

Generating and Interpreting Referring Expressions as Belief State Planning and Plan Recognition *

Dustin A. Smith
MIT Media Lab
E15-358; 20 Ames St.
Cambridge, MA USA
dustin@media.mit.edu

Henry Lieberman
MIT Media Lab
E15-320F; 20 Ames St.
Cambridge, MA USA
lieber@media.mit.edu

Abstract

Planning-based approaches to reference provide a uniform treatment of linguistic decisions, from content selection to lexical choice. In this paper, we show how the issues of lexical ambiguity, vagueness, unspecific descriptions, ellipsis, and the interaction of subsecutive modifiers can be expressed using a belief-state planner modified to support context-dependent actions. Because the number of distinct denotations it searches grows doubly-exponentially with the size of the referential domain, we present representational and search strategies that make generation and interpretation tractable.

1 Introduction

Planning-based approaches¹ to reference are appealing because they package a broad range of linguistic decisions into actions that can be used for both generation and interpretation. In section 2, we present linguistic issues and discuss their implications for designing planning domains and search algorithms. In section 3, we describe AIGRE,² our belief space planner, and explain how it efficiently handles the issues from section 2. Lastly, we demonstrate AIGRE’s output for a suite of generation and interpretation tasks, and walk through a trace of an interpretation task.

1.1 The two linguistic reference tasks

A linguistic act of **referring** aims to communicate the identity of an object, agent, event or collection thereof to an audience. Depending on the

agent’s dialogue role, referring involves one of two tasks. The speaker completes a **referring expression generation (REG)** task: given a *context set* and a designated member of it called the *target set*, he produces a *referring expression* that allows the listener to isolate the target from the rest of the elements in the context set, called the *distractors* (Dale and Reiter, 1995). A listener completes a **referring expression interpretation (REI)** task: given a referring expression and an assumed context set, her goal is to infer the targets that the speaker intended.

1.2 Reference generation as planning

Many approaches to REG (see (Krahmer and van Deemter, 2012) for an overview) have focused exclusively on the sub-task of **content determination**: given context and target sets, they search for content that distinguishes the targets from the distractors. This content is then passed to the next module in an NLG pipeline (c.f. (Reiter, 1994)) to ultimately become a noun phrase embedded in a larger construct.

These “pipeline” architectures prevent information from being shared between different layers of linguistic analysis, contrary to evidence that the layers interact (Altmann and Steedman, 1988; Danlos and Namer, 1988; Stone and Weber, 1998; Krahmer and Theune, 2002; Horacek, 2004). As an alternative, one can take an integrated “lexicalized approach,” following (Stone et al., 2003; Koller and Stone, 2007; Garoufi and Koller, 2010; Koller et al., 2010), in which each lexical unit’s syntactic, semantic, and pragmatic contributions are represented as a *lexical entry*.

Lexicalized approaches presume that lexical entries can be designed to contain all of the syntactic and semantic ingredients required to synthesize a phrase or sentence. As such, the REG problem is reduced to choosing (i.e., content selection *and* lexical choice) and serializing lexical

We thank Nicolas Bravo and Yin Fu Chen for their contributions to AIGRE; the three anonymous reviewers for their comments; and the sponsors of the MIT Media lab.

¹Throughout this paper planning is framed as a heuristic search problem.

²Automatic interpretation and generation of referring expressions. In French, it means “sour”.

units (putting them into a flat sequence), which bears strong similarities to **automated planning** (Ghallab et al., 2004). Automated planners try to find *plans* (sequences of actions), given (1) a fixed *planning domain* that describes how the relevant aspects of the world are changed by actions, and (2) a *problem instance*: a description of the initial state and the desired goal states.

For planning-based approaches to reference, the set of actions defined by the planning domain is analogous to a *lexicon*: each action corresponds to a lexical unit and is responsible for defining its semantic effects, along with the local syntactic and compositional constraints that are relevant to the lexical unit (Appelt, 1985; Heeman and Hirst, 1995; Koller and Stone, 2007; Koller et al., 2010; Garoufi and Koller, 2011).

1.3 Automated planning as heuristic search

When solving an instance of a planning problem, planners internally generate a directed graph called a **planning graph**, where the nodes represent hypothetical states and the labeled edges correspond to actions that represent valid transitions between the states. A planning domain and an initial state thus characterize an *implicit* graph of all the possible states and transitions between them, which is usually infeasible to enumerate. To avoid constructing parts of the planning graph that are irrelevant to particular problem, planning tasks are often solved using heuristic search (Bonet and Geffner, 2001), which is the same framework underlying popular approaches to content selection (Bohnet and Dale, 2005).³ Heuristic search is useful for balancing costs (e.g. the cost of a given word) against benefits (e.g. meeting the communication goals): lower-cost⁴ solutions are inherently preferred. The effectiveness of heuristic search is determined by the search algorithm and **heuristic function**, which gives a numerical estimation of a given state’s distance to a goal state, $h(\mathbf{s}) \rightarrow [0, 1]$, that guides the search algorithm toward states that have a lower estimated distance to a goal.

The automated planning community has developed domain-independent techniques for automatically deriving a heuristic function from the struc-

³FULL BREVITY ALGORITHM is simply breadth-first search; GREEDY ALGORITHM is best-first search; and the INCREMENTAL ALGORITHM is a best-first where actions are sorted by preferences (Bohnet and Dale, 2005)

⁴If a plan’s cost is just its length, heuristic search will bake-in the brevity sub-maxim of Grice’s Cooperative Principle (Dale and Reiter, 1995).

ture of a planning domain, provided it is encoded a certain way. These approaches solve a simplified version of the original planning problem, calculate each generated state’s minimal distance to a goal, and then use that distance as a lower-bound estimate in the heuristic function for the original problem (Bonet et al., 1997; Hoffmann, 2001).

