

UC Irvine

ICS Technical Reports

Title

Explanation-based learning of indirect speech act interpretation rules

Permalink

<https://escholarship.org/uc/item/1tv3x5t2>

Authors

Schulenburg, David
Pazzani, Michael J.

Publication Date

1989-05-10

Peer reviewed

Notice: This Material
may be protected
by Copyright Law
(Title 17 U.S.C.)

Z
699
C3
no. 89-11

Explanation-Based Learning of
Indirect Speech Act Interpretation Rules

David Schulenburg (SCHULENB@ICS.UCI.EDU)
Michael J. Pazzani (PAZZANI@ICS.UCI.EDU)

Technical Report - 89 - 11

10 May 1989

Irvine Computational Intelligence Project
Department of Information & Computer Science
University of California, Irvine, CA 92717 USA

copyright © 1989 University of California, Irvine

Abstract

We describe an approach to deriving efficient rules for interpreting the intended meaning of indirect speech acts. We have constructed a system called SALLY that starts with a few, very general principles for understanding the intention of the speaker of an utterance. After inferring the intended meaning of a particular utterance, SALLY creates a specialized rule to understand directly similar utterances in the future.

Introduction: Indirect Speech Acts

Responding appropriately to a question often requires the listener to understand the intention of the speaker. For example, consider the following simple question:

Q: Do you have a match?

Taken literally, this question is a request for information. However, in most contexts, this question should be interpreted as a request for the listener to give the speaker a match. This is a kind of speech act (Austin, 1962) called an indirect speech act (Searle, 1975), in which the intent of the speaker differs from the direct, literal meaning of the speaker's utterance. For a computer to take part in a conversation, it is essential that it have the ability to understand indirect speech acts. An important part of this capability is to gain an understanding of the class of situations in which the indirect interpretation should be preferred to the direct interpretation. For example, a slight variant of the above question is typically interpreted differently:

Q: Do you have a BMW?

Two approaches to the interpretation of indirect speech acts have been proposed in computational linguistics. One approach, typified by QUALM (Lehnert, 1979), makes use of a large number of fairly specific, knowledge-intensive interpretation rules. For example, QUALM contains one rule that interprets a question to verify if the listener possesses an object as a request for the listener to give the speaker the object, if the object is small and inexpensive. The primary advantage of the knowledge-intensive approach is that it is efficient. A discrimination net that indexes the interpretation rules directs the search for an interpretation. There are several disadvantages with the knowledge-intensive approach as implemented in QUALM. First, it is difficult if not impossible, to encode an exhaustive set of rules that would perform well on a large variety of examples. Second, the knowledge-intensive approach does not capture any of the generalities among interpretation rules. A wide variety of knowledge-intensive interpretation rules are specialized forms of a general rule: one interpretation of a question to verify that a precondition of a plan is true is that the speaker wants the listener to execute the plan. Finally, as a cognitive model, the approach does not specify how the interpretation rules might be acquired or extended as new plans are learned. The interpretation of an indirect speech act is a function of the plans that the speaker believes the listener is capable of executing (or understanding) (Perrault and Allen, 1980). When an additional plan is acquired, it may be necessary for the knowledge-intensive approach to acquire additional interpretation rules.

The alternative approach to finding the intended meaning of an indirect speech act is to have a small set of general rules that a listener may use to infer the speaker's plan from the utterance (Allen and Perrault, 1980; Cohen and Perrault, 1979). This approach takes advantage of planning formalisms (Wilensky, 1983; Fikes, 1971) to represent the content of a conversation (Grosz and Sidner, 1986; Litman and Allen, 1987). Unlike knowledge-

intensive rules, these general rules can be applied to a variety of examples since the rules operate on a specification of the speaker's or listener's plans. However, there are also several disadvantages to this approach. First, the search for an interpretation can be inefficient. Second, as a cognitive model, it is not clear that a human listener goes through the long inference process that is necessary to arrive at the interpretation. For example, Searle (1975) has said: "In normal conversation, of course, no one would consciously go through the steps involved in this reasoning." Allen and Perrault (1980) have made similar statements that do not leave open the possibility that people unconsciously go through a long inference chain:

Note that, in actual fact, people probably use much more specialized knowledge to infer the plans of others, thereby bypassing many of the particular inferences we suggest. Our approach so far, has been to specify a minimal set of reasoning tools that can account for the behavior observed.

