

Planning and learning in domains providing little feedback

David Pautler

The Institute for the Learning Sciences
1890 Maple Avenue
Evanston, IL 60201
pautler@ils.nwu.edu

Abstract

Letters, and speech acts in general, provide little feedback about their success or failure, which makes it difficult for generators to improve their planning knowledge. Informal observation indicates that humans may learn better writing techniques through discovery rather than failure, however. I present a bottom-up, case-based planner that was developed to overcome some shortcomings of classical planners in *planning* speech acts, and I discuss how this planning technique addresses the problem of *learning* better techniques as well.

Introduction

The focus of my research is LetterGen, a system that generates business letters in four languages: English, Japanese (both Romaji and Kanji), French, and Spanish. The system presents options and follow-up questions to the user about the purpose of the desired letter, and plans a series of speech acts to satisfy the purpose selected. Currently, the domain is limited to the letter-writing needs of a typical research lab. For example, some of the purpose options are: an internal memo announcing a new hire, a letter responding to a speaker invitation, including travel arrangements, and a letter introducing the lab to a potential corporate partner. The language used by a letter is considered part of its purpose.

This paper begins by explaining the two problems in applying classical planning to speech acts that motivated the design of LetterGen. I discuss traditional speech act planners that partially address the two problems and then present the LetterGen architecture. Following that background, I discuss the problem of learning better writing techniques in the face of little feedback and how LetterGen might be further developed to solve the problem, since the learning problem has some relation to the planning problems.

Two planning problems

The generation of speech acts is usually treated as a planning task [Cohen & Perrault, 1979; Appelt, 1985; McKeown, 1985; Hovy, 1988]. And classical planning is the traditional method of planning speech acts. Classical planners require an initial goal state to focus inference, and they restrict inference to the matching of goals with operators. These restrictions make plan formulation much less expensive than undirected inference. But the speech act domain ultimately cannot be kept within these constraints.

For example, all actions, subgoals, and constraints generated by a classical planner exist solely to satisfy the *short-term* initial goal state. There is no direct way to influence the planning process with long-term goals [Schank & Abelson, 1977; McDermott, 1978; Wilensky, 1983; Birnbaum, 1986]. Yet, long-term goals like *Be helpful* and *Be brief* are an important part of conversational speech acts [Grice, 1975]. There are many long-term goals specific to letter-writing as well. For example, if one is enclosing something with a letter, he should mention in the letter what the enclosures are. And if he intends to call at a future date to verify that the package arrived, that should also be mentioned. These actions could be implemented as conditional effects or conditional decompositions for each operator that leads to a letter. But such an extension would do violence to the ideal that operators encode only information relevant to their actions. And the great number of long-term goals that would have to be incorporated this way would make the operators impossibly unwieldy.

The second search strategy of classical planners, limiting inference to the matching of goals with operators, is also a problem in the speech act domain. Operators compile the inferential relations between actions, preconditions, effects, and decompositions into a single data structure. As long as these inferential relations are unique and certain, compilation is not a problem. But the effects of several speech acts, like persuasion, depend on what the speaker

and hearer mutually believe, and modelling mutual beliefs is a recursive process of arbitrary depth [Clark & Marshall, 1981; Ballim & Wilks, 1991] which cannot be compiled in advance. Also, models of mutual belief, both human and computerized, are highly fallible. Due to this fallibility, people often plan several speech acts to satisfy a single persuasion goal. For example, in one of the letters reconstructed by LetterGen¹, the human writer invited a potential sponsor to visit ILS, described the history and purpose of ILS, and gave a phone number. I interpret the second two speech acts as attempts to increase the likelihood that the potential sponsor will respond to the invitation. The all-or-nothing satisfiability of goals in classical planners does not allow for planning multiple actions to increase the likelihood of success.