(Koller and Petrick, 2011; Koller and Hoffmann, 2010) applied domain-independent planners toward REG, but found them “too slow to be useful in real NLG applications.” It is important to note, however, that their results were for a specific implementation of a planning domain and set of heuristic search techniques, of which there are many variations (Edelkamp and Schroedl, 2011). For example, (Koller and Hoffmann, 2010) later reported being able to speed a planner by making its action proposal function more restrictive.

1.4 Interpretation as plan recognition

If generating a sentence can be modeled as a planning problem, then interpretation can be modeled as **plan recognition** (Heeman and Hirst, 1995; Geib and Steedman, 2006). Plan recognition can be seen as an “inversion” the planning problem, and solved using planning techniques (Baker et al., 2007; Ramírez and Geffner, 2010): Given an initial state (context set), a sequence of partially observed actions (words), what are the most likely goals (interpretations)?

Moreover, addressing both generation and interpretation in tandem places a strong constraint on how the lexicon can be designed—an otherwise underconstrained knowledge engineering problem. Because the same planning domain (lexicon) is used for multiple problem instances, a relevant evaluation of a planning-based approach is its **coverage** of a range of various linguistic input (for REI tasks) and output (for REG tasks). One goal of this paper is to analyze several problematic referring expressions and draw conclusions from how they can be used to guide planning-based approaches to REG and REI.

2 Problems for Referring Expressions

In this section, we describe several linguistic issues using example referring expressions that are applied to two visual referential domains: KIN-DLE (Figure 1) and CIRCLE (Figure 2).

Imagine you are a clerk selling the *Amazon Kindle* in Figure 1. Three separate customers ask you




				
kindle	kindle touch	kindle touch 3g	kindle dx	kindle fire
\$79.00	\$99.00	\$149.00	\$379.00	\$199.00
5.98oz weight 2Gb hard drive 6.0" screen	7.50oz weight 4Gb hard drive 6.0" screen	7.80oz weight 4Gb hard drive 6.0" screen	18.90oz weight 4Gb hard drive 9.7" screen	14.60oz weight 8Gb hard drive 7.0" screen

Figure 1: The KINDLE referential domain containing 5 items: k_1, k_2, k_3, k_4 and k_5 .

to pass them:

(R1) *the big one*

(R2) *the inexpensive ones*

(R3) *a kindle touch*

2.1 The problem of lexical ambiguity

The problem with the referring expression (R1) is that it contains **lexical ambiguity**: did the customer intend the sense big_1 , which modifies the `size` attribute, or big_2 , which modifies the `hard_drive.size` attribute? Although one is much more likely, they are both mutually exclusive possibilities $\llbracket the\ big\ one \rrbracket = \{k_4\} \oplus \{k_5\}$.

What does this mean for planning-based approaches to REG and REI? For generation, it means that some words can cause the listener to draw multiple interpretations—but only in certain contexts (which provides an example of how word meanings draw from the context set). For interpretation, this means that we need a way to represent conflicting interpretations; and, if there are multiple interpretations for a given observed plan, we need a way to pick among the alternative interpretations.

2.2 The problem of gradable adjectives

Referring expression (R2) does not contain lexical ambiguity; however, it does suffer from **vagueness** as a result of having a gradable adjective, “inexpensive,” in the positive form modifying a plural noun, “ones.” Vagueness is problematic because it can lead to different interpretations depending on how the listener determines whether a referent is/a cluster of referents are INEXPENSIVE or \neg INEXPENSIVE (van Deemter, 2010). If we assume vagueness comes down to the interpreter inferring the speaker’s implicit *standard*—a specific value of `Price` as a cut off, we can exhaust all possibilities by considering all unique prices. At

one extreme, *only* the cheapest Kindle is inexpensive, at the other extreme *all* of the Kindles are inexpensive (i.e. the comparison class is a proper superset of the KINDLE domain): (R2) has four distinct denotations: $\llbracket the\ inexpensive\ ones \rrbracket = \{k_1, k_2\} \oplus \{k_1, k_2, k_3\} \oplus \{k_1, k_2, k_3, k_5\} \oplus \{k_1, k_2, k_3, k_5, k_4\}$. Like ambiguity, the use of a vague lexical unit can cause multiple distinct interpretations, and these outcomes are a function of the available options in the context set at the time the lexical unit is used.

2.3 The problem of unspecific descriptions

Referring expression (R3) is problematic because there are two possible denotations $\llbracket a\ kindle\ touch \rrbracket = \{k_2\} \vee \{k_3\}$ ⁵ but in a way that differs subtly from having two mutex interpretations like in (R1). The indefinite article “a” indicates that the speaker has not only communicated a description that matches multiple targets, but also the authority to choose on his behalf. Either $\{k_2\}$ or $\{k_3\}$ is acceptable. For planning-based approaches, this means that we should be able to represent *a choice* between multiple alternative targets in an interpretation, and distinguish it from the mutex alternatives created by vagueness and ambiguity.

2.4 The problem of word ordering

This and the next problem use this CIRCLE reference domain for their examples:

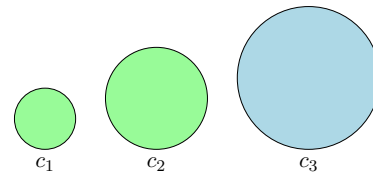


Figure 2: The CIRCLE referential domain.