Psycholinguistic studies have shown that in many circumstances, it takes no longer for a person to recognize an indirect speech act than to find the direct meaning of an utterance. For example, in one experiment (Gibbs, 1983), subjects found it no more difficult to find the indirect interpretation of a request such as "Can't you be friendly?" than the literal interpretation:

The approach that we take is a hybrid between the specific, knowledge-intensive approach and the general, plan-based approach. In particular, we make use of general plan-based rules to interpret novel (to the system) utterances. However, once an interpretation has been found, we derive a knowledge-intensive rule to interpret directly "similar" utterances in the future. The knowledge-intensive rule is created by explanation-based learning techniques (Mitchell, Kedar-Cabelli, and Keller, 1986; DeJong and Mooney, 1986). The "similar" utterances are those that share the features that the plan-based analysis needed to check to infer the interpretation of the indirect speech act.

Explanation-based learning

Explanation-based learning (EBL) is a learning method which analytically generalizes an example. EBL systems share a common approach to generalization. First, an example problem is solved producing an explanation (occasionally called a justification, or a proof) that indicates what information (e.g., features of the example and inference rules) was needed to arrive at a solution. Next, the example is generalized by retaining only those features of the example which were necessary to produce the explanation. This generalization characterizes the class of problems that will have the same solution for the same reason as the training example. EBL explicates (or operationalizes (Keller, 1987)) information that is implicitly represented in a system. For example, ACES (Pazzani, 1987) is a system that learns diagnosis heuristics (i.e., efficient heuristics that associate faults with symptoms) from a functional device description. In this work, we are using a modified version of the EGGS (Mooney and Bennett, 1986) explanation-based learning algorithm to explicate conditions under which an indirect interpretation of a speech act can be inferred.

EBL
def

If the effect of act is e and actor wants e then actor wants act.	[Action-Effect Rule]
If actor1 wants actor2 to want to do act then actor1 wants actor2 to do act	[Want-Action Rule]
A precondition for actor atransing object is that actor possess object.	[give requires have]
actor1 will atrans (cheap) object to actor2 if actor1 and actor2 are friends.	

Figure 1. Speech act interpretation rules¹

Learning to interpret indirect speech acts

We illustrate the process that SALLY goes through to learn a rule to interpret directly indirect speech acts with an example. Consider again the request: "Do you have a match?" The surface speech act here is a verification of possession of a match. However, in most contexts, the intent of the speaker is not to ask for a verification. Rather, the speaker is requesting some action of the hearer, e.g., to give the match to the speaker. The ATRANS-Request Conversion rule (Lehnert, 1978) states that given a verification request of a possession state of some object which has little value, a possible target interpretation is a request of the hearer to give the speaker that object. We address here the issue of learning this rule. We can trace through the understanding cycle used to generate the ATRANS interpretation using the method of Allen and Perrault (1980). SALLY makes use of backward chaining inference rules for inferring the speakers intentions, and for indicating the effects and preconditions of plans. Figure 1 illustrates four of the rules that are used in the following example.²

The initial representation of the surface speech act is:³

HBSW (S-REQUEST (S, H, INFORMIF (H, S, POSSESS (H, MATCH))))

This is read as: "The hearer believes the speaker wants ..." (this is the intentional part of the speaker's speech act) "... to perform a yes-no question regarding the hearer's possession of a match." Using the action-effect (an effect of the S-REQUEST is that HW(HDo action)) and want-action rules, SALLY can infer:

HBSW (INFORMIF (H, S, POSSESS (H, MATCH)))

That is, the hearer believes that the speaker wants the hearer to inform the speaker whether

¹ Action-Effect and Want-Action are from Allen & Perrault (1980).

² The Appendix lists the rules using SALLY's actual representation.

³ In this discussion, we use Allen and Perrault's notation because it is more concise than the equivalent representation used in the computer implementation.

or not the hearer possesses the match. An effect of INFORMIF is KNOWIF; using the action-effect rule again SALLY can infer:

HBSW (KNOWIF (S, POSSESS (H, MATCH)))

From here, SALLY can use the know-positive rule and infer:

HBSW (POSSESS (H, MATCH))

Since possessing a match is a precondition to giving it away, SALLY can use the precondition-action rule to infer:

HBSW (ATRANS (H, S, MATCH))

This inference process interprets the surface speech act of asking about possession of a match as an indirect speech act of requesting the hearer to give a match to the speaker. For a request speech act to have the desired effect, it is necessary that the hearer want to comply with the request. SALLY has a rule stating that someone will give someone else an object if it is of little value and there is an amicable relationship between the two people. Using a small data-base of plan-based rules, SALLY constructs an inference chain of length seven to infer the speaker's intent in this question. Explanation-based learning techniques can be used to "compile" this inference process.

