Plan, Activity, and Intent Recognition: Papers from the AAAI 2013 Workshop

Using Plan Recognition for Interpreting Referring Expressions*

Dustin A. Smith and **Henry Lieberman**
MIT Media Lab
20 Ames St.
Cambridge, MA 02139

Abstract

Referring expressions such as *"a long meeting"* and *"a restaurant near my brother's"* depend on information from the context to be accurately resolved. Interpreting these expressions requires pragmatic inferences that go beyond what the speaker *said* to what she *meant*; and to do this one must consider the speaker's decisions with respect to her initial belief state and the alternative linguistic options she may have had. Modeling reference generation as a planning problem, where actions correspond to words that change a belief state, suggests that interpretation can also be viewed as recognizing belief-state plans that contain implicit actions. In this paper, we describe how planners can be adapted and used to interpret uncertain referring expressions.

Recognizing and Synthesizing Phrase-Level Plans

Although plan recognition has a long connection with natural language processing (NLP), historically researchers have focused on two high-level problems: (1) inferring a speaker's communicational and task goals from a speech act (Allen and Perrault 1980; Litman and Allen 1984; Hußmann and Genzmann 1990) and (2) identifying a plan embedded in the content of text (Bruce 1977; Charniak and Goldman 1993; Raghavan and Mooney 2011). In this paper, we examine plan recognition at a finer grained level of linguistic analysis, where observed actions correspond to individual words[1] and the plans correspond to referring expressions.

Researchers in the NLP sub-field of natural language generation (NLG) have deployed AI *planning* techniques to the problem of sentence planning (Koller and Stone 2007; Bauer 2009; Koller, Gargett, and Garoufi 2010; Garoufi and Koller 2010). These planning approaches treat the *lexicon* as the *domain theory* from automated planning: each action (word) is annotated with a description of its meaning (implementing a theory of *lexical semantics*) including the language's serialization constraints, expressed by a *lexicalized grammar* theory. The planner's objective is to choose a sequence of words that adhere to syntactic, semantic and pragmatic constraints in order to achieve the communicational goals while minimizing costs.

In this paper, we describe the problems of generating and interpreting referring expressions, and formulate them as planning and plan-recognition problems. Next, we review two cost-difference approaches to plan recognition and investigate where they can contribute to interpreting referring expressions.

What is plan recognition?

Imagine you see a robot at position s_2, equidistant from two goal states: g_1 and g_2 (Figure 1). Without additional knowledge, you must concede that both goals are equally likely.

However, if you are also given (or estimate) the robot's **initial state**, s_0, and some **observed actions** $\mathcal{O} = \langle a_1, a_2 \rangle$, then you can reason about the possible decisions it made to get from s_0 to s_2. This may give you more information: that the robot's intention was more likely g_2, which you may have inferred by reasoning along the lines "*if it had wanted to go to g_1, it would have taken a left sooner than at s_1.*"

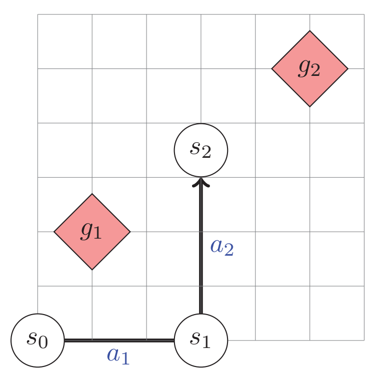


Figure 1: A robot's path from s_0 to s_2 along $\langle a_1, a_2 \rangle$.

Now, suppose a stranger approaches you on the street and hands you a picture of two men, Ronald Reagan and Marvin

*We are grateful for the support of the sponsors of the MIT Media Lab.

[1] There is no good reason to require *words* to be the primitive actions. A more general theory would include all *lexical units*: morphemes to words to idiomatic phrases. Here, we describe our actions as "words" for lack of a better lexical unit.

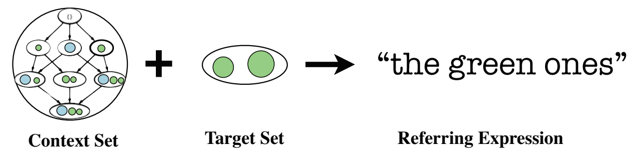
Minsky, and asks you "*Who is that guy*?" Let the ambiguous referring expression be $\mathcal{O} = \langle that, guy \rangle$, and suppose it resolves equally to two disjoint goal states $g_1 = Minsky$ and $g_2 = Reagan$. Depending on your estimations of this stranger's background knowledge[2] you may decide to rule out one candidate after considering the possible linguistic choices in light of a presumed initial belief state: e.g., "*everybody knows who Marvin Minsky is, therefore the stranger probably intended 'that guy' to mean Ronald Reagan.*"

In both cases, inferring the goal involves **abductive reasoning** (Charniak and Goldman 1993). Abduction (informally: "guessing") is an inference to the best explanation: given that B was observed, along with possible explanations $A_1 \rightarrow B$ and $A_2 \rightarrow B$, an abductive process selects the best A_i that accounts for B, where "best" is usually a quantification of the trade-off between benefits and costs. Abduction has been used to address problems in NLU that require making assumptions, such as resolving ambiguity, metonymy and unpacking compound nominals (Hobbs et al. 1993) and reasoning about conversational implicatures (Benotti 2010). In the context of planning, if the observed sentence plan is incomplete (i.e., due to ellipsis, conversational implicature, or presuppositions) or the actions are only partially observed (i.e., due to vagueness or ambiguity), then plan recognition can be used to augment the speaker's observed plan with the listener's assumptions, by abductively inferring implicit actions.[3] By putting language generation and interpretation into the framework of planning and plan-recognition, the concrete computational ideas from planning may shed light on challenging NLP problems.