Related planning work

Previous speech act planners defined operators like:

```
INFORM(speaker, hearer, P)
precond: believe(speaker, P)
decomp: SURFACE-INFORM(speaker, hearer, P)
effect: believe(hearer, P)
```

These operators strengthen the important analogy between normal, physical acts and "speech acts" by making explicit that speech actions also have prerequisites, consequences, and typical methods of implementation. For example, if person A has a goal that person B believe a particular fact, then an appropriate action for A to pursue is INFORMing B of the fact, based on the known effect of INFORMing. But in order to qualify as INFORMing rather than, say, LIEing, A must actually believe what he is about to say. This is reflected in the operator as a precondition². A common way of INFORMing is to express the fact as a declarative sentence, which is what SURFACE-INFORM does as the action decomposition.

In appealing to an analogy, however, one should make explicit what is being smoothed over or ignored in order to make the fit. The inferential relations in physical action operators between preconditions, decompositions, and effects are causal chains. Since all physical objects obey physical causality, we can be as certain about the outcome of plans in the physical world as our confidence in our knowledge of that causality allows. But many of the

¹The LetterGen rule base was developed by reverse-engineering a set of about 30 letters collected from the files of the Institute for the Learning Sciences (ILS) and Andersen Consulting. Thus, some of the letters produced by the system are reconstructions of found examples.

²A filter condition would be more appropriate, since there are no actions A can pursue to negate his own beliefs. And even if A does not have an opinion either way with respect to the truth of what he is saying, people do not appear to create subgoals in order to form such an opinion.

inferential relations in speech act operators are obligatory. For example, believing what we say is merely an obligation. Unlike physical causality, there are two sources of uncertainty: how confident we are that the obligation exists *and* how sure we are that a particular person will obey the obligation. The physical domain only has the first form of uncertainty. Since people can lie, since they can ignore our requests, speech act planning is especially uncertain.

One method of addressing the uncertainty is to make the inferential relations among preconditions and so forth in action descriptions explicit, so the planner can infer how uncertain the outcome is and perhaps plan multiple acts to improve the odds. Notice that the obligations of truth-telling and responding to requests are really long-term goals. They describe the way people prefer to behave, and they can conflict with short-term and other long-term goals. In making the inferential relations explicit, the planner can also notice these conflicts, perhaps deciding that it really does want to lie, for example. LetterGen replaces operators with rule trees to make relations explicit, allowing for the interactions with long-term goals I have just described.

The LetterGen planner

Much of LetterGen's planning knowledge is expressed in rules like:

```
((_interested-in_ ?hearer ?subject)
IMPLY
(_cause_
(_mention-to_that_ ?speaker ?hearer
(_has-background-in_ ?person ?subject) )
(_be-likely
(_desire_ ?hearer (_meet-with_ ?hearer ?person))) ) ) )3
```

that are activated by rules like:

```
(_cause_
(( _desire_ ?person ?outcome) AND
(_believe_ ?person
(_cause_ ?event (_be-likely ?outcome) ) )
(_desire_ ?person ?event) ) )4
```

These two rules describe how an action, mentioning another person's background, is selected on the basis of its likely effect, getting the hearer to want to meet the person.

³This means: "If the hearer is interested in a given subject, then telling him that someone has a background in that subject will cause the hearer to want to meet him." Underbars (_) indicate where arguments should be placed around the predicate, in order, so the proposition can be read as a sentence, a notational convenience.

⁴This means: "If someone desires a certain outcome, and he believes some event will make that outcome likely, then he wants the event to occur."

I am writing to introduce you to Northwestern University's Institute for the Learning Sciences. || As you may know, || the Institute was formed in September of 1989 under the direction of Roger Schank. Our organization establishes common ground between university research and real-world problems in the area of education and corporate training through the building of innovative interactive computer software.

We have a unique approach that allows Institute corporate sponsors both to improve their corporate training and to support efforts to improve our school systems. In addition to our founding sponsor, Andersen Consulting, we are supported by two Institute Partners, Ameritech and North West Water, a water utility company in the United Kingdom. Additional funding comes from IBM, Encyclopaedia Britannica Educational Corporation, Northwestern University, and the Advanced Research Projects Agency (ARPA).