The effect of explanation-based learning on this example is to create a knowledge-intensive rule which avoids many of the intermediate steps of the plan-based inference. The knowledge-intensive rule has the same conclusion as the longer inference process. The preconditions on this rule are exactly those features of the surface speech act and the situation which were tested during the inference process to establish the conclusion. In this case, these preconditions are that the object of the inquiry be of little value, and that the relationship between the speaker and the hearer be an amicable one. Figure 2 illustrates the result of the explanation-based learning process on this example.

Once SALLY has acquired the rule in Figure 2, the interpretation of similar queries is more direct. For example, to interpret the question "Do you have some gum?" requires an inference chain of length two. The constraints that are derived during the explanation-based learning process do not allow utterances such as "Do you have a BMW?" to be interpreted as requests.

Current status

SALLY is implemented in Common Lisp. It does not currently contain a parser. The input to SALLY is a representation of the surface speech act of an utterance; the output is an identification of the intended speech act (e.g., REQUEST, INFORM, etc.). Using a similar line of reasoning to the above example, we have been able to reconstruct some of QUALM's

```

(<- (INTERPRETATION (S-REQUEST (SPEAKER ?S)
                                (HEARER ?H)
                                (ACT (TYPE INFORMIF)
                                      (ACTOR ?h)
                                      (TO ?S)
                                      (STATE (TYPE POSSESS)
                                            (OBJECT (P-OBJ (TYPE ?T)
                                                         (OWNER ?H)
                                                         (LOC ?L)
                                                         (VALUE CHEAP)))
                                            (ACTOR ?H))))))
    (REQUEST (SPEAKER ?H)
             (HEARER ?S)
             (ACT (TYPE ATRANS)
                   (ACTOR ?H)
                   (OBJECT (P-OBJ (TYPE ?T)
                                   (OWNER ?H)
                                   (LOC ?L)
                                   (VALUE CHEAP)))
                   (TO ?S)
                   (FROM ?H))))
    (RELATIONSHIP ?H ?S AMICABLE))

```

Figure 2. A knowledge-intensive indirect speech act interpretation rule.³

knowledge-intensive heuristics from a minimal set of interpretation rules and a library of plans. For example, we can generate the Function-Request Conversion rule. Consider: "Do you have your car here?" The surface speech act is a verification request regarding the location of the hearer's car. An indirect interpretation could be "Would you drive your car?" Recognizing that possession of some object in the immediate vicinity is a precondition to using that object, we can infer the desired indirect interpretation.

One limitation of SALLY is that it is limited to utterances that can be addressed in the plan-based approach to interpreting indirect speech acts. For example, there is no difference in the literal meanings of "Can you pass the salt?" and "Are you able to pass the salt?" However, the indirect interpretation is acceptable of the former but not the latter. Plan-based approaches such as ours ignore the linguistic information associated with the literal meaning of the utterance (Hinkleman, 1988). One solution to this problem, which is more faithful to the psycholinguistic data, is to have rules that map phrases to interpretations without having an intermediate representation of the literal meaning.

Conclusion

We have proposed a hybrid approach to the interpretation of indirect speech acts that combines the best features of knowledge-intensive and plan-based approaches. In particular, the intended meaning of common types of indirect speech acts are found rapidly by knowledge-intensive rules that directly map the surface speech act of an utterance to the intended

⁴ Questions of the form "Do you possess an inexpensive object?" are interpreted as a request for the hearer to give the speaker the object if the speaker and the hearer have an amicable relationship.

meaning. However, it is not necessary to hand-code and maintain a large set of interpretation rules. The knowledge-intensive rules are acquired by using explanation-based learning after the interpretation of a novel utterance is found by a general, but inefficient search process.

Acknowledgements

We would like to thank Ray Mooney for discussions on EGGS and Elizabeth Hinkleman for discussions on interpreting indirect speech acts.