Given the visual scene above, how would you interpret the following referring expressions?

(R4) *the biggest green shape*

(R5) *the second biggest green circle*

(R6) *the biggest*

(R7) *the first one*

⁵Our use of the disjunction operator here is non-standard, but we are not familiar of alternative notation for this distinction.

By incrementally evaluating each word in the sequence (R4), at the second word we have (R6) $\llbracket \text{“the biggest”} \rrbracket = \{c_3\}$. If every word’s meanings were combined by intersecting their denotations, adding the next word, $\llbracket \text{“green”} \rrbracket = \{c_1\} \vee \{c_2\} \vee \{c_1, c_2\}$, would denote nothing: $\llbracket \text{“the biggest”} \rrbracket \cap \llbracket \text{“green”} \rrbracket = \emptyset$.

An incremental planning system should be able to handle the non-monotonicity created by these so-called *subsecutive*⁶ adjectives: (R4) should yield an interpretation that is not included in (R6), even though (R6) is a prefix of (R4). If the model of REI aims to reflect human abilities, it should be able to incrementally process the words and switch between disjoint interpretations in real time, as the psycholinguistic research suggests (Altmann and Steedman, 1988; Tanenhaus, 2007).

Now, consider when multiple *subsecutive* adjectives occur before a noun, as in (R5). Does “second biggest”⁷ modify both $\llbracket \text{“green circle”} \rrbracket$ or just $\llbracket \text{“circle”} \rrbracket$? This depends on who is interpreting: when we asked 108 self-reported native English speakers on Mechanical Turk to interpret (R5) “the second biggest green circle” the uncertainty was high, but $\{c_1\}$ was favored over $\{c_2\}$ by 3:2 odds. An REI must decide whether it interprets on behalf of an individual or population; and REG approaches may want to avoid such expressions that can lead to conflicting interpretations.

The issues raised by *subsecutive* adjectives can be seen as symptoms of a more general problem: that of deciding how to combine the meanings of individual lexical units. This is the responsibility of a syntactic theory; its duty is to describe how the combinatoric constraints on surface forms relates to the “evaluation order” of their semantic parts. For planning-based approaches, the syntactic theory should be *incremental*, capable of producing an interpretation at any stage of processing, and *invertible*, capable of being used in generation and interpretation.

2.5 The problem of ellipsis

(R6) is missing a noun, and in (R7), “the first one,” the ordinal “first” appears without a grad-

⁶Characterizing adjectives set-theoretically, (Siegel, 1976; Partee, 1995) contrasted **intersective** and **subsecutive** meanings. Unlike intersective adjectives, the *subsecutive* adjectives cannot be defined independently of their nouns.

⁷The two words “second biggest” are treated as a single modifier: just as adjectives can modify nouns, ordinals like “second” modify superlatives like “biggest,” changing its meaning so that it skips over the first biggest.

able adjective. We take these to be instances of **ellipsis**: when the meaning of a word is present but its surface form is omitted. In our view, these expressions should be interpreted as:

(R6’) *the biggest [one_{NN}]*

(R7’) *the first [leftmost_{JJS}] one*

For planning-based approaches to REI, accommodating the phenomenon of ellipsis involves inferring missing actions—interleaving the partially⁸ observed actions of the speaker with abductively inferred actions of the listener (Hobbs et al., 1993; Benotti, 2010). For a REG, this means that the speaker can decide to elide some surface forms under certain conditions—such as if the listener is expected to infer it from context.

3 AIGRE: a belief-state planning approach to REI and REG

We used these problematic referring expressions to guide the design of our belief-state planner, AIGRE. Both REG and REI tasks begin with an *initial belief state* about a referential domain. In addition, the REI task is given a *referring expression* as input, and the REG task is given a *target set*.

3.1 Representing states (interpretations) as beliefs

We draw an analogy between the representation for an interpretation in a reference task and the concept of a belief state from artificial intelligence. A **belief state** characterizes a state of uncertainty about some lower layer, such as the world or another belief state. The standard representation of a belief state is the power set of the states in the lower layer, $b = \mathcal{P}(\mathcal{W})$, containing $2^{|\mathcal{W}|}$ members, or more generally as a probability distribution, $b = p(\mathcal{W})$, representing degree of belief.

Given a referential domain, R , REG systems that can refer to sets (van Deemter, 2000; Stone, 2000; Horacek, 2004) explore a hypothesis space containing $2^{|R|} - 1$ denotations, which is representationally equivalent to a belief state about the hypothesis space of only singleton referents. In our

⁸The actions are not fully observed because of ellipsis and, as we have seen with vagueness and ambiguity, different *senses* of a word can produce the same surface form of the lexical unit.

case, we want to be able to represent multiple interpretations about sets (due to unspecific descriptions, vagueness and ambiguity) so our hypothesis space contains $2^{2^{|R|}-1} - 1$ interpretations. This state-space grows large quick: for the CIRCLE domain, where $|R| = 3$, there are 127 denotations; while for KINDLE, where $|R| = 5$, there are over two billion.

Fortunately, there are ways to avoid this double-exponent. First, a belief state uses *lazy evaluation* to generate its contents: the members of the power set of the referential domain that are consistent with its intensional description and arity constraints (more details in section 3.1.1).

