Towards Situated Open World Reference Resolution

Tom Williams
Saurav Acharya
Matthias Scheutz
Human-Robot Interaction Laboratory
Tufts University, Medford, MA, USA
{williams,sachar01,mscheutz}@cs.tufts.edu

Stephanie Schreitter
Austrian Research Institute for Artificial Intelligence
Freyung 6/6, A-1010, Vienna, Austria
stephanie.schreitter@ofai.at

Abstract

Natural language dialogue provides the opportunity for truly natural human-robot interaction. A robot participating in natural language dialogue must identify or create new representations for referenced entities if it is to discuss, reason about, or perform actions involving that entity, a capability known as *reference resolution*. In previous work we presented algorithms for resolving references occurring in definite noun phrases. In this paper we propose an algorithm for resolving references in a wider array of linguistic forms, using the *Givenness Hierarchy*.

Introduction

Natural language dialogue provides the opportunity for truly natural human-robot interaction. A robot participating in natural language dialogue must identify or create new representations for referenced entities if it is to discuss, reason about, or perform actions involving that entity, a capability known as *reference resolution*. In previous work we presented algorithms for resolving references occurring in definite noun phrases (Williams et al. 2013; Williams and Scheutz 2015a; 2015b). We designed those algorithms to handle open worlds (i.e., in which not all entities are known of *a priori*) and (in the case of (Williams and Scheutz 2015a; 2015b)) uncertain contexts (i.e., in which knowledge of whether a certain entity has a certain probability can be assigned a degree of uncertainty) because such contexts are commonplace in human-robot interaction scenarios.

In this paper we propose an algorithm for resolving references in a wider array of linguistic forms, including indefinite noun phrases and pronominal expressions. The proposed algorithm uses the *Givenness Hierarchy*(Gundel, Hedberg, and Zacharski 1993), a linguistic framework which associates the form of a referential expression (e.g., whether a pronominal, definite noun phrase or indefinite noun phrase is used) with a presumed "cognitive status" of that expressions' referents (e.g., whether they are in the focus of attention, short term memory or long term memory). Other researchers have proposed reference resolution algorithms using *portions* of the GH, but to the best of our knowledge our algorithm is the first to implement the GH in its *entirety*. Our algorithm also improves on previous algorithms through its ability to handle open-world and uncertain contexts.

The rest of the paper proceeds as follows. First, we describe the GH in more detail. Second, we describe previous reference resolution algorithms which have used the GH. Third, we present the results of an empirical study which suggests modifications which should be made to the GH. Fourth, we describe our proposed algorithm, aspects of which are motivated by those suggestions. Finally we lay out the work which will be necessary to complete our proposed implementation.

The Givenness Hierarchy

Gundel et al.'s *Givenness Hierarchy* (GH), as seen in Fig. 1, contains six nested levels or tiers of cognitive accessibility (Gundel 2010). For example, any piece of information that is in *FOCUS* is also activated, familiar, uniquely identifiable, referential, and type identifiable, whereas a piece of information that is *at most* familiar is also uniquely identifiable, referential and type identifiable, but not in focus or activated.

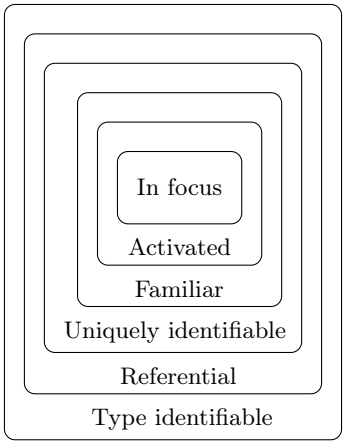


Figure 1: The Givenness Hierarchy

Each level of the GH corresponds with a different cog-

nitive status and is cued by a different set of linguistic forms, as seen in Table 1. Row one of Table 1 indicates that when a speaker uses "it" to refer to an entity, she believes that that entity is in her interlocutor's focus of attention. Row three indicates that when a speaker uses "that N" to refer to an entity, she believes that that entity is in her interlocutor's long term memory (and may or may not also be in her interlocutor's working memory or focus of attention).