Through our educational efforts, we have established relationships with local school districts, both in the city of Chicago and the neighboring suburbs. It is through the integration of our educational software into these school systems that we hope to effect a positive change in the way children learn. ||

To help further introduce the Institute, I have enclosed an information packet reviewing its history, educational theories, and direction. || Further, please allow me to extend an invitation to you and your colleagues to visit the Institute. We have received some information on <company/proj> and would like to share with you some of the projects currently under development at the Institute. ||

In the meantime, if you have any questions or need additional information, please feel free to contact me at (708) 491-3710 or my assistant, Elizabeth Brown, at (708) 491-3640. || I will contact your offices the week of <date> to determine your receipt of this package and to discuss a possible visit at your earliest convenience.

A "cold call" letter from ILS's files reconstructed by LetterGen

(The vertical bars separate the different text snippets planned by the system.)

Figure 1

The effect is conditional on the hearer's interest in some subject the person has a background in. Once the action is selected, another rule directs the agent to want to satisfy any preconditions. And yet another rule will cause the agent to search for a way of implementing the mention through a specific utterance. Altogether, the tree of inferences formed from these rules performs the same function as an operator. This inferential framework is similar to "rational" planning [Cohen & Levesque, 1990; Sadek, 1994].

Other LetterGen rules encode long-term goals, for example:

```
(_oblige_
  (_with-tool_ (_give_to_ Sender ?item Addressee)
    Postal-service )
  (_mention-to_that_
    Sender
    Addressee
    (_with-tool_ (_give_to_ Sender ?item Addressee)
      Postal-service )))5
```

and ways of incorporating those goals into the planning process:

```
((?situation1 AND
  ((_oblige_ ?situation1 ?situation2) AND
  (_has-agent_ ?situation2 ?agent) ))
  IMPLY (_desire_ ?agent ?situation2) )6
```

Unlike the previous set of rules, and classical planners, these rules are triggered primarily by what the planner knows about the state of the world, not the goal state (although some do modify goals). Since many of the system's goals will be triggered by the world state through these rules, the planner cannot use its initial goal state to focus search; planning will be bottom-up. Rules triggered by the uncertainty of an action's effectiveness and which select subsidiary actions are not covered here, but that is also a bottom-up process.

An advantage operators have over the first set of rules is the memory management provided by the data structure. As discussed so far, the rule tree must be reconstructed at runtime each time it is used, even if it is used frequently. LetterGen's rule base was reverse-engineered from a set of letters written by human beings. A casual observation of how these people wrote the letters revealed that they would

⁵This means: "Sending something indirectly through the Postal Service obliges you to mention to the intended recipient that you're sending it." The bold text indicates constants that are set to new gensym's each time a new letter is started. These constants are typically used by many rules.

⁶This means: "If a situation exists that obliges someone to do an action, that person will want to do that action."

often look for a letter that nearly met their goals and adapted it. In order to recapture some of the memory management lost when I gave up operators, and to mimic the case-based planning behavior of the original letter writers, I store rule trees with MOPs [Schank, 1982] and justification structures [Doyle, 1979]. LetterGen reuses and adapts these cases by following stored rule trees whenever possible during inference [Pautler, 1993, 1994]. When the stored path conflicts with the current set of bindings, the system does ordinary rule retrieval. Retrieved rules may be part of a tree elsewhere in the case base or may be independent; the system pursues rules in other trees before independent rules. New paths are stored for reuse. This framework is similar to the top-down reuse of operator trees done by Derivational Analogy [Carbonell, 1986; Veloso & Carbonell, 1993]. The system currently generates about 50 different letter types, like that in Figure 1.

There are several advantages to operators that have not yet been reproduced using stored rule trees. Perhaps the most important feature is the delete list. The system could assert the negation of any delete list and use a JTMS to resolve the resulting contradiction (the justification structures are already present), but this has not been implemented. There is also presently no method of protecting intervals, largely due to the fact that there was no evidence of goal threat in the letter corpus used, nor is the problem mentioned in the speech act planning literature I researched.