Second, the base exponent is avoided altogether, as we derive it by aggregating states from the planning graph. The initial belief state, one of complete uncertainty, implicitly represents $2^{|R|} - 1$ possible target sets: it is the branching of non-deterministic actions that gives rise to the first exponent (due to lexical ambiguity and vagueness; see 3.2). This gives a clear way to distinguish unspecific interpretations (when the listener has a choice over multiple targets) from the other mutually exclusive targets (choices that were artifacts of the interpretation process): If two candidate target sets belong to the same belief state, then they are the result of unspecificity; whereas, if they are in different belief states, then they are mutually exclusive. For example, a REI procedure may produce two belief states as results: $b_x = \{t_1\} \vee \{t_2\} \vee \{t_3\}$ and $b_y = \{t_1, t_2, t_3\}$. From this, we conclude its denotation is: $(\{t_1\} \vee \{t_2\} \vee \{t_3\}) \oplus \{t_1, t_2, t_3\}$.

In the field of automated planning, belief-state planning using heuristic search (Bonet and Geffner, 2000; Hoffmann and Brafman, 2005) has been used to relax some assumptions of classical planning, such as the requirement that the problem instance contains a single (known) initial state, and that each action in the planning domain only changes the state in a single (deterministic) way. Belief state planners allow one action to have multiple effects, and instead of finding linear plans, they output plan trees that describe which action the agent should take contingent upon each action's possible outcomes.

Furthermore, because a belief state represents an interpretation, we can stop and inspect the search procedure at any point and we will have a complete interpretation; thus, achieving the incremental property we desired.

3.1.1 Belief state implementation details

The key responsibilities of a belief state are to represent and detect equivalent or inconsistent information at the intensional level. Its function is to aggregate all actions' informational content and detect whether a partial information update is inconsistent or would cause the interpretation to be invalid (i.e., have no members). In AIGRE, belief states are represented as a collection of objects, called *cells*,⁹ which hold partial information and manage the consistency of information updates. AIGRE's belief states contain the following components:

- **target** an attribute-value matrix describing properties that a referent in the domain must entail to be considered consistent with the belief state.
- **distractor** an attribute-value matrix describing properties that a referent in the domain *must not* entail to be considered consistent with the belief state. This allows AIGRE to represent negative assertions, such as “*the not big one*” or “*all but the left one*.”
- **target_arity** an interval (initially $[0, \infty)$) representing the valid sizes of a target set.
- **contrast_arity** an interval (initially $[0, \infty)$) representing the valid sizes of the difference in the sizes of a target set and the set containing all consistent referents.
- **part_of_speech** a symbol (initially S) representing the previous action's part of speech.
- **deferred_effects** a list (initially empty) that holds effect functions and the trigger **part_of_speech** symbol that indicates when the function will be executed on the belief state.

A belief state does not have to store all $2^{|R|} - 1$ target sets; it can lazily produce its full denotation only when needed. It does this by generating the power set of all elements in the referential domain that entail the **target** description, do not entail the **distractor**, and are consistent with two arity constraints: The **target_arity** property requires the target set's size to be within its interval, and it is used to model number agreement and cardinal modifiers. The **contrast_arity** requires that the *difference* between a given target set and the largest target set in the belief state (the number of consistent referents) is a size within its interval, and is used to model the semantics of determiners and qualifiers.

Actions operate on AIGRE's belief states, yet the belief state influences much of the behavior of

⁹The idea behind *cells* comes from the *propagator framework* of (Radul and Sussman, 2009) and our Python library is available from <http://eventteam.github.io/beliefs/>

the action’s effects. As we will see in the next section, the contents of a belief state determine the number of effects an action will yield, the specific values within the effect’s belief (using late binding), and whether or not the update is valid.

3.2 Representing context-dependent actions

AIGRE’s lexicon is comprised of lexical units—actions that can change belief states. Each action/word is an instantiation of an action class and has (1) a syntactic category (part of speech), (2) a lexical unit, (3) a specific semantic contribution—determined in part by its syntactic category, (4) a fixed lexical cost, and (5) a computed effect cost. Actions are defined by instantiating class instances, for example:

```
GradableAdj('big', attr='size')
CrispAdj('big', attr='size', val=[5,∞))
```

When instantiating an action, the first argument is its lexeme in its *root form*; the class’ initialization method uses the root lexeme to also instantiate variant actions for each derivative lexical unit (e.g. plural, comparative, superlative, etc).

3.2.1 Actions yield effect functions, not states

Actions in AIGRE receive a belief state as input and lazily generate 0 or more effect functions as output, depending on the contents of the belief state. Unlike conventional planners, actions produce effect functions rather than successor states because (1) it allows us to defer the execution of an effect, as we describe in 3.2.3, (2) generating effect functions is fast; copying belief states is slow, and (3) actions can annotate the yielded effect functions with an *estimated cost*, giving the search process an additional degree of control over what successor state is created next. We view an action that does not yield any effects to be analogous to a traditional planning domain’s action that does not having its preconditions satisfied; unlike traditional domains, an action’s behavior is opaque until it is explicitly applied to a belief state.

3.2.2 Ambiguity and vagueness using non-deterministic actions

Gradable adjectives yield an effect for each same-named attribute¹⁰ (lexical ambiguity) for each value (vagueness) in the parent belief state’s consistent referents. For example, given the action `BIGJJ` applied to an initial belief state about the `KINDLE` referential domain, b_0 , the action

¹⁰Ordered by breadth-first traversal of targets’ properties.

yields a separate effect for each unique value of each unique attribute-path that terminates with `size` for all consistent referents. In this case, the referents have two `size` properties, `size` and `hard_drive.size`, each with 3 distinct values, so the `BIGJJ` action applied to b_0 yields 6 effects in total: $\text{BIG}(b_0) \rightarrow e_0, e_1 \dots e_6$. When executed on a belief state, e_0 would add the nested property `size` to its `target` property (if it doesn’t already exist) and then attempt to merge it with an interval beginning at the largest `size` value¹¹ of a referent consistent with b_0 : $[7, \infty)$.