Table 1: Cognitive Status and Form in the Givenness Hierarchy

Level	Cognitive Status	Form
In focus	in focus of attention	*it*
Activated	in working memory	*this,that,this* N
Familiar	in LTM	*that* N
Uniquely id-able	in LTM or new	*the* N
Referential	new	indef. *this* N
Type id-able	new or hypothetical	*a* N

Gundel et al.'s "coding protocol" suggests the cognitive status to associate with different pieces of information (Gundel et al. 2006). For example, the coding protocol suggests that the syntactic topic of the immediately preceding sentence should be *in focus*, speech acts and targets of gestures or sustained eye gaze should be *activated*, and any entity mentioned previously in the current dialogue should be *familiar*.

Taken together, the GH and its coding protocol represent a powerful framework for reference resolution, as they provide (1) data structures needed for reference resolution, (2) guidelines regarding population of those data structures, and (3) guidelines regarding retrieval from those data structures. Furthermore, Gundel et al. have provided strong experimental justification for their framework (e.g., (Gundel et al. 2010)).

Several previous research efforts in Human-Robot or Human-Agent interaction have made use of the GH, as its guidelines for simultaneously dealing with information coming from one's environmental, dialogue, and pre-existing knowledge make it especially valuable for situated contexts. However, while several such efforts have made tangential use of the GH (e.g., Byron et al.'s use of the Givenness Hierarchy's six levels as possible outputs of a learned decision tree for referential expression generation), we are only aware of two previous attempts to make full use of the GH. In the next section we describe those two implementations.

Previous Implementations

The first implementation of the GH that we will examine is the implementation presented in (Kehler 2000). In that work, Kehler et al. present the modified hierarchy seen in Fig. 2. As seen in Fig. 2,Kehler et al. omit the referential and type identifiable levels of the GH. They omit these tiers because they are primarily interested in pen-and-tablet interfaces, and when using such an interface it is unlikely for one to refer to unknown or hypothetical entities. Kehler et al. used their modified hierarchy to craft four simple rules which they found capable of resolving all the references they encountered (presented here verbatim):

1. If the object is gestured to, choose that object
2. Otherwise, if the currently selected object meets all semantic type constraints imposed by the referring expression (i.e., "the museum" requires a museum referent; bare forms such as "it" and "that" are compatible with any object), choose that object.
3. Otherwise, if there is a visible object that is semantically compatible, then choose that object (this happened three times; in each case there was only one suitable object).
4. Otherwise, a full NP (such as a proper name) was used that uniquely identified the referent.

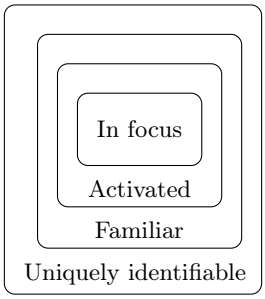


Figure 2: Kehler's Modified Hierarchy

The second implementation of the GH we will examine is the implementation presented in (Chai, Prasov, and Qu 2006). Chai et al. identify two principle problems with this implementation (Chai, Prasov, and Qu 2006). First, it is impossible to identify or resolve ambiguities using the four rules above. For example, if one cannot unambiguously determine which of several objects a gesture was targeting, problems may arise when considering rule one. If one cannot unambiguously determine which of several objects an underspecified referential expression was targeting, problems may arise when considering rule three. Second, Kehler et al.'s implementation is unable to handle utterances containing multiple referential expressions or gestures.

To address these concerns, Chai et al. created their own implementation of the GH (Chai, Prasov, and Qu 2006), in which they combined a subset of the GH with Grice's theory of Conversational Implicature (Grice 1970)to produce the modified hierarchy seen in Fig. 3.