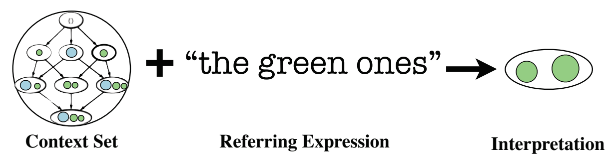
Two Tasks: Generating and Interpreting Referring Expressions

Of the innumerable functions of communication, **referring** is perhaps the most common and has been studied extensively across disciplines; its aim is to convey the identity of an object, agent, event or collection thereof to an audience.

Depending on the participant's role in an exchange of linguistic reference, he will complete one of two tasks: The speaker completes a **reference generation task**: given the possible interpretations, called the *context set* (Stalnaker 2004), and a designated member of the context set containing the intended referent(s), called the *target set*, his goal is to produce a *referring expression*, a noun phrase, that allows the listener to isolate the target set from the remainder of elements in the context set, called the *distractors*:



Given the speaker's referring expression and an assumed context set, the listener is faced with the **reference interpretation task**: infer the targets that the speaker intended to communicate:



For the generation task, the desired semantic content is fixed and the linguistic choices are open; while for interpretation, the linguistic contents are relatively fixed and the semantic possibilities are open.

Running example: The CIRCLES referential domain Throughout this paper, we will use the following three circles to illustrate examples of reference tasks:

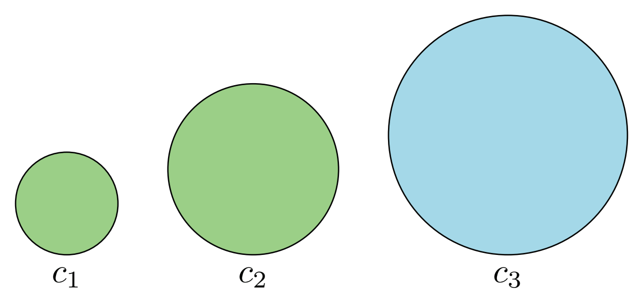


Figure 2: The CIRCLES referential domain containing three referents: c_1, c_2 and c_3.

Given the R referents in Figure 2, a valid interpretation is any non-empty grouping of these referents: $\{c_1\}, \{c_2\}, \{c_3\}, \{c_1, c_2\}, \{c_1, c_3\}, \{c_2, c_3\}$ and $\{c_1, c_2, c_3\}$. An invalid interpretation is represented by the empty set. For both generation and interpretation tasks, any one of these sets can be a target that the speaker intends to encode using a referring expression. However, because of problems like vagueness and ambiguity, referring expressions can lead to *interpretive uncertainty*: when a single referring expression evokes more than one different interpretation. With uncertainty, any one of these $2^R - 1 = 7$ valid sets can either be a known target (in all interpretations), a known distractor (excluded from all interpretations), or *unknown* (in some but not all interpretations). Therefore *uncertain* referring expressions can theoretically have up to $3^{2^{|R|}}$ distinct outcomes!

[2](Clark and Bangerter 2004) conducted a similar experiment and were able to reliably change the listener's interpretation depending on how the question was framed. In planning terminology, this question caused the listener to infer that the speaker began his language plan from a different initial belief state.

[3]Which inferences are justified is often up to fiat. Imagine you are next to a box of chocolate candies, with one large and one small candy remaining. Your friend, who you know is on a diet, asks you to pass him *"a candy."* His use of the indefinite article, *"a"*, suggests that he is giving you permission to disambiguate. However, if he used the definite article *"the"* you would have to decide whether you want to cooperate with his short-term or longer-term goal, or whether you should ask him for clarification.

The result of interpreting a referring expression is a *denotation*, represented with the $[\![\cdot]\!]$ symbol, and is comprised of the set of elements it refers to, given a context set. For example, $[\![$*"the green ones"*$]\!] = \{c_1, c_2\}$.

Generating Reference as Planning

The NLG community has given the reference generation task much attention (see (Krahmer and van Deemter 2012) for a good overview). The original formulation of the problem, often attributed to (Dale and Reiter 1995), restricts the task to **content determination**—selecting the information a referring expression should contain, while deferring the natural language generation to subsequent processing. The output of the content determination is a description that distinguishes the target set from the distractors. For example, if the target set is $\{c_1, c_2\}$, then a valid output from a content determination algorithm is this description, usually in either *attribute-value matrix* or *logical form*:

$$s = \begin{bmatrix} \textsc{type} & \text{circle} \\ \textsc{color} & \text{green} \end{bmatrix} \qquad \begin{aligned} s = \textsc{type}(\texttt{x0,circle}) \\ \wedge\ \textsc{color}(\texttt{x0,green}) \end{aligned}$$

The content s would then be handed to the next step in an NLG pipeline (c.f. (Mellish et al. 2004)) with the ultimate goal of becoming a referring expression (e.g. *"the green circles," "the two green circles," "some green circles"*), embedded in a larger construct, like a sentence. Two components of the so-called "sentence-planning" pipeline include **lexical choice**, choosing the words that express the preselected meaning, and **surface realization**, organizing the syntactic form of the sentence.