Effects for vague and ambiguous actions proliferate: if the adjective `BIG` has s senses, and there are r referents compatible with the belief state, then it can yield as many as $s \times r$ effect functions. In section 3.3.1, we will show how the search algorithm can mitigate this complexity by conservatively generating effects.

3.2.3 Effects can be deferred until a trigger

We view subjective adjectives (see 2.4) as having their context-specific meaning evaluated within the scope of the noun’s meaning (i.e., after evaluating the noun). To achieve this without changing the words’ surface orderings, each adjective’s effects are deferred until a syntactic trigger: when the belief state’s `part_of_speech` indicates it has reached a noun state. Deferred effect functions are stored in the belief state’s `deferred_effects` queue along with a trigger. This solution makes the search harder: deferred actions have no immediate effect on the belief state, and so (in the eyes of the search algorithm) they do not move the belief toward the search goal.

3.3 Controlling search through belief states

A heuristic search planner must specify how to determine which state to expand next, and how to determine when a search process has succeeded, i.e., a **goal test function**. AIGRE approaches the first issue in a variety of ways: by (a) using a **heuristic function** to rank the candidate nodes so that the most promising nodes are expanded first (b) using an **action proposal function** to restrict the actions used to expand the current node (c) using a greedy **search algorithm** that does not generate all successor nodes.

Note that although both `REG` and `REI` tasks involve choosing belief-changing actions that map

¹¹Gradable (vague) values are represented with intervals, where one extreme is the *standard*.

an initial belief state onto a target belief state, the two search processes are subject to very different constraints. With generation, 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. We use these differences to create task-specific heuristics, action proposal mechanisms, and goal-test functions; and find that the interpretation task tends to search a much smaller space than that of generation.

3.3.1 Heuristic functions

For REI, the action proposal function is so restrictive that we can generate and test the entire search space; therefore, no heuristic is necessary.

For REG, the heuristic function characterizes its communicational objective: to describe the target(s) and none of the distractors. For this we use the **F1** score (*F-measure*) from information retrieval, because it rewards inclusion of targets (recall) and penalizes inclusion of distractors (precision). Given a belief state, \mathbf{s} , and the intended target set, $\hat{\mathbf{t}}$:

$$h(\mathbf{s}) = \max \mathbf{F1}(\hat{\mathbf{t}}, \mathbf{t}) \forall \mathbf{t} \in \mathbf{s} \quad (1)$$

This heuristic iterates over each target set, \mathbf{t} , in a belief state to find the biggest set difference according to the F1 score. By taking the worst possible score of any target, it always is greater than or equal to the true distance.

3.3.2 Goal test functions

For REI, a goal state is one in which all observations have been accounted for, and the belief state’s part of speech is a noun. For REG, a goal state is one in which only the targets are described (i.e. its heuristic, Equation 1, returns 0), and the belief state’s part of speech is a noun.

Both goal test functions impose a syntactic constraint: the requirement that plans terminate in a noun state. This all-or-nothing constraint, along with the language model in the action proposal function, forces the generated expressions to be syntactically well-formed English expressions.

3.3.3 Action proposal functions

While expanding a search state, instead of generating effects for every action in the lexicon, the action proposal restricts the set of actions that are considered. It is passed the parent belief state,

whose `part_of_speech` property tells the syntactic category of the last action that changed it. Actions are proposed only if they are consistent with a **language model** that describes valid transitions between syntactic categories. Our (limited) language model is expressed in a regular language: `DT? CD? (ORD? JJS)* JJ* (NN|NNS)+`.

For the problem of REI, we are licensed to make the action proposal function even more restrictive. AIGRE restricts those whose lexical units can produce the text that appears in the remaining observation sequence.

In addition to enforcing syntactic constraints, the action proposal function gives us a nice way to handle omitted actions. During interpretation, AIGRE allows **default actions**, representing elided words or conventional implicatures, to be inferred at a cost, but only under rare circumstances. A designated subset of actions are marked as default actions, indicating that they can be assumed even though their lexical unit is not present. A default action is only suggested if (1) none of the other actions have matched the remaining observed input text and (2) its precondition is met.

For example, the language model forbids the `ORD`→`NN` transition and the goal test function requires that all noun phrases terminate with a noun. Consequently, “*the second*” is interpreted as “*the_{DT} second_{ORD} [leftmost_{JJS}] [one_{NN}]*”, assuming the default actions `LEFTMOSTJJS` and `ONENN`. For (R6), the requirement of ending with a noun allows the subjective meaning of “*biggest*” to be evaluated: its deferred effect is triggered after `ONENN`.

3.3.4 Search strategies

Because the action proposal function is so restrictive for REI, the *entire* search space can be explored usually under a second. For REG, expanding the complete planning graph to a depth of 5 using ≈ 100 actions takes several minutes.

To complete the REG task efficiently, we have experimented with search strategies and found optimal **A* search** to be too slow. Although they give up guarantees of optimality and completeness, hillclimbing-based approaches rescue the REG task from having to expand every relevant action’s effect by committing to the first effect whose successor shows an improvement over the current state. Because we do not want the same results each time (non-deterministic output is characteristic of human reference generation (van Deemter

et al., 2011)), AIGRE randomly chooses effects with a probability inversely proportional to the action’s lexical cost, which is a kind of **stochastic hillclimbing**. The results are promising: non-deterministic outputs can be generated in usually less than a second (see Figure 4).