Chai et al. use Grice's theory of Conversational Implicature (Grice 1970) to argue that gestures must have the highest cognitive status, since a gesture intentionally singles out an entity or group of entities. The second tier in their hierarchy is "Focus", which subsumes Gundel's *in focus* and *activated* tiers. The third tier in their hierarchy is "Visible", which subsumes Gundel's *familiar* and *uniquely identifiable* tiers. Finally,

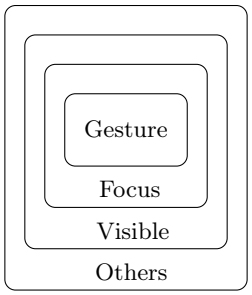


Figure 3: Chai's Modified Hierarchy

the lowest level of their hierarchy is the "Others" tier, which subsumes Gundel's *referential* and *type identifiable* tiers. This final tier does not appear to be used by Chai et al. or to be accessible using their algorithm. Like Kehler et al., Chai et al. are primarily interested in reference resolution in the context of an interface in which it is unlikely for one to refer to unknown or hypothetical entities. In particular, Chai et al. investigate reference resolution in the context of a graphical interface with which participants interact through speech, pointing, and circling.

Chai et al. present a greedy reference resolution algorithm which makes use of their modified hierarchy. This algorithm first assigns a score between each referential expression in a given utterance and each entity contained in three vectors: (1) A vector associated with the first tier of their modified hierarchy, containing all entities that have been gestured towards during the utterance, (2) a vector associated with the second tier of their modified hierarchy, containing all other entities thought to be in focus, and (3) a vector associated with the third tier of their modified hierarchy, containing all other visible entities. The score assigned to each entity contained in one of these vectors is calculated by multiplying (1) the probability of the entity being selected from it's vector, (2) the probability of that tier being selected given the linguistic form of the examined referential expression, and (3) the "compatibility" between the examined referential expression and the examined entity. This compatibility is said to be 1 if the target entity has all the properties mentioned in the referential expression, is of the same type as that mentioned in the referential expression (if any), has the same name as that mentioned in the referential expression (if any), and was gestured towards at the same time as the referential expression was uttered (if any gestures occurred). If any of these conditions do not obtain, the compatibility is said to be 0. This compatibility is thus *binary* in nature and does not account for uncertainty.

Once the algorithm has scored all visible entities, it marks the referential expression most likely to have been used to describe each entity. The algorithm then proceeds top-to-bottom through the hierarchy of vectors (i.e., from "Gesture" to "Visible"), greedily assigning the best match. For example, consider Example 1.

(1) Compare **it** with **this house** and **this one**.

When this utterance is heard, the algorithm first scores each of the three bolded referential expressions against each visible entity, and marks for each entity which of the three expressions is most likely to refer to it. The algorithm then assigns to each referential expression the highest scored entity, if any, associated with it in the "Gesture" vector. If any expressions are left unassigned, the process is repeated with the "Focus" vector, and then with the "Visible" vector.

We believe that this algorithm and modified hierarchy are insufficient for realistic human-robot interaction scenarios. First, the algorithm presented by Chai et al. assumes that it always knows with complete certainty whether or not an entity has a certain property. In reality, an agent may only be able to say that an entity has a certain property *with some probability* or *with some degree of confidence*. Furthermore, an agent may be cognizant of the fact that it simply does not know whether an entity has a certain property.

Second, consider the following command, of a form not unreasonable to expect in many human-robot interaction scenarios:

(2) Get **my laptop** from **my office**, and if you see **a charger** bring that too.

This sentence contains three referential expressions (i.e., the three bolded noun phrases). None of these three expressions could be properly handled using Chai's algorithm and modified hierarchy.

1. **my laptop** cannot be resolved because its referent is (presumably) not currently visible.
2. **my office** cannot be resolved because its referent is (presumably) not currently visible. What is more, **my office** is not an object, per se, and it is unclear whether Chai's modified hierarchy would be equipped to handle such entities, which cannot be gestured at in the same way as objects or icons.
3. **a charger** cannot be resolved because its referent is (presumably) not currently visible. What is more, it is not even known to exist, as it is *hypothetical* in nature. In order to resolve referents to such entities, it is necessary to model one's environment as an *open world* in which new entities may be added through experience or through dialogue. Chai's algorithm and modified hierarchy are not equipped to handle either type of new entity.

