those in Figure 2, define the *episode*: a relevant subset of semantic memory which represents the coherence representation of the text. The episode will be used in the next two stages. This representation has three features that are important for the functionality of the following stages: unconnected features will not be included, since they cause no collisions, thus eliminating irrelevant information; it is a 'copy' of a piece of semantic memory (i.e., the representation is built from the 'vocabulary' and 'structure' of semantic memory); and it is in the form of a directed acyclic graph (DAG).

Stage 2: Analysis of the Coherence Graph

During this stage of processing, IIMP analyzes the coherence representation of the instructions that was produced in stage 2. This is done using summarization techniques described in Alterman & Bookman (forthcoming). These techniques take advantage of the DAG structure imposed on the input as a result of the semantic memory encoding during the coherence building stage.

The first step is to find the **conceptual roots** of the coherence graph; roughly, these correspond to the basic notions of the understanding. The conceptual roots are the minimal set that *covers* the interpretation graph, where coverage is defined in terms of *reachability* in graph theory. Since the interpretation graph is a DAG, the conceptual roots are simply all the nodes unreachable from any other node. The conceptual roots for the pay phone example are PAYMENT, INSERT, RESTART-CALL, and TELEPHONE-PLAN. Using the conceptual roots, IIMP is able to succinctly explain the connection between each instruction and the ongoing activity (see Figure 3). The adaptive planner is then able to proceduralize this connection and incorporate it into a new plan.

Also derived from the coherence graph is a measure of importance that quantifies the conceptual emphasis of the understanding. The importance of any given node is roughly the number of nodes that node covers. For example, the most important node from the pay phone example is 'payment' with an importance of

```
((hang-up :type instruction :isa restart-call
      :modifier ((momentarily)))
 (listen-for-dial-tone :type instruction :step-of telephone-plan)
 (insert-coin :type instruction :isa payment :with
      ((:value * ten-cents)))
 (dial :type instruction :modifier ((again)) :step-of telephone-plan))
```

Figure 3: The results produced for SCAVENGER.

12, the next is 'insert with an importance of 9, and so on. Alterman & Bookman (*ibid*) prove that each of the most important nodes is either a conceptual root or is covered by one of the conceptual roots. In the next stage, the importance measure will be used in conjunction with the conceptual roots to determine the goal concept of the example.

Stage 3: Explanation-based Learning

The instructions provided with a device provide sufficient information, when read in the context of the situation of engagement, to perform the task required. However, the instructions provide more information than simply mindless commands. That is, not only do the instructions say *how* to accomplish a goal, but they say, implicitly, *why* the steps given will accomplish that goal. While understanding the reasons for taking a step is not necessary to perform the step, this understanding makes it possible to learn from this implicit information in a way that can aid in the use of instructions in future situations.

Traditional EBL is unable to cope with the possibility of learning from instructions. This is because EBL 1) begins with a functional description of a goal concept, then 2) builds an explanation of why a training example demonstrates this goal concept, and 3) generalizes this explanation. The problem with directly applying the EBL approach is that, frequently, instructions do not explicitly provided a goal concept. IIMP deviates from EBL by first performing text inferencing to construct an explanation, in the form of a coherence graph. IIMP then determines the goal concept, using the most important conceptual root from this graph. Moreover, the summary of the coherence graph serves as the generalization.

Generalization as summarization

Many techniques have been used for forming the generalization in the EBL literature (*partial evaluation*: van Harmelen & Bundy; *goal regression*: Mitchell et al.; *explanation patterns*: Schank & Leake). The key to generalization is to determine the relevant features of the training example while building the explanation — which is exactly the role of summarization. Based on the results from stage 2, IIMP constructs a summarized version of the basic event content of the instructions and treats it as the generalization. This summary consists of those conceptual roots with above average importance (and, implicitly, the subset of the DAG which these conceptual roots cover). Alterman & Bookman

(1991) empirically show that this summary has four interesting properties:

- **coherence**: The summary holds together and make sense.
- **coverage**: The summary covers, at least implicitly, many of the events of the original text.
- **importance**: The summary includes the important parts of the text.
- **workload**: It takes less work to construct an interpretation of the summary than of the original understanding.

Each of these properties has significance for the generalization stage of EBL. Ensuring that the generalization constructed by EBL provides coherence and coverage means that the generalization will include the important points of the training example and will have internal consistency.