4 AIGRE’s Output for REI and REG

In lieu of a formal evaluation, we have included examples of AIGRE’s output for several tasks involving the CIRCLE and KINDLE reference domains: see Table 1 for output of the REG task; and Figures 5 and 6 for outputs of REI tasks.

AIGRE’s word costs were derived from their inverse token frequencies in the Open American National Corpus (Ide and Macleod, 2001). They are only an approximation and clearly do not accurately quantify the costs of human linguistic decisions. With this in mind, the referring expression’s denotations’ relative likelihoods, which are derived from costs, should not be given much credence. Our point here is that this large hypothesis space can be represented and searched efficiently.

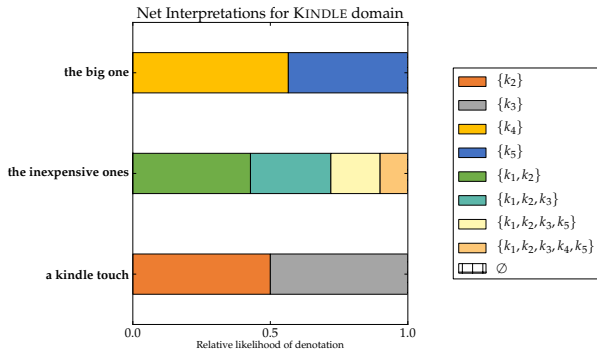


Figure 3: REI results for R1, R2 and R3 in the KINDLE domain. Each color represents a different target set, and more than one color in a bar indicates the interpretation is *uncertain*.

5 An example trace of a REI task

The interpretation task begins with an initial state containing the belief state b_0 about the KINDLE referring domain¹² (figure 1) and the referring expression, “any two cheap ones.” The search procedure begins by selecting actions to transform b_0 into successor states. The actions are sorted by how much of the prefix of the observed text they

¹²To AIGRE, each Kindle is an attribute-values matrix rather than a visual image.

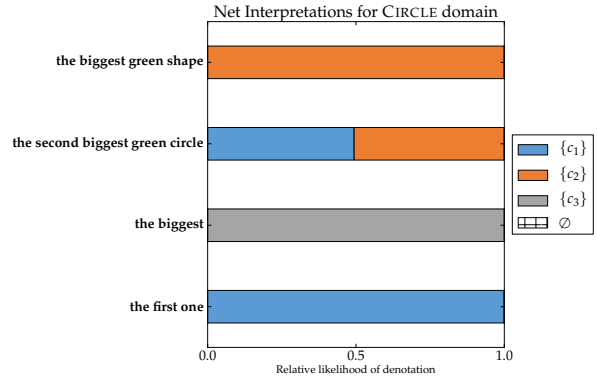


Figure 4: REI for R4-R7 in the CIRCLE domain.

match; and for “any two cheap ones,” the first action is ANY_{DT} and it transforms b_0 into b_1 :

$$b_0 = \begin{bmatrix} TARGET_ARITY & [0, \infty) \\ CONTRAST_ARITY & [0, \infty) \\ TARGET & [] \\ DISTRACTOR & [] \\ PART_OF_SPEECH & S \\ DEFERRED_EFFECTS & [] \end{bmatrix}$$

(Note: For lack of space, we just show the parts of the belief state that change)

$$b_1 = \begin{bmatrix} CONTRAST_ARITY & [1, \infty) \\ PART_OF_SPEECH & DT \end{bmatrix}$$

The `contrast_arity` property allows AIGRE to represent the notion of conveying a choice from alternatives, as with the indefinite meanings of “some” or “any,” as well as the fact that definite descriptions take the maximal set.¹³

Applying the effect of the action, TWO_{CD} , for the word “two” transforms b_1 into b_2 :

$$b_2 = \begin{bmatrix} TARGET_ARITY & [2, 2] \\ PART_OF_SPEECH & CD \end{bmatrix}$$

To be concrete, the initial belief state, b_0 , models all 31 groupings of referents: $b_0 \models \{k_1\}, \{k_3, k_5\}, \dots$; the belief state b_1 contains 30 sets—all but the set containing all 5 kindles; and b_2 represents $\binom{5}{2} = 10$ alternative sets.

The action $CHEAP_{JJ}$ corresponding to the gradable adjective “cheap” is non-deterministic: it yields a different effect for each distinct attributes’ values, starting with the lowest price, \$79.00. This

¹³The power set of the belief state’s referents forms a lattice under the subset operator, and for the definite article “the” we only want the top row. We model its meaning with a deferred effect that sets `contrast_arity` to $[0,0]$ after a noun. The indefinite article “a” sets `contrast_arity` to $[1,\infty)$ and `target_arity` to 1; “a” has the same meaning as “any one.”