Third, a robot may need to resolve references to events, speech acts, or other entities that *cannot* physically exist, as seen in Examples 4a and 4b.

(3) I'm sorry, but I failed to complete the task.

(4) a. Can you repeat it?
b. Can you repeat that?

Fourth, because Chai's modified hierarchy combines the first two levels of the GH, their algorithm would not be

able to properly distinguish between Examples 4a and 4b even if it were able to handle references to physically nonexistent entities. When Example 4a is used to respond to Example 3, "it" unambiguously refers to "the task". However, this is not the case when Example 4b is used. The GH predicts that when a referential form associated with the *activated* level is used, one should prefer an activated referent (such as a speech act) to an in focus referent (such as the focus of the previous sentence), because if the speaker had meant to refer to an in-focus entity she could have used an in-focus cueing linguistic form. Thus, while Example 4b could refer to either the speech act or the failed task, the speech act should be preferred. Gundel et al. have empirically verified that these two hierarchical levels are distinguished between in a wide variety of languages beyond English, including Eegimaa, Kumyk, Ojibwe, and Tunisian Arabic (each of which is genetically and typologically unrelated to the other three) (Gundel et al. 2010).

Fifth, in natural human-robot dialogues it is not unreasonable to expect complex noun phrases such as:

(5) Do you see the red block on that blue block?

Because Chai et al.'s algorithm uses a greedy approach (instead of, e.g., the graph matching approach they used in previous work), it may choose an incorrect referent for the first considered referential expression, and may thus be unable to successfully resolve subsequent referential expressions. Chai et al. argue that using a greedy approach is advantageous because it allows significant pruning of the search space. However, their algorithm scores all entities against all referential expressions before employing its greedy approach. In a realistic Human-Robot Interaction scenario, this may not be practical, as a robot may know of hundreds or thousands of entities. Furthermore, the process of checking whether certain properties hold for all entities may be cost prohibitive. For example, while determining whether a given person is a man or not may be accomplished by a simple database lookup, determining whether two rooms are across from each other may require more expensive computation. An algorithm which performed such assessments lazily (i.e., only when needed) as the search space was pruned would potentially be much more efficient.

We argue that the shortcomings of previous GH implementations are sufficient to warrant a new implementation. In the following section, we present the results of an empirical study which suggest modifications to the GH itself which should be incorporated into such an implementation.

Experimental Results

In (Schreitter and Krenn 2014), Schreitter et al. examine how humans refer to objects in natural human-human and human-robot task descriptions. In that experiment, performed at the Technical University Munich, subjects participated in human-human or human-robot dyads consisting of a human *instructor* who was asked to explain four tasks in German to a human or robot *listener*. In this section, we analyze the results collected by Schreitter et al. in the human-human dyads of two of the four tasks. In the first task **(T1)**, the instructor and listener worked together to carry a board around a table and place it in a particular location (22 dyads). In the second task **(T2)**, the instructor explained and demonstrated to the listener how to connect two sections of tubing and affix the tubing to a box (16 dyads). We examine these two tasks in particular due to the significant difference in types of objects and references they include. We examine the human-human dyads in particular as this is the type of communication toward which human-robot communication strives.

Gaze and Gesture Handling

The results of this experiment emphasize the importance of developing a genuinely multi-modal approach to reference resolution: Without accounting for gesture or eye gaze, only 21.24% of task-relevant referring expressions in **T1** and 11.36% of task-relevant referring expressions in **T2** can be resolved. When accounting for gesture and eye gaze, 75.22% of such expressions can be resolved in **T1**, and 50% can be resolved in **T2**.