It is difficult to evaluate the quality of plans produced by LetterGen. A planner open to influence from long-term goals must be held to a higher standard than merely satisfying the user's stated goals, since long-term goals can be interpreted as plan quality criteria the user simply did not bother to make explicit. The system must determine on its own which of these criteria are relevant at a given time and *then* satisfy them. Part of evaluation must be how successful the planner is at recognizing these criteria. LetterGen finds all of them, because it runs to quiescence. There are approximately 500 rules in the rule base, but 300 of these are syntactic or used to produce language-specific utterances. Running to quiescence on the remaining 200 is not overly expensive, but adding more planning knowledge could make it so, easily. Search heuristics are needed. The preference of the planner to follow stored plans could be part of such a heuristic. Since stored cases represent plans that have been judged to have successfully found the right long-term goals⁷, case structure will lead the planner to previously relevant long-term goals first. If resource constraints are placed on search, perhaps we can judge the efficiency of the planner in finding appropriate long-term goals.

⁷In the current case, judged so by the human programmer.

A learning problem

An important obstacle to evaluation remains, however. How are we to judge the effectiveness of a letter in persuading someone, for example? In conversation, participants offer immediate indications of their mental state by assenting or presenting a counter argument. The lack of immediacy in letter-writing means the writer must do a more thorough job of anticipating reader responses.

But even if the reader does what we ask, the lag time between our appeal and his response allows for a number of other factors to become probable motivators. The usual method of learning in case-based planners is from plan failure [Hammond, 1986]. But there seem to be too many factors, and the connections too tenuous, to rely on feedback to improve the planner's knowledge. People do not appear to be especially good at diagnosing letter results, either. By introspection, I believe that improvements in letter style and presentation are more discovery-based than failure-based. More specifically, it is a two step process in which new methods are developed by experimenting with one set of rules, and those methods are evaluated using another set of rules. For example, we might experiment with the ordering of making a compliment and asking for a favor. Using the second set of rules, we could place ourselves in the place of the reader and judge his reaction to each ordering.

A consequence of this explanation, perhaps a troubling one, is that this account of learning is indistinguishable from what the planner already does on its own. That is, any result the learner could achieve could also be achieved by the planner, given the same rule base. As mentioned earlier, planning to persuade someone requires that we infer some relevant beliefs and mutual beliefs, like political leanings, for example. That belief modelling done during planning is the same as what would be done during the second learning stage. Perhaps the only way of distinguishing the two is by external circumstance, like whether the reasoning was done for a particular task, and whether the results and path to these results were stored.

The observation is troubling because learning and planning are often thought of as separate modules, manipulating the same set of knowledge, but working independently. However, the letter domain suggests that the planner and learner can be the same process -- insight into how to express something more clearly or to be more persuasive is simultaneously a learning result and a planning result. If there were feedback, learning would certainly be distinct from planning, but there is very little feedback.

One case where we do seem to learn better writing techniques from failure and success is when we read letters written by others. Perhaps the most important contribution of speech act research is the insight that understanding

involves a reconstruction of the speaker's intentions. For example, to correctly respond to an indirect request like, "Do you know the time?", we must realize that the speaker wants the time, not a yes-or-no answer. In reconstructing the plans of a letter-writer, we can examine how we might have tried to meet the same goals and compare our reactions to each. If the real letter does a better job, we can adopt the plan we have reconstructed for it into our own case base. But this learning strategy requires incorporating an entire interpretation component into the planner and occasional time away from planning for the system to digest a corpus of letters to compare with its own experiences.

Perhaps the crucial learning problem in a bottom-up, case-based inference system is knowing which rule trees are worth saving. I have already mentioned that the cases currently in LetterGen were chosen because I judged them to be successful plans. On its own, the system stores all the rule trees it infers. Since case structure is used as a search heuristic, the system should be more selective about what is stored. Certainly, all trees leading from goals to actions that satisfy them, and all trees leading to long-term goals, should be saved. An argument for storing other trees, like the regresses involved in inferring mutual belief from a range of evidence, can also be made. Defining an answer to this question is the primary goal of future research.