Creating a generalization that decreases the workload for the system is important for ensuring the operationality of the result. Operationality requires two properties: usability and utility. That is, the resulting concept definition must be usable by the system, and the result must improve the system's performance. It is important to explicitly show operationality in traditional EBL systems, so the final result will be strictly distinguishable from the original statement of the goal concept. From the perspective of summarization, the property of utility is seen as equivalent to the property of reduced workload. Thus, the utility of the explanation is assured by the 'simplification' property of the summary.

Below, we will show that the question of importance has additional significance for EBL: the most important concept of the generalization provides valuable information for determining the goal concept when one is not explicitly provided.

Finding the goal concept

The information that IIMP needs to derive the goal concept comes from the analysis of the coherence graph and the summarization. What remains of the task of finding the goal concept is to give it a name and decide where it belongs in semantic memory.

The most important concept is *related* to the goal concept. The goal concept from any text is going to be related to the most important concept. This is assured by the importance measure itself, which was developed to choose the main thrust of a piece of text. For the pay phone example, the relationship between the goal concept and the most important conceptual root is that 'payment on a pay telephone' is an adjustment for the ongoing activity on how 'payment' is achieved.

The most important concept is *not* the goal concept. When IIMP began reasoning about the pay phone instructions, the notion of payment and its connections to the telephone plan and coin insertion were already in semantic memory. Analyzing the coherence

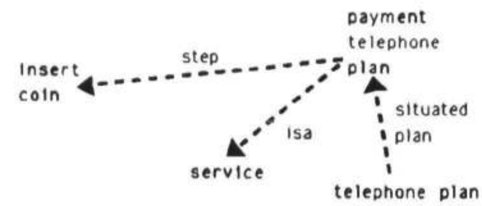

Figure 4: The results of operationalizing and classifying the goal concept

graph identified 'payment' as the most important conceptual root, but IIMP does not need to learn about payment. What IIMP does need to learn about is payment *on a pay telephone*, and how it is related to using a phone at home. If this notion were already in semantic memory, the graph built in stage 1 would have a different character, and the EBL mechanism would act differently. Alternately, if the background knowledge were too specific, where some EBL systems might build an entirely new concept for the new example, this system would relate the new example to the too-specific piece of memory.

It is true in general that the most important concept is not the goal concept. Simply put, IIMP already knows about the concept expressed by the conceptual root, since it was copied directly from IIMP's original memory. The concept IIMP is trying to learn should be a new concept that somehow encapsulates the understanding of the example. In other words, semantic memory has a gap that IIMP needs to fill.

Technique.

IIMP's EBL technique uses the results from the previous stages as follows:

1. Let **N** be the most important concept as determined in stage 2 (e.g., 'payment').
2. Let **C** be the collision occurring at N (this includes the piece of semantic network defined by the collision as well as the collision type, e.g., 'new step')
3. if **C** is a 'new step' collision:
   (a) Let **N1** be the beginning of the 'goal of plan' path shape (e.g., 'telephone plan').
   (b) Let **N2** be the beginning of the 'goal of action' path shape (e.g., 'insert coin').
   (c) Create a new node, **X**, using **N1** and **N** (e.g., 'payment telephone plan'). **X** is a 'situated plan' of **N1**, and **N2** is a 'step' for **X**.
4. if **C** is a 'similar action' collision:

⋮

The new piece of semantic built by this technique is consistent with the rest of semantic memory, and will maintain the integrity of the marker passing in later scenarios.

Classifying the goal concept

The previous step operationalized the goal concept, 'payment on a pay telephone', adding all the relevant information from the instructions to semantic memory. This involved giving it a name and a place in semantic memory. However, the structure of semantic memory provides some implicit information which can be made explicit now. This information could be found when doing text inferencing (stage 1), but making semantic memory more explicit now will make future inferences more efficient.

In particular, IIMP finds the relationship between the goal concept and the nodes covered by the most important concept. By passing markers from the new node in semantic memory, IIMP is able to learn more from how it is related to other concepts in semantic memory. In the case of the pay phone example, the new concept, 'payment telephone plan', has a relationship with the original telephone plan at the 'service' node, so an 'isa' link is added from the new concept to 'service'.

The final result of operationalizing and classifying the goal concept for the pay phone is shown in Figure 4.