The use of eye gaze in particular was striking: In both **T1** and **T2**, participants frequently uttered underspecified definite noun phrases to refer to the object they were currently looking at. This is consistent with the GH coding protocol, which suggests that entities that are the subject of speech-simultaneous gesture or gaze should be considered to have the cognitive status *ACTIVATED*. However, it may not always be possible to identify the unique entity at which an interlocutor is looking. We thus suggest that *all* entities in an interlocutor's field of view be considered *ACTIVATED*. As *ACTIVATED* is roughly equivalent to short term memory (STM), this represents the possibility of any entity in the vicinity of an interlocutor's gaze being in her STM. (Gundel 2010) suggests that there may be differing degrees of being *in focus*. We suggest this consideration be extended to *ACTIVATED* entities. Thus, while all entities in an interlocutor's field of view may be considered *ACTIVATED*, those recently or sustainedly looked at would have higher *activation scores*.

In **T2**, some instructors used the underspecified referring expression "the tube" to refer to one of the two visible tubes. However, such references were easily resolved by interlocutors because the expressions were uttered while gazing or pointing at one of the two tubes. Instead of specifically checking what is the target of one's gaze or gesture when resolving a reference (as performed by Chai), one could resolve such a reference by increasing the activation score of the target tube when it is gazed or pointed at. If entities of a particular cognitive status are considered in decreasing order of activation level, then the tube that is the target of gaze or gesture will naturally be arrived at first, allowing it to be identified without the use of explicit gaze or gesture checking during the resolution process.

In **T1**, instructors often used underspecified noun phrases such as *Objekt* (object), *Gerät* (device), or *Ding* (thing) to refer to the uniquely identifiable board. These underspecified noun phrases would also be resolveable using the suggested approach.

In both **T1** and **T2**, linguistic forms cueing entities in *FOCUS* could often not be resolved using automatic anaphora resolution due to the linguistic distance between the form used and the last reference to the entity. Just as gesture and eye gaze may be used to increase the *activation level* of entities considered *ACTIVATED*, we suggest these measures also be used to increase the *focus level* of entities considered to be in *FOCUS*. This contrasts with the approaches presented by Gundel et al. and Chai et al., neither of which use visual cues when considering entities in the *FOCUS* tier.

Information Retrieval

As described previously, a drawback of Chai et al.'s approach is its strictly top-down search through the tiers of the GH. However, while the tiered nature of the GH suggests a *bottom-up* approach, this is not always appropriate. In **T1**, participants often referred to the board simply as "the board". According to the GH, a definite noun phrase of this form cues the *UNIQUELY IDENTIFIABLE* tier. However, starting at this tier and then proceeding up the hierarchy would suggest that the board in the task setting is a less appropriate choice than any other board the listener may have previously encountered. This problem manifests itself at the *FAMILIAR* level of the hierarchy as well. As such, we suggest that referential forms cueing the *UNIQUELY IDENTIFIABLE* and *FAMILIAR* tiers be processed in the same way as referential forms cueing the *ACTIVATED* tier, considering the tier they cue only after considering the *ACTIVATED* and *FOCUS* tiers. This is in line with (Gundel, Hedberg, and Zacharski 2012), in which Gundel et al. argue that for definite noun phrases, referents in one's current perceptual environment are preferred to those found by searching Long Term Memory (LTM). This suggestion yields Table 2, which lists search plans for forms cueing the first four levels of the GH. Unfortunately, Table 2 does not differentiate between using *this N* to cue the *ACTIVATED* tier and to cue the *REFERENTIAL* tier. While it may be possible to use factors such as tense to tell when one is using the *REFERENTIAL*-cueing sense, we believe that a first step towards appropriately handling the *REFERENTIAL*-cueing sense would be to treat all uses of *this N* as *ACTIVATED*-cueing so long as a suitable referent can be found at the *ACTIVATED* or *FOCUS* tiers, and otherwise treating such a use as *REFERENTIAL* cueing. A first step might also use a single process to handle *REFERENTIAL* and *TYPE-IDENTIFIABLE* cues, as both lead to the construction of new representations. These suggestions yield Table 3.