References

- Allen, J. & Perrault, C. (1980). Analyzing intention in utterances. *Artificial Intelligence*, 15, 143-178.
- Austin, J. (1962). *How to do things with words*. Cambridge, MA: Harvard University Press.
- Cohen, P. & Perrault, C. (1979). Elements of a plan-based theory of speech acts. *Cognitive Science*, 3, 177-212.
- DeJong, G. & Mooney, R. (1986). Explanation-based learning: An alternate view. *Machine Learning*, 1, 145-176.
- Fikes, R. & Nilsson, N. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2, 189-208.
- Gibbs, R. (1983). Do people always process the literal meaning of indirect requests? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 3, 524-533.
- Gibbs, R. (1984). Literal meaning and psychological theory. *Cognitive Science*, 8, 275-305.
- Grosz, B. & Sidner, C. (1986). Attention, intentions and the structure of discourse. *American Journal of Computational Linguistics*, 12, 175-204.
- Hinkleman, E. & Allen, J. (1988). *How to do things with words, computationally speaking* (Technical Report). Rochester, NY: Computer Science Department, University of Rochester.
- Keller, R. (1987). Defining operationality for explanation-based learning. *Proceedings of the National Conference on Artificial Intelligence* (482-487). Seattle, WA: Morgan-Kaufmann.
- Lehnert, W. (1978). *The process of question answering*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Litman, D. & Allen, J. (1987). A plan recognition model for subdialogues in conversation. *Cognitive Science*, 11, 163-200.

- Mitchell, T., Kedar-Cabelli, S., & Keller, R. (1986). Explanation-based learning: A unifying view. *Machine Learning*, 1, 47-80.
- Mooney, R. & Bennett, S. (1986). A domain independent explanation-based generalizer. *Proceedings of the Fifth National Conference on Artificial Intelligence* (551-555). Philadelphia: Morgan Kaufmann.
- Pazzani, M. (1987). Explanation-based learning for knowledge-based systems. *International Journal of Man-Machine Studies*, 26, 413-433.
- Perrault, C. & Allen, J. (1980). A plan-based analysis of indirect speech acts. *American Journal of Computational Linguistics*, 6, 167-182.
- Searle, J. R. (1975). *Indirect speech acts*. In P. Cole & J. L. Morgan (Eds.), *Syntax and Semantics, Vol. 3, Speech Acts*. New York: Academic Press.
- Wilensky, R. (1983). *Planning and understanding*. Reading, MA: Addison-Wesley.

Appendix

```

; indirect interpretation
  (<- (interpretation (s-request (speaker ?s)
                                (hearer ?l)
                                ?a1)
        (request (speaker ?l)
                 (hearer ?s)
                 ?a2)))

; "want precondition"
  (want ?s (request (speaker ?s)
                   (hearer ?l)
                   ?a1)
          ?a2)
  (want-perform ?l ?s ?a2))

; "action effect" rule
  (<- (want ?a ?act ?res)
      (effect ?act ?e)
      (want ?a ?e ?res))

; "want-action" rule
  (<- (want ?a (want ?s ?act ?res1) ?res2)
      (want ?a ?act ?res2))

; "know positive" rule
  (<- (want ?a (knowif ?a ?p) ?res)
      (want ?a ?p ?res))

; "precondition-action" rule
  (<- (want ?a ?p ?res)
      (precondition ?p ?res))

; an "amicable" relationship is required for ATRANSing cheap objects
  (<- (want-perform ?actor ?for (act (type atrans)
                                     (actor ?actor)
                                     (object (p-obj (type ?x)
                                                (owner ?actor)
                                                (loc ?loc)
                                                (value cheap)))
                                     (to ?for)
                                     (from ?actor)))
      (relationship ?actor ?for amicable))

```

```
; precondition for ATRANSing is possession
  (<- (precondition (state (type possess)
                          (object ?o)
                          (actor ?a))
      (act (type atrans)
           (actor ?a)
           (object ?o)
           (to ?to)
           (from ?a))))))

; precondition for using is possession
  (<- (precondition (state (type possess)
                          (object ?o)
                          (actor ?l))
      (plan (type use)
            (actor ?a)
            (object ?o))))))

; effect of request is want
  (<- (effect (request (speaker ?a)
                      (hearer ?s)
                      ?act)
      (want ?s ?act ?res)))

; effect of informif is knowif
  (<- (effect (act (type informif)
                  (actor ?a)
                  (to ?s)
                  ?p)
      (knowif ?s ?p)))
```