A strict pipeline limits the information that is shared between the processes in undesirable ways (Stone and Webber 1998; Krahmer and Theune 2002). As (Horacek 2004) noted, the precise representation of the content may depend on what expressive resources are available to the surface and lexical choice modules. For example, suppose you are trying to generate a referring expression to identify one out of two men, and the target has a full head of hair but the distractor is bald. Instead of the content selection algorithm producing the logical formula, $\textsc{has\_hair}(\texttt{x1})$, it may be preferable to instead use the logically equivalent formula $\neg\ \textsc{bald}(\texttt{x1})$ because it has a simpler surface form and can be expressed as *"not bald,"* whereas in English there is no succinct modifier for $\textsc{has\_hair}(\texttt{x1})$.

Beginning with the SPUD system (Stone et al. 2003), the modular, pipelined approach to the miscellaneous sentence-planning tasks was abandoned for a "lexicalized approach" that defines each word's syntactic, semantic, and (conventional) pragmatic contributions together in its *lexical entry*, collectively stored in a *lexicon*. Each word asserts information that constrains which targets are possible or which elements in the context are distractors.

The exact representational details of these lexical entries leave plenty of design decisions open; however, this approach requires a lexicalized, incremental theory of grammar to specify how the meanings of individual words (actions) interact. In SPUD and its derivatives, syntactic constraints are expressed using a lexicalized version of the tree-adjoining grammar (LTAG) theory, in which larger trees are assembled from smaller elementary trees using only two compositional operations: substitution and adjunction.

(Koller and Stone 2007) observed that this approach could be represented as a classical Strips planning problem and provided examples in a system called CRISP. The planner's *domain theory* is an implementation of a *lexicon* whose entries describe belief-dynamics of the speaker and listeners' information states, rather than changing a description of the state of the world. CRISP defines word-actions in PDDL, whereby each action has a precondition that requires the appropriate substitution or adjunction site for the word's elementary tree and effects that describe its semantic content and syntactic constraints.[4] The goal-test function verifies that the state's syntax is a proper LTAG tree (a single parse tree without any unbound substitution nodes) and that the desired semantic content is asserted. (Garoufi and Koller 2010; Koller, Gargett, and Garoufi 2010) showed that the planning approach of CRISP could also incorporate world context into generation: in addition to manipulating the discourse context, words can be designed to constrain extra-linguistic context by using preconditions that hinge upon the state of a simulated world. The idea is to interpret referring expressions such as "*[take] the second door to your left*," which, assuming the door is out of view, requires the listener to update his non-linguistic context by performing the physical action of moving to the left. This captures some of the so-called presuppositions and conventional implicatures, whereby a word's meaning constrains the previous context set and requires the listener to reason backwards to align her context set with the speaker's.

Despite the apparent advantages of a planning approach, (Koller and Petrick 2011; Koller and Hoffmann 2010) initially reported difficulty designing a domain that could feasibly be searched by a leading heuristic-search planner, FF (**?**). Later, they overcame some of the inefficiencies by modifying FF so that it removed tautologies from the planning domain during preprocessing and restricted actions to those that were beneficial in solving a relaxed version of the planning problem. This leads us to suspect that alternative formulations of planning domains will be able to overcome these performance problems.

Interpretation and Generation with AIGRE We have developed a fast belief-state planner, AIGRE, that can generate and interpret simple English referring expressions. The current implementation handles noun phrases with determiners, cardinals, ordinals, crisp and gradable adjectives (base, comparative and superlative forms), and nouns (singular and plural). AIGRE's approach is to plan over *belief-states*, which represent complete interpretations—implicitly representing all possible targets. Because of the asymmetry of the two tasks, interpretation is a much simpler

[4]It should be noted that the SPUD and CRISP systems do not perform *content selection*–the content to be expressed is given input in logical form.

search problem (over belief-changing word-producing actions) than generation. For interpretation, AIGRE uses a complete, optimal A* search toward valid interpretations. For generation, it uses a stochastic heuristic search toward a communication goal (see (Smith and Lieberman 2013) for more details). Because states are complete interpretations, the planner is incremental and a denotation can be output at any stage. For example, here is its word-by-word interpretation of *"the green ones:"*

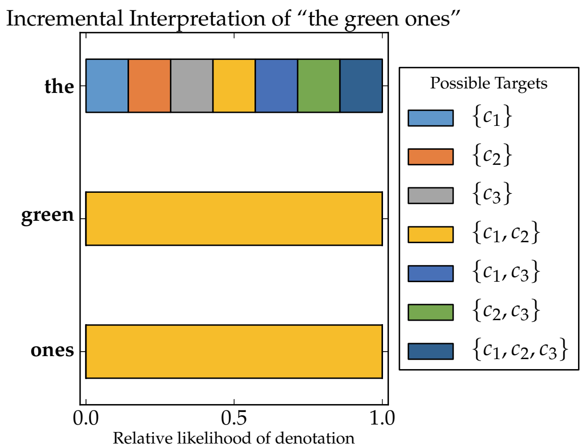


Figure 3: All denotations' relative likelihoods during incremental interpretation of *"the green ones,"* with respect to the CIRCLES domain. More than one color in a bar indicates the interpretation at that point is *uncertain*.