Table 2: Search Plans for GH tiers 1-4

Level	Search Plan
FOCUS	FOC
ACTIVATED	ACT → FOC
FAMILIAR	ACT → FOC → FAM
DEFINITE	ACT → FOC → LTM

Table 3: Search Plans for Complete GH

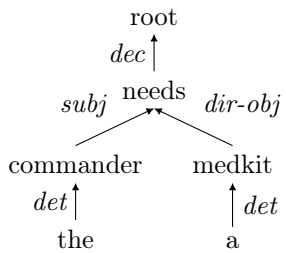
Level	Search Plan
FOCUS	FOC
ACTIVATED	ACT → FOC
FAMILIAR	ACT → FOC → FAM
DEFINITE	ACT → FOC → LTM
THIS-N-ACTIVATED	ACT → FOC → HYP
INDEFINITE	HYP

Our Approach

In this section, we propose an algorithm for resolving referential expressions associated with all six levels of the GH. This approach combines the suggestions proposed in the previous section with our previous reference resolution work ((Williams and Scheutz 2015a; 2015b)). We will first discuss how utterances are parsed and analyzed. We will then describe the data structures we use and how they are updated. Finally, we will describe how those data structures are used to resolve references in parsed utterances. All capabilities described in these sections are performed by components of the Distributed, Integrated, Affect, Reflection and Cognition (DIARC) architecture (Scheutz et al. 2007), as implemented in the Agent Development Environment (ADE) (Scheutz 2006; Scheutz et al. 2013).

Parsing

Trees representing the syntactic structure of incoming utterances are generated using a CCG parser with a dependency grammar. The structure of this tree is then analyzed to produce (1) a set of logical formulae representing the surface semantics of the utterance, (2) a set of "status cue" mappings for each referenced entity, and (3) the *type* of utterance which was heard. For example, the utterance "The commander needs a medkit" would be parsed into the following tree:



From this tree, we then extract:

1. The set of formulae $\{needs(X,Y) \wedge commander(X) \wedge medkit(Y)\}$.

2. The set of status cue mappings $\{X \rightarrow \text{DEFINITE}, Y \rightarrow \text{INDEFINITE}\}$.
3. The utterance type "STATEMENT" (indicated by the label "dec" on the arc pointing to the root node).

Data Structure Population

Our approach uses four data structures: *FOC*, *ACT*, *FAM*, and *LTM*, corresponding with the first four levels of the GH (levels five and six do not have associated structures, as they involve construction of new representations). In this section, we describe how these four data structures are populated, as summarized in Table 4. Lines marked with a star are not implemented, and represent future work.

Table 4: Contents of Relevant Data Structures

Level	Contents
FOCUS	Main clause subject of clause n-1 Syntactic Focus of clause n-1 * Event denoted by clause n-1
ACTIVATED	* Entities visible in int.'s region of attention * Focus of int.'s gesture, if any * Focus of int.'s sustained eye gaze, if any * Speech act associated with clause n-1 * All propositions entailed by clause n-1
FAMILIAR	All entities referenced in clause n-1 * The robot's current location
LTM	All declarative memory

For clause n of some natural language utterance, we first update *FOC*, *ACT* and *FAM* using (primarily) the guidelines specified in Gundel et al.'s GH coding protocol (Gundel et al. 2006). Linguistically, this entails placing the main clause subject, syntactic focus, and event denoted by clause n-1 into *FOC* (each of which may be extracted from the syntactic representation of clause n-1), placing the speech act and any propositions entailed by clause n-1 into *ACT*, and placing all entities referenced at all in clause n-1 into *FAM*. In addition, each location visited by the robot and its interlocutor should be placed into *FAM*, and any entities within the interlocutor's region of attention should be placed into *ACT*. The linguistic contents of *FOC* and *ACT* are reset after each clause, and *FAM* is reset after each dialogue. *LTM* is never reset. Once these structures are updated, we resolve all references contained in clause n.