TARGET	SECONDS	REFERRING EXPRESSIONS (AND COSTS)
$\{c_1\}$	0.66 ± 0.3	the small one (2.3), the left one (2.4), the smaller one (2.4), the smallest one (2.4), the leftmost one (2.4) ...
$\{c_2\}$	1.05 ± 0.5	the center one (2.4), the medium-sized one (2.4), the center circle (2.4), the green big one (3.4)
$\{c_3\}$	1.63 ± 1.1	the blue one (2.3), the right one (2.3), the big one (2.3), the large one (2.4), the larger one (2.4)...
$\{c_1, c_2\}$	0.37 ± 0.1	the green ones (2.3), the green circles (2.3), the 2 green ones (3.4), the small ones (3.4)
$\{c_1, c_3\}$	0.52 ± 0.1	the 2 not center ones (4.5), the 2 not center circles (4.5), the 2 not medium-sized ones (4.5)
$\{c_2, c_3\}$	0.41 ± 0.1	the right ones (3.4), the 2 right ones (4.4), the 2 right circles (4.4), the 2 big ones (4.5)
$\{c_1, c_2, c_3\}$	0.19 ± 0.1	the ones (1.2), the circles (1.2), the 3 ones (2.3)
$\{k_1\}$	3.24 ± 2.0	the left one (2.4), the light one (2.4), the small cheap one (3.5), the small cheapest one (3.5)
$\{k_2\}$	0.94 ± 0.2	the left touch (3.4), the small center one (3.5), the small center touch (3.6), the small center cheap one (4.7)
$\{k_3\}$	1.11 ± 1.0	the center one (2.4), the small heavy one (3.5), the small heavier one (3.5), the small heaviest touch (3.6) ...
$\{k_4\}$	0.20 ± 0.2	the kindle dx (1.2), the big one (2.3), the big kindle dx (2.4)
$\{k_5\}$	0.19 ± 0.1	the kindle fire (1.2), the right one (2.3), the right kindle fire (2.4)

Table 1: AIGRE’s outputs for REG tasks (each repeated for 20 trials). If the output is **bold**, it means that when we fed the referring expression back to AIGRE as a REI task, it was able to derive multiple alternative interpretations and the referring expression is uncertain.

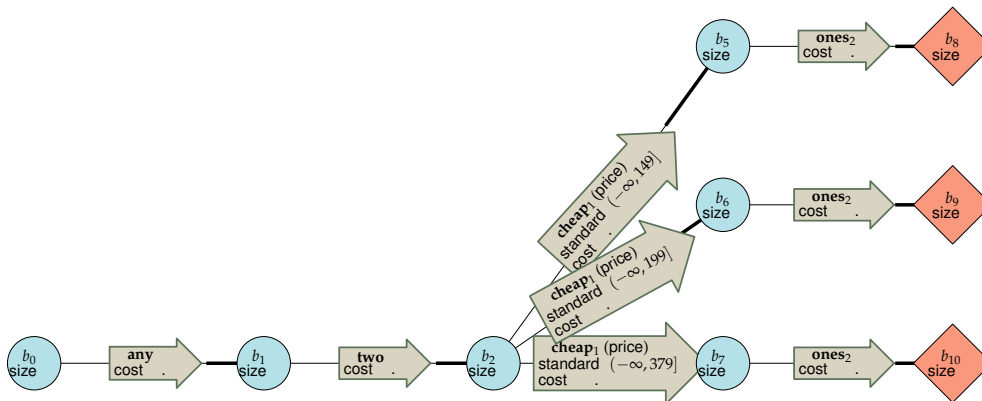


Figure 5: The **planning graph** for interpreting, “any two cheap ones.” Search proceeds from the initial state b_0 rightward toward goal states (diamonds). The labeled edges represent the actions, and contain the cumulative path costs. Only intermediate states that lead to a goal are shown—even though CHEAP_{JJ} initially had 5 successors, two were *invalid* belief states because they had 0 members.

effect *adds* a new attribute `target.price` to the belief state and sets its value to be the open interval $(-\infty, 79.00]$. The action’s next effect creates a separate belief state for the second lowest price from the referents, \$99.00, and so on, all the way up to the most expensive price, \$379.00.

$$b_3 = \left[\text{TARGET} \left[\text{PRICE} \left(-\infty, 79.00 \right] \right] \right]$$

$$b_4 = \left[\text{TARGET} \left[\text{PRICE} \left(-\infty, 99.00 \right] \right] \right]$$

$$b_5 = \left[\text{TARGET} \left[\text{PRICE} \left(-\infty, 149.00 \right] \right] \right]$$

...

The last word, “ones,” invokes an action `ONESNNS` whose effect adds the `target.type=entity` property to the belief state and then merges `targetset.arity` with $[2, \infty)$ because it is plural (though its value doesn’t change).

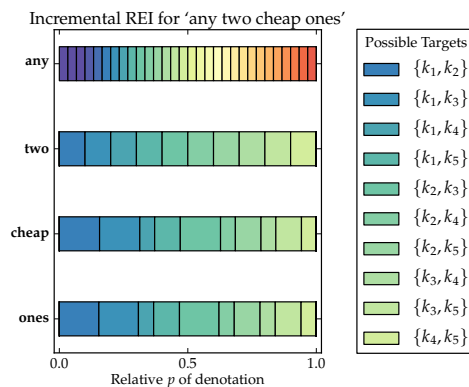


Figure 6: All denotation’s relative likelihoods. Each row corresponds to a column of the planning graph in Figure 5: the first row, “any,” is just node b_1 and the last row is the aggregate of the belief states b_8 , b_9 and b_{10} —derived by summing all the denotations’ inverted costs.