Conclusion

The emphasis of the past two years of work on LetterGen has been generating a range of useful output (letters fostering corporate and community interest in ILS, arranging conference travel, memos announcing new hires, etc.) Reverse-engineering a set of planning knowledge from the initial letter corpus revealed that generating letters requires more than operators. In order to generate all the appropriate goals, long-term goals were incorporated and the system made bottom-up. Storing inferential relations leading to long-term goals and then to relevant actions as cases is the primary heuristic for guiding the bottom-up search. But the reliance on the case library for coherent plans begs the question: The current library was built by the programmer; how could the system develop its own coherent cases? The lack of direct contact between the writers and readers of letters makes learning from failure more difficult than for other domains. Learning by discovery seems more plausible. The problem for future research is determining which of these discoveries is worth storing as a useful case.

References

- Appelt, D. (1985). *Planning English sentences*. New York: Cambridge University Press.
- Ballim, A. & Y. Wilks. (1991). *Artificial Believers*. Lawrence Erlbaum Associates.

- Birnbaum, L. (1986). *Integrated processing in planning and understanding*. Ph.D. dissertation, Yale University.
- Carbonell, J. G. (1986). Derivational analogy: A theory of reconstructive problem-solving and expertise acquisition. In R. Michalski, J. Carbonell, and T. Mitchell (eds.), *Machine Learning*. (Volume 2). Morgan Kaufman Publishers.
- Clark, H. H. & Marshall, C. R. (1981). Definite reference and mutual knowledge. In A. K. Joshi, B. Webber, and I. A. Sag (eds.) *Elements of discourse understanding*. Cambridge, UK: Cambridge University Press.
- Cohen, P.R. & H. Levesque. (1990). Rational Interaction as the Basis for Communication. In P.R. Cohen, J. Morgan, and M.E. Pollack (eds.), *Intentions in Communication*. Cambridge, MA: MIT Press.
- Cohen, P.R. & C.R. Perrault (1979). Elements of a Plan-based Theory of Speech Acts. *Cognitive Science* 3:177-212.
- Doyle, J. (1979). A truth maintenance system. *Artificial Intelligence* 12, 231 - 272.
- Grice, H.P. (1975). Logic and conversation. In P. Cole & J.L. Morgan (eds.), *Semantics: an Interdisciplinary reader in Philosophy, Linguistics, and Psychology*. Cambridge, UK: Cambridge University Press.
- Hammond, K. (1986). *Case-based planning: An integrated theory of planning, learning, and memory*. Ph.D. dissertation, Yale University.
- McDermott, D. (1978). Planning and acting. *Cognitive Science* 2, 71 - 109.
- McKeown, K. (1985). *Text generation*. New York: Cambridge University Press.
- Pautler, D. (1993). LetterGen: Writing from examples. Proceedings of the Fifteenth Cognitive Science Society. Lawrence Erlbaum Associates.
- Pautler, D. (1994). *Obligation and Experience in Speech Act Planning*. Proceedings of AAAI-94 Workshop on Planning for Interagent Communication.
- Sadek, M.D. (1994). *Communication theory = Rationality principles + Communicative act models*. Proceedings of AAAI-94 Workshop on Planning for Interagent Communication.
- Schank, R. & R. Abelson. (1977). *Scripts, plans, goals, and understanding*. Lawrence Erlbaum Associates.
- Schank, R. (1982). *Dynamic memory: A theory of reminding and learning in computers and people*. Cambridge University Press.
- Veloso, M. & J.G. Carbonell. (1993). Derivational Analogy in PRODIGY: Automating case acquisition, storage, and utilization. *Machine Learning* 10. Kluwer Academic Publishers.
- Wilensky, R. (1983). *Planning and Understanding*. Reading, MA: Addison-Wesley.