Reference Interpretation as Plan Recognition

In this section, we will review two approaches to plan recognition that are based on classical planning. Second, we will show some idiosyncrasies of the interpretation problem. Last, we will compare the two approaches with a baseline.

Plan recognition is often viewed as an inversion of planning (Carberry 1990); however, until recently, these problems were rarely[5] addressed together. Perhaps due to advancements in automated planning and Moore's law, recently *hybrid approaches* have been developed for decision-theoretic (Baker, Tenenbaum, and Saxe 2007) and classical heuristic-search planners (Ramírez and Geffner 2009; 2010). These hybrid approaches allow the same resources that were used for generation to also be used for interpretation. Here, we restrict our attention to the classical planning approach because of its better scalability.

[5]An exception is the unique approach proposed by (Wilensky 1981) that suggests using meta-knowledge about "how to plan" to characterize the planning process, thereby enabling the problems of planning and plan recognition to be described declaratively as meta-goals. One merit of this approach is its ability to recognize meta-goals; for example, from the story of "[being married and] going out with one's secretary," it would hypothesize a 'Resolve-Goal-Conflict' meta-goal.

Cost-based approaches to plan recognition

Ramírez and Geffner presented two cost-difference approaches to hybrid planning and plan recognition, which we call **R&G-1** (Ramírez and Geffner 2009) and **R&G-2** (Ramírez and Geffner 2010). Both approaches work by generating plans that contain the observations as a subsequence for each possible goal, and then comparing the plans' costs. Like our example in Figure 1, these approaches were founded upon the assumption that the observed actions were performed by a rational agent and were chosen to most efficiently achieve goals. The only difference between the two approaches is how they calculate the costs: **R&G-1** estimates the likelihood of candidate goals by comparing the difference between a plan that contains the observations and an optimal plan that possibly includes observed actions, whereas **R&G-2** compares costs of plans to achieve the goal with and without the observed actions.

We follow the authors by describing the approach in terms of a Strips-based planning domain; however, these approaches are more general. Beginning with a description of actions, $\mathcal{A}$, they receive as input (1) a set of **goal states**, $\mathcal{G}$, (2) a sequence of **observed actions**, $\mathcal{O} = \langle a_0, a_1 \ldots a_n \rangle$ where all of the observed actions are in $\mathcal{A}$, and (3) an initial state s_0 preceding the first observed action.

The actions in $\mathcal{A}$ that produce observations in $\mathcal{O}$ are transformed so that they create a new fluent (a time-indexed propositional variable) that indicates that the action has been observed. For example, if $\mathcal{O} = \langle \texttt{left}, \texttt{right} \rangle$ then two fluents will be introduced, $p_{\texttt{left}}$ and $p_{\texttt{right}}$. The first observation in the sequence is transformed differently than the rest of the observed actions: in this example, `right` is modified to introduce a positive effect (in Strips, this means a literal is added to the *add list*):

- the fluent $p_{\texttt{left}}$ is added to `left`'s effects.

The rest of the actions in the observation sequence are augmented with a *conditional effect* so that the fluent is only added when all previous observations in the sequence have been observed:

- the conditional effect[6], when $p_{\texttt{left}}$ then $p_{\texttt{right}}$, is added to `right`'s effects.

For the goal transformations, the set of candidate goals, $\mathcal{G}$, is redefined into two new goal sets: $\mathcal{G}\mathcal{O}$ and $\mathcal{G}\bar{\mathcal{O}}$. For $\mathcal{G}\mathcal{O}$, each original goal $s_G \in \mathcal{G}$ is extended to also require that all observations have been observed:

$$s_{G\mathcal{O}} = s_G \wedge \underbrace{p_{\texttt{right}}}_{\text{fluent of last action in } \mathcal{O}} \quad (1)$$

Similarly, for constructing our negated goal set, $\mathcal{G}\bar{\mathcal{O}}$, we introduce a modified version of the original goal where the

[6]Conditional effects are a convenient sugaring of the Strips formalism that can be rewritten equivalently as two actions: one action with the condition moved into its precondition and the conditional effect a normal effect, and a second action with the negation of the condition moved into the precondition that does not introduce the conditional effect (Gazen and Knoblock 1997).

observation sequence is not a subsequence of the resulting plan:

$$s_{G\bar{\mathcal{O}}} = s_G \wedge \underbrace{\bar{p}_{\texttt{right}}}_{\text{negated fluent of last action in } \mathcal{O}} \tag{2}$$

(We use the notation $\pi(s_0, s_G) = \langle a_0, a_1 \ldots a_j \rangle$ to denote a plan that transforms the initial state, s_0, into the goal state, s_G.) Each plan has a cumulative cost, denoted $c(\pi(\cdot, \cdot)) = \sum c(a_i)$, equal to the sum of its actions' costs.