References

- Gerry Altmann and Mark Steedman. 1988. Interaction with context during human sentence processing. *Cognition*.
- Douglas E Appelt. 1985. Planning English referring expressions. *Artificial Intelligence*.
- Chris L Baker, Joshua B Tenenbaum, and Rebecca R Saxe. 2007. Goal inference as inverse planning. *Proceedings of the 29th annual meeting of the cognitive science society*.
- Luciana Benotti. 2010. Implicature as an Interactive Process. *Ph.D. Thesis*.
- Bernd Bohnet and Robert Dale. 2005. Viewing referring expression generation as search. *Proc. IJCAI-05*.
- Blai Bonet and Hector Geffner. 2000. Planning with Incomplete Information as Heuristic Search in Belief Space. *AIPS 2000: Proceedings of the Conference on Artificial Intelligence Planning Systems*.
- Blai Bonet and Hector Geffner. 2001. Planning as heuristic search: New results. *Artificial Intelligence*.
- Blai Bonet, Gábor Loerincs, and Hector Geffner. 1997. A Robust and Fast Action Selection Mechanism for Planning. In *Proceedings of AAAI-1997*.
- Robert Dale and Ehud Reiter. 1995. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*.
- Laurence Danlos and Fiammetta Namer. 1988. Morphology and cross dependencies in the synthesis of personal pronouns in Romance languages. In *COLING '88: Proceedings of the 12th conference on Computational linguistics*.
- Stefan Edelkamp and Stefan Schroedl. 2011. *Heuristic Search*. Morgan Kaufmann.
- Konstantina Garoufi and Alexander Koller. 2010. Automated planning for situated natural language generation. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*.
- Konstantina Garoufi and Alexander Koller. 2011. Combining symbolic and corpus-based approaches for the generation of successful referring expressions. In *ENLG '11: Proceedings of the 13th European Workshop on Natural Language Generation*.
- Christopher W Geib and Mark Steedman. 2006. On Natural Language Processing and Plan Recognition. *Proceedings of the 20th International Joint Conference on Artificial Intelligence*.
- Malik Ghallab, Dana Nau, and Paolo Traverso. 2004. *Automated Planning*. Morgan Kaufmann.
- Peter A Heeman and Graeme Hirst. 1995. Collaborating on referring expressions. *Computational Linguistics*.
- Jerry R Hobbs, Mark E Stickel, Douglas E Appelt, and Paul Martin. 1993. Interpretation as abduction. *Artificial Intelligence*.
- Jörg Hoffmann and Ronen Brafman. 2005. Contingent Planning via Heuristic Forward Search with Implicit Belief States .
- Joerg Hoffmann. 2001. FF: The Fast-Forward Planning System. *AI Magazine*.
- Helmut Horacek. 2004. On Referring to Sets of Objects Naturally. In *Natural Language Generation*.
- Nancy Ide and Catherine Macleod. 2001. The american national corpus: A standardized resource of american english. In *Proceedings of Corpus Linguistics 2001*, volume 3.
- Alexander Koller and Jörg Hoffmann. 2010. Waking up a sleeping rabbit: On natural-language sentence generation with FF. In *Proceedings of AAAI 2010*.
- Alexander Koller and Ronald P A Petrick. 2011. Experiences with planning for natural language generation. *Computational Intelligence*.
- Alexander Koller and Matthew Stone. 2007. Sentence generation as a planning problem. *Annual Meeting of the Association of Computational Linguistics*.
- Alexander Koller, Andrew Gargett, and Konstantina Garoufi. 2010. A scalable model of planning perlocutionary acts. In *Proceedings of the 14th Workshop on the Semantics and Pragmatics of Dialogue*.
- Emiel Krahmer and Mariët Theune. 2002. Efficient generation of descriptions in context. *Proceedings of the ESSLLI workshop on the generation of nominals*.
- Emiel Krahmer and Kees van Deemter. 2012. Computational Generation of Referring Expressions: A Survey. *Computational Linguistics*.
- Barbara Partee. 1995. Lexical semantics and compositionality. *An invitation to cognitive science: Language*.
- Alexey Radul and Gerald Jay Sussman. 2009. The Art of the Propagator. *Technical Report MIT-CSAIL-TR-2009-002, MIT Computer Science and Artificial Intelligence Laboratory*.
- Miquel J Ramírez and Hector Geffner. 2010. Probabilistic plan recognition using off-the-shelf classical planners. *Proceedings of the Conference of the Association for the Advancement of Artificial Intelligence (AAAI 2010)*.
- Ehud Reiter. 1994. Has a consensus NL generation architecture appeared, and is it psycholinguistically plausible? *Proceedings of the Seventh International Workshop on Natural Language Generation (INLG 1994)*.

- Muffy E A Siegel. 1976. Capturing the adjective. *PhD. Thesis. University of Massachusetts Amherst.*
- Matthew Stone and Bonnie Webber. 1998. Textual Economy through Close Coupling of Syntax and Semantics. *arXiv.org.*
- Matthew Stone, Christine Doran, Bonnie L. Webber, Tonia Bleam, and Martha Palmer. 2003. Microplanning with communicative intentions: The spud system. *Computational Intelligence*, 19(4):311–381.
- Matthew Stone. 2000. On identifying sets. In *INLG '00: Proceedings of the first international conference on Natural language generation.*
- Michael K Tanenhaus. 2007. Spoken language comprehension: Insights from eye movements. *Oxford handbook of psycholinguistics.*
- Kees van Deemter, Albert Gatt, Roger P G van Gompel, and Emiel Krahmer. 2011. Toward a Computational Psycholinguistics of Reference Production. *Topics in Cognitive Science.*
- Kees van Deemter. 2000. Generating vague descriptions. In *INLG '00: Proceedings of the first international conference on Natural language generation.*
- Kees van Deemter. 2010. Not Exactly: In Praise of Vagueness. *Oxford University Press.*