Summary and Remarks

The problem of reading an instruction in the context of the ongoing activity is a problem of text inferencing. IIMP combines the advantages of marker passing and summarization to produce a coherent interpretation that is built in relation to plan notions and actions and is given in a form that a planner can use to construct a new plan. This approach is able to provide information that is unavailable when instruction usage is separated from action, and, thereby, increases the domains in which a planner is able to act.

Using an augmented version of explanation-based learning, IIMP is able to build new concepts in semantic memory, thus making the content of the instructions available for use when interpreting future instructions for the device. For example, a subsequent encounter with the pay phone involves payment for a long distance call. Rather than constructing the interpretation of payment from scratch, IIMP simply augments the interpretation built during the first encounter with the pay phone.

References

[Alterman and Bookman, forthcoming] Richard Alterman and Lawrence A. Bookman. Reasoning about a semantic memory encoding of the connectivity of events. *Cognitive Science Journal*, forthcoming.

[Alterman *et al.*, 1991] Richard Alterman, Roland Zito-Wolf, and Tamitha Carpenter. Interaction and instruction usage. *To appear in* The Journal of the Learning Sciences, 1991.

[Alterman, 1988] Richard Alterman. Adaptive planning. *Cognitive Science*, 12:393–421, 1988.

[Chapman, 1990] David Chapman. Vision, instruction, and action. Technical Report AI-TR 1024, MIT Artificial Intelligence Laboratory, 1990.

[Charniak, 1986] Eugene Charniak. A neat theory of marker passing. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 584–588. Morgan Kaufmann, 1986.

[DeJong and Mooney, 1986] Gerald DeJong and Raymond Mooney. Explanation-based learning: An alternative view. *Machine Learning*, 1(2):145–176, 1986.

[Goodman *et al.*, 1991] Marc Goodman, Scott Waterman, and Richard Alterman. Interactive reasoning about spatial concepts. To appear in *Thirteenth Annual Meeting of the Cognitive Science Society*, 1991.

[Graesser, 1981] Arthur Graesser. *Prose comprehension beyond the word*. Springer-Verlag, 1981.

[Hendler, 1986] James Hendler. *Integrating marker-passing and problem-solving: A spreading activation approach to improved choice in planning*. PhD thesis, Brown University, 1986.

[Kieras and Bovair, 1986] David E. Kieras and Susan Bovair. The acquisition of procedures from text: A production-system analysis of transfer of training. *Journal of Memory and Language*, 25:507–524, 1986.

[Kintsch and van Dijk, 1978] Walter Kintsch and Teun van Dijk. Toward a model of text comprehension and production. *Psychological Review*, 85:363–394, 1978.

[LeFevre and Dixon, 1986] Jo-Anne LeFevre and Peter Dixon. Do written instructions need examples? *Cognition and Instruction*, 3(1):1–30, 1986.

[Lehnert, 1981] Wendy Lehnert. Plot units and narrative summarization. *Cognitive Science*, 5:293–331, 1981.

[Mannes and Kintsch, 1991] Suzanne M. Mannes and Walter Kintsch. Routine computing tasks: Planning as understanding. Unpublished manuscript, 1991.

[Mitchell *et al.*, 1986] T. Mitchell, R. Keller, and S. Kedar-Cabelli. Explanation-based generalization: A unifying view. *Machine Learning*, 1:47–80, 1986.

[Norvig, 1989] Peter Norvig. Marker passing as a weak method for text inferencing. *Cognitive Science*, 13(4):569–620, 1989.

[Schank and Leake, 1989] Roger C. Schank and David B. Leake. Creativity and learning in a case-based explainer. *Artificial Intelligence*, 40:353–385, 1989.

[Trabasso and van den Broek, 1985] Tom Trabasso and Paul van den Broek. Causal thinking and representation of narrative events. *Journal of Memory and Language*, 24:612–630, 1985.

[van Harmelan and Bundy, 1988] Frank van Harmelan and Alan Bundy. Explanation-Based Generalisation=Partial Evaluation. *Artificial Intelligence*, 36:401–412, 1988.

[Zito-Wolf and Alterman, 1990] Roland Zito-Wolf and Richard Alterman. Ad-hoc fail-safe plan learning. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, pages 908–914. Lawrence Erlbaum Associates, 1990.