R&G-1 proceeds by computing two optimal plans' costs for each goal: (1) $c(\pi(s_0, s_{G\mathcal{O}}))$ with its goal $s_{G\mathcal{O}}$ from the set of transformed goals $\mathcal{G}\mathcal{O}$ and (2) $c(\pi(s_0, s_G)$ the cost of the plan achieving s_G regardless of $\mathcal{O}$. The **cost difference** for a given goal s_G and observations $\mathcal{O}$ is computed by subtracting the costs:

$$\Delta(s_G, \mathcal{O}) = |c(\pi(s_0, s_{G\mathcal{O}})) - c(\pi(s_0, s_G))| \tag{3}$$

For example if $\Delta(s_G, \mathcal{O}) = 0$, it means that to achieve s_G, the agent did not incur any additional costs in making the observed actions; whereas if $\Delta(s_G, \mathcal{O}) = 3$, then the observations are suboptimal for s_G. This metric doesn't take into account the possibility that there are other optimal plans that also achieve the goal *without* including the observation sequence. In other words, this metric fails to account for whether the observations were *necessary* for achieving the plan. **R&G-2** adds this feature by instead comparing the costs of (1) $c(\pi(s_0, s_{G\mathcal{O}}))$ with its goal $s_{G\mathcal{O}}$ from the set of transformed goals $\mathcal{G}\mathcal{O}$ and (2) $c(\pi(s_0, s_{G\bar{\mathcal{O}}}))$ for the corresponding goal $s_{G\bar{\mathcal{O}}}$ in the transformed goals $\mathcal{G}\bar{\mathcal{O}}$:

$$\Delta(s_G, \mathcal{O}) = |c(\pi(s_0, s_{G\mathcal{O}})) - c(\pi(s_0, s_{G\bar{\mathcal{O}}}))| \tag{4}$$

The cost functions for both approaches are inverted and multiplied by a *goal prior*, $P(s_G)$, (in our experiments, the prior is uninformative) to get an equation proportional to the desired goal posterior:

$$P(s_G|\mathcal{O}) \propto \Delta(s_G, \mathcal{O})^{-1} P(s_G) \tag{5}$$

And from 5, to recognize the goal, we search for the goal(s) with the maximal posterior probability:

$$\arg\max_{s_G \in \mathcal{G}} P(s_G|\mathcal{O}) \tag{6}$$

Adapting cost-difference approaches to NLU

Although we have been emphasizing the similarities between language interpretation and plan recognition, this problem domain has several differences:

Observed words frequently map to multiple actions
Ambiguities arise when a single word has multiple meanings, for instance when a surface lexical unit corresponds to multiple actions, or when components can be combined in multiple ways, for example when multiple tree merging operations are possible. Lexical ambiguity can be seen as *partially observed* actions, whereas the syntactic decisions can be viewed as *non-determinism*, where a given action has multiple effects.

Observed sentences (plans) rarely omit words (actions)
In both original formulations of **R&G-1** and **R&G-2** the plan recognizer is allowed to hypothesize any sequence of actions as long as it contains the observations as a subsequence when computing $s_{G\mathcal{O}}$. In natural language, words are only omitted in very special cases, such as when a word is *elided*. For example, the expression *"the biggest"* contains an omitted noun *"the biggest [one]"*; however the listener is not be licensed to supply *any* noun she choses, because convention constrains what values the defaults can take on.

Observed sentences (plans) are communicated intentionally
Linguistic actions are being conveyed by the speaker to his audience during the communication act, consequently the *keyhole methods* for recognizing plans of an uncooperative agent do not appear to apply.

A simple baseline: MinCost

Given these differences, plan recognition approaches can make an even stronger assumption when interpreting natural language: that possible actions are restricted to those whose lexical units produce the text in the remaining observation sequence. In AIGRE, we implemented this by filtering actions according to (1) a language model, and (2) the remaining observation sequence: implicit actions (i.e. assumptions, and elided words) are only introduced when no action's surface text matches the next observation. This search is extremely efficient, and the average interpretation time is $\approx 100ms$ to generate and test the entire search space. Goals states are defined as those that have accounted for all the observations. When there are multiple alternative interpretations, then the result of **MinCost** is the goal state(s) whose plan has the minimal cost:

$$\arg\min_{s_G \in \mathcal{G}} = c(\pi(s_0, s_{G\mathcal{O}})) \tag{7}$$

Comparing the three approaches

In this section we compare the three approaches: a baseline **MinCost**, and the two cost-difference approaches: **R&G-1** (Ramírez and Geffner 2009) and **R&G-2** (Ramírez and Geffner 2010). As input, these are given (1) a sequence of n partial observations of a plan: $\mathcal{O} = \langle a_0, a_1 \ldots a_n \rangle$, (2) a set of possible **goal states**, $\mathcal{G}$, and (3) an initial state s_0 preceding the first observed action. As output, they produce a ranking over the possible goals, which can be used to derive the single most likely goal given the observations.