Anaphora and Reference Resolution

To resolve the references in a given clause, that clause is first viewed as a graph whose vertices and edges are the respective variables and formulae seen in the semantics of that clause[1]. This graph is then partitioned into connected components. For each partition, GH-RESOLVE is used to resolve all references found in that partition, producing a set of variable-entity bindings.

Alg. 1 (GH-RESOLVE) takes three parameters: (1) S (the semantics of clause n), (2) M (the set of status cue mappings for clause n), (3) GH (containing the FOC, ACT, and FAM data structures), and (4) POWER (a module which performs Probabilistic, Open-World Entity Resolution to interface with LTM, as described in (Williams and Scheutz 2015b)). GH-RESOLVE first collects the variables appearing in S and sorts them with respect to the tier they are cued towards. For example, $X \rightarrow FOCUS$ and $Y \rightarrow FAMILIAR$ appear in M, then X will appear before Y (Alg. 1 line 2).

Algorithm 1 GH-RESOLVE($S, GH, POWER$)

1: S: set of formulae, M: set of status cue mappings, GH: FOC, ACT, and FAM data structures, $POWER$: a Probabilistic, Open-World Entity Resolver
2: $V = [v | v \in S.vars]$ sorted by $M(v)$
3: $\Theta = create\_plan\_table(M)$
4: $H = \emptyset$
5: **for all** $P \in \Theta$ **do**
6: $\quad P_d = [p | p \in P, p.tier = LTM]$
7: $\quad V_p =$ new list
8: $\quad$ **for all** $p \in (P \setminus P_d)$ **do**
9: $\quad\quad V_p = p.var \cup V_p$
10: $\quad\quad H = ASSESS(S, V_p, H, p.var, p.tier, POWER)$
11: $\quad$ **end for**
12: $\quad$ **for all** $h \in H$ **do**
13: $\quad\quad h = POWER.resolve(bind(S, h), order(P_d.vars)))$
14: $\quad$ **end for**
15: $\quad H = [h | h \in H, h.prob >= \tau_{resolve}]$
16: $\quad$ **if** $|H| > 0$ **then**
17: $\quad\quad$ BREAK
18: $\quad$ **end if**
19: **end for**
20: **if** $|H| \neq 1$ **then**
21: $\quad$ **return** H // AMBIGUOUS or UNRESOLVEABLE
22: **else**
23: $\quad$ **return** $POWER.assert(bind(S, H[0]))$
24: **end if**

Before GH-RESOLVE begins trying different variable-entity assignments, it must determine which data structures to examine when looking for those entities. This is determined by using the *plan* associated with each level of the hierarchy, as seen in Table 3. This table dictates that if M contains $X \rightarrow FOCUS$ then X must be looked for in the *FOCUS* data structure, whereas if M contains $X \rightarrow ACTIVATED$ then X must be looked for in the activated structure, and if no satisfactory match is found there, it must be looked for in the $FOCUS$ data structure.

(6) The box on this red boat

To handle multi-variable expressions, GH-RESOLVE creates a table Θ, storing all multivariable plan combinations. For example, if the referential expression

[1]To properly handle declerative and imperative utterances, we omit the formula associated with the main clause verb from consideration. As later discussed, future work will include using common-sense reasoning to account for this formula.

seen in Example 6 is parsed as: {*box(X)* ∧ *boat(Y)* ∧ *red(Y)* ∧ *on(X,Y)*} with status cue mappings {*X*→ *DEFINITE, Y*→ *THIS-N-ACT*}, Table 5 of exploration strategies will be created.