- **MinCost** Derives the plans that contain all observations for each goal, and then selects the goal(s) with the minimal cost.
- **R&G-1** (Ramírez and Geffner 2009) selects the goal(s) with plans whose cost (i.e., the outcome of **MinCost**) minimally deviates from the cost of the optimal plan for the goal.
- **R&G-2** (Ramírez and Geffner 2010) selects the goal(s) with plans whose cost minimally deviates from the cost of an optimal plan for the goal that does not include the observed plan.

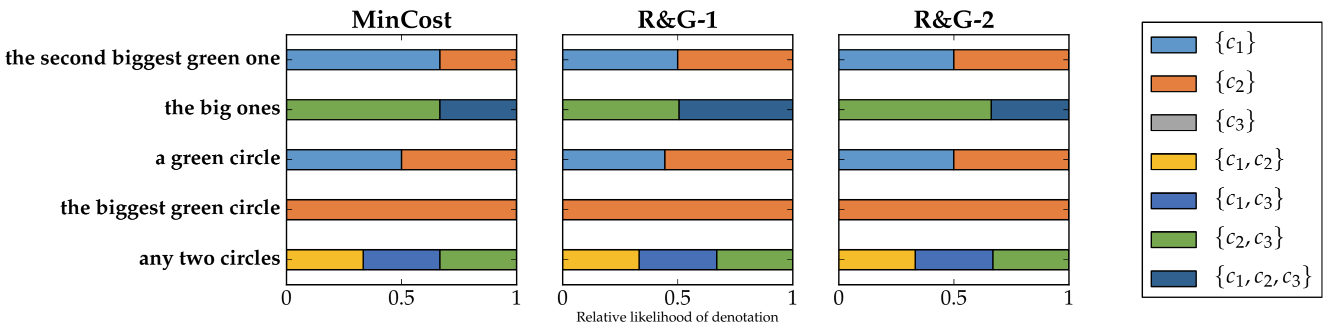


Figure 4: The results of AIGRE's interpretations using **MinCost**, **R&G-1**, and **R&G-2**. For each approach, the bars represent the net relative likelihoods of the denotations for 5 referring expressions in the CIRCLES domain.

The baseline, **MinCost**, ranks the alternative interpretations by simply comparing the likelihood of the plans. The cost-difference approaches look only at the goals $\mathcal{G}$ that are output by **MinCost**, *generate* alternative optimal plans for each of them, and then rerank them by the cost differences. In terms of language, the cost-difference approaches compare each possible way to interpret what the speaker *said* and compare that against the alternative ways the speaker *could have said it* to infer the intended meaning.

When are cost-difference approaches useful? The cost-difference approaches are useful for arbitrating between multiple hypothetical interpretations: a situation that arrises when interpreting text that is uncertain. They are not needed for understanding text for which a single interpretation can be clearly ascertained. Referring to the results in Figure 4, the problematic referring expressions are those that produce multiple outcomes, represented by bars containing multiple colors (all but *"the biggest green circle"*). These referring expressions are problematic for the following reasons:

- The referring expression *"the second biggest green one"* has a syntactic ambiguity: which modifier, 'green' or 'second biggest,' is applied to the noun first? Although (to the authors) 'green' seems to clearly come before 'second biggest', when we surveyed 108 people over Mechanical Turk, about 1 in 3 selected the alternative reading.
- The gradable adjective 'big' in *"the big ones"* suffers from *vagueness* and leads to borderline cases. When gradable adjectives take the positive form and modify a plural noun, they can be open to multiple interpretations and depend upon an implicit choice of comparison class and standard of reference. In this example context set, there are two alternative interpretations: (1) only the two biggest are BIG, or (2) *all* of the circles are BIG.
- The indefinite article in *"a green circle"* leaves the exact outcome intentionally undecided. It conveys that the interpreter is allowed to pick one of the green circles from a set that contains more than one.
- With *"any two circles,"* *any* conveys a choice in a similar way as the indefinite article.

How do we avoid planning for $2^{|R|}$ goals? The goal-test function for the **MinCost** search process selects any plan that accounts for all observations, and the action proposal function is restricted to actions that can produce the observation sequence. These hard constraints allow us to explore the entire search space, and enumerate all of the interpretations that are consistent with the observations. Consequently, we can restrict our hypothesis space to these goals. The two cost-difference approaches also involve generation tasks, which are much more computationally expensive than interpretation.

Analyzing the outcomes of the three approaches How do the cost-difference approaches to plan recognition influence the overall ranking of alternative interpretations? Let's review the results of Figure 4:

- **MinCost** The resulting preference ordering of interpretations:
  - **the second biggest green one** $\{c_2\} \prec \{c_1\}$
  - **the big ones** $\{c_2, c_2, c_3\} \prec \{c_1, c_3\}$
  - **a green circle** $\{c_1\} \sim \{c_2\}$
  - **the biggest green circle** $\{c_2\}$
  - **any two circles** $\{c_1, c_2\} \sim \{c_1, c_3\} \sim \{c_2, c_3\}$
- **R&G-1** The likelihoods changed for all but *"the biggest green circle"*, however the orderings were the same as **MinCost**, except:
  - **the second biggest green one** $\{c_1\} \prec \{c_2\}$
  - **a green circle** $\{c_1\} \prec \{c_2\}$
  - **any two circles** $\{c_2, c_3\} \prec \{c_1, c_2\} \prec \{c_1, c_3\}$
- **R&G-2** Although the likelihoods were different from **R&G-1**, it produced the same orderings.

These three approaches ranked interpretations as a function of the aggregate costs of a plan's actions. In all experiments reported here, AIGRE's word costs were derived from their inverse token frequencies in the Open American National Corpus (Ide and Macleod 2001). Clearly word frequencies are only an approximation and do not precisely quantify the costs of human linguistic decisions. Consequently, the specific outcomes of AIGRE is less important than the details of the general approach for reasoning about alternative linguistic decisions.

TARGETS	MEAN TIME (SEC)	AIGRE'S REFERRING EXPRESSIONS
$\{c_1\}$	7.03 ± 4.9	the left one (1.9), the small one (1.9)
$\{c_2\}$	2.37 ± 1.3	the medium-sized one (1.9), **the right small one** (2.9)
$\{c_3\}$	9.14 ± 7.8	the blue one (1.9), the rightmost one (1.9)
$\{c_1, c_2\}$	0.50 ± 0.0	the green ones (2.0), **the left circles** (2.0), the 2 green circles (3.0)
$\{c_1, c_3\}$	13.93 ± 9.6	the 2 not center ones (3.9), the 2 not center circles (3.9), the 2 not medium-sized ones (3.9)
$\{c_2, c_3\}$	0.51 ± 0.1	**the right circles** (2.0), **the big ones** (2.0), **the two right circles** (2.3)
$\{c_1, c_2, c_3\}$	0.17 ± 0.1	the ones (1.0), the circles (1.0)

Table 1: Example of AIGRE's generative output for all 7 valid targets in CIRCLES. The **bold** referring expressions have been automatically flagged by the AIGRE's interpreter as problematic, because they can be interpreted in multiple ways.

Discussion of the results The cost-difference approaches **R&G-1** and **R&G-2** changed the relative likelihoods for all problematic referring expressions (they didn't change *"the biggest green circle"* because there was nothing to change). Despite this, they both changed the ordering and the most likely interpretation(s) in only 3 of the 4 problematic cases.

For *"the second biggest green one"* (3.99), **MinCost** had a small bias in favor of c_1, because of AIGRE minor syntactic preference for evaluating subsective adjectives in a LIFO rather than FIFO order. **R&G-1** compared, against its optimal alternatives, *"the small one"* (cost: 1.9397) for c_1 and *"the right small one"* (2.9144) for c_2; and **R&G-2** produced *"the small shape"* (1.9916) for c_1 and *"the right small shape"* (2.9664) for c_2. This was *marginally* enough to tip the scale in favor of c_2 for both approaches.

For *"a green circle"* (2.59), **MinCost** put both interpretations c_1 and c_2 into an equivalence class. Yet again, the high cost of the optimal plan for referring to c_2 resolved the ambiguity in favor of c_2 by a extremely small margin for **R&G-1**. The margin for **R&G-1** was much smaller than in the previous expression, because *"a green circle"*'s difference in cost to the optimal *"the small one"* was not nearly as large as it was with *"the second biggest green one"*'s distance to *"the small one:"* it had a smaller cost differential. We want to stress that it is most important to look at the relative magnitude of the change, because the costs could be set or scaled differently to make these minor fluctuations become significant changes.

Lastly, we note that the very minor change among the three options for *"any two circles"* (although imperceptible in Figure 4) was created by the following optimal referring expressions, which were the same for both **R&G-1** and **R&G-2**: $\{c_1, c_2\}$ *"the small ones"* (1.9896); $\{c_2, c_3\}$ *"the right ones"* (1.9717); and $\{c_1, c_3\}$, which had the smallest cost differential with *"the 2 not center ones"* (3.38783).

In conclusion, this demonstrated is that the approaches of **R&G-1** and **R&G-2** succeed in reranking the interpretations based on the *alternative, optimal* ways of referring to them. What this means for the language understanding domain is that if an ambiguous utterance U yields two mutually exclusive interpretations, x and y, then **R&G-1** and **R&G-2** will pick the interpretation whose optimal description is closest in cost to the cost of U. At the moment, there is not enough evidence to suggest one of the two cost-difference methods is superior to the other.

Conclusion

In this paper, we described a promising direction for dealing with the uncertainty created by context-dependent referring expressions, based on reasoning about alternative linguistic decisions. For the problem of generation, planning allows the costs of speech acts to be pitted against achievement of communication goals. For the problem of interpretation, plan recognition offers an abductive framework for balancing the costs of potentially ad-hoc interpretation decisions against the benefits—loosely defined as "a successful interpretation." Additionally, hybrid approaches give a unified framework in which components can be shared between generation and interpretation tasks, and lead to a mutually cooperative division of labor between the two tasks. The research described in this paper explored the possibility of using generation to improve interpretation, via cost-difference approaches to plan recognition. Similarly, we can take advantage of interpretation to detect when the outcome of a generation task is problematic, as we show in Table 1.

The primary drawback of the cost-difference approaches is that they are highly sensitive to the costs that are assigned to each action. The resulting behavior depends on the plan's cost, the contents of the lexicon, and the referring task at hand. Further, corpus estimates of costs like those used in AIGRE omit the syntactic and cognitive costs of computing a word's meaning, *inter alia*. Corpus-estimated costs are a crude approximation for quantifying true human linguistic behavior; learning costs that accurately reflect the performance of a population of human language users makes a challenging follow-up research project. Still, we believe the general idea behind this approach will be useful for dialogue systems that need to perform pragmatic inferences.

References

Allen, J., and Perrault, C. R. 1980. Analyzing intention in utterances. *Artificial Intelligence*.

Baker, C.; Tenenbaum, J. B.; and Saxe, R. 2007. Goal inference as inverse planning. *Proceedings of the 29th annual meeting of the Cognitive Science Society*.

Bauer, D. 2009. Statistical natural language generation as planning. Master's thesis, Department of Computational Linguistics, Saarland University, Saarbrücken, Germany.

Benotti, L. 2010. Implicature as an Interactive Process. *Ph.D. Thesis*.

Bruce, B. C. 1977. Plans and social actions.

Carberry, S. 1990. Plan Recognition in Natural Language Dialog.

Charniak, E., and Goldman, R. P. 1993. A Bayesian model of plan recognition. *Artificial Intelligence*.

Clark, H. H., and Bangerter, A. 2004. Changing ideas about reference. *Experimental pragmatics*.

Dale, R., and Reiter, E. 1995. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*.

Garoufi, K., and Koller, A. 2010. Automated planning for situated natural language generation. *ACL '10: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*.

Gazen, B., and Knoblock, C. 1997. Combining the expressivity of UCPOP with the efficiency of Graphplan - Springer. *Recent Advances in AI Planning*.

Hobbs, J.; Stickel, M.; Appelt, D.; and Martin, P. 1993. Interpretation as abduction. *Artificial Intelligence*.

Horacek, H. 2004. *Lecture Notes in Computer Science*. Springer Berlin Heidelberg.

Hußmann, M. J., and Genzmann, H. 1990. On trying to do things with words: another plan-based approach to speech act interpretation. In *COLING '90: Proceedings of the 13th conference on Computational linguistics*.

Ide, N., and Macleod, C. 2001. The american national corpus: A standardized resource of american english. In *Proceedings of Corpus Linguistics 2001*, volume 3.

Koller, A., and Hoffmann, J. 2010. Waking up a sleeping rabbit: On natural-language sentence generation with FF. In *Proceedings of AAAI 2010*.

Koller, A., and Petrick, R. P. A. 2011. Experiences with planning for natural language generation. *Computational Intelligence*.

Koller, A., and Stone, M. 2007. Sentence generation as a planning problem. *Annual Meeting of the Association of Computational Linguistics*.

Koller, A.; Gargett, A.; and Garoufi, K. 2010. A scalable model of planning perlocutionary acts. In *Proceedings of the 14th Workshop on the Semantics nd Pragmatics of Dialogue*.

Krahmer, E., and Theune, M. 2002. Efficient generation of descriptions in context. *Proceedings of the ESSLLI workshop on the generation of nominals*.

Krahmer, E., and van Deemter, K. 2012. Computational Generation of Referring Expressions: A Survey. *Computational Linguistics*.

Litman, D. J., and Allen, J. F. 1984. A plan recognition model for clarification subdialogues. In *ACL '84: Proceedings of the 10th International Conference on Computational Linguistics and 22nd annual meeting on Association for Computational Linguistics*.

Mellish, C.; Reape, M.; Scott, D.; Cahill, L.; Evans, R.; and Paiva, D. 2004. A Reference Architecture for Generation Systems. *Natural Language Engineering*.

Raghavan, S., and Mooney, R. J. 2011. Abductive Plan Recognition by Extending Bayesian Logic Programs. In *Machine Learning and Knowledge Discovery in Databases*.

Ramírez, M. J., and Geffner, H. 2009. Plan Recognition as Planning. *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence*.

Ramírez, M. J., and Geffner, H. 2010. Probabilistic plan recognition using off-the-shelf classical planners.

Smith, D. A., and Lieberman, H. 2013. Generating and interpreting referring expressions as belief state planning and plan recognition. *Under review*.

Stalnaker, R. C. 2004. Assertion Revisited: On the Interpretation of Two-Dimensional Modal Semantics. *Philosophical Studies*.

Stone, M., and Webber, B. 1998. Textual Economy through Close Coupling of Syntax and Semantics. *arXiv.org*.

Stone, M.; Doran, C.; Webber, B. L.; Bleam, T.; and Palmer, M. 2003. Microplanning with communicative intentions: The spud system. *Computational Intelligence* 19(4):311–381.

Wilensky, R. 1981. Meta-planning: Representing and using knowledge about planning in problem solving and natural language understanding. *Cognitive Science*.