Table 5: Sample Strategy Table

Y	X
ACT	ACT
ACT	FOC
ACT	LTM
FOC	ACT
FOC	FOC
FOC	LTM
IND	ACT
IND	FOC
IND	LTM

After this table is created (line 3), an empty set of candidate hypotheses H is created. GH-RESOLVE then goes through Θ one row at a time until a solution is found or the end of the table is reached. For each table entry P, GH-RESOLVE first separates the variables for which it must query LTM from all other variables (line 6). It then intitializes an empty list V_p to hold variables that have been examined thus far for entry P (line 7). Next, it iterates over each (variable, tier) pair in that table row, as we will now describe.

Consider the first row of Table 5. GH-RESOLVE would first examine the first entry in this row, which says to look for Y's referent in the *ACT* data structure. The search through the appropriate data structure is effected through the call to *ASSESS* on line 10. When *ASSESS* is called with *p.tier* equal to *HYP* (an instruction to hypothesize a new entity), it creates a new binding between *p.var* and "?". Otherwise, it adds *p.var* to V_p, and POWER to find the maximum likelihood assignment from variables in V_p to known entities.

For example, if Example 6 is heard and there is one entity in *ACT* (e.g., *obj_13*), *ASSESS* would consult POWER to see to what degree *obj_13* could be considered to be a boat, and to what degree it could be considered to be red, and then create a hypothesis mapping Y to *obj_13* with probability equal to the product of the two probabilities returned by POWER. Once all formulae containing only variable *p.var* are examined, all those containing both *p.var and* any other previously examined variables are examined. For Example 6, this would involve inquiring to what degree the candidate entities for X could be considered to be "on" each candidate entity for Y. After each variable is considered, all candidate bindings whose likelihoods fall below a certain threshold are removed.

For example, if resolving Y produces hypothesis list

$$\{((Y \rightarrow obj\_13) \rightarrow 0.8), ((Y \rightarrow obj\_12) \rightarrow 0.75)\},$$

and resolving X produces the hypothesis list

$$\{((X \rightarrow obj\_5) \rightarrow 0.9)\},$$

these are combined into:

$$\{((Y \rightarrow obj\_13, X \rightarrow obj\_5) \rightarrow 0.72),$$
$$((Y \rightarrow obj\_12, X \rightarrow obj\_5) \rightarrow 0.675)\}.$$

If ASSESS determines that $on(X, Y)$ has probability 0.2 for the first of these hypotheses and 0.9 for the second, the two hypotheses are updated to

$$\{((Y \rightarrow obj\_13, X \rightarrow obj\_5) \rightarrow 0.144),$$
$$((Y \rightarrow obj\_12, X \rightarrow obj\_5) \rightarrow 0.6075)\}.$$

If τ_assess is set to 0.6, for example, then the first of these hypotheses would be removed.

GH-RESOLVE now considers all variables which were previously set aside because they were to be searched for in LTM. If any such variables exist, GH-RESOLVE considers each candidate binding in H. For each, S is bound using h's variable bindings, and an ordering of the variables V_h to be queried in LTM is created (e.g, based on the prepositional attachment observed in S). These bound semantics and variable ordering are then used by POWER to determine (1) whether any of the variables in V_h refer to unknown entities, and (2) which entities in LTM are the most probable referents for each other variable in V_h (line 13). The POWER resolution algorithm is beyond the scope of this paper, but is described in (Williams and Scheutz 2015b). The set of hypotheses H is then updated using these results.

Finally, once a solution is found or all table rows are exhausted, the *number* of remaining hypotheses is examined. If more or less than one hypothesis was found, GH-RESOLVE returns the set of solutions. This signifies that the referential expression was either ambiguous or unresolveable. If only one hypothesis remains, however, GH-RESOLVE uses the variable bindings of that hypothesis to update the set of semantics S, and then uses POWER to assert new representation for each variable bound to "?" (line 24). For example, if the results of Example 6 are a single hypothesis with probability 0.7 in which X is bound to *obj_4* and Y is bound to "?", POWER will create a new object (perhaps with identifier 5) with properties $\{boat(obj\_5), red(obj\_5), on(obj\_4, obj\_5)\}$ and $\{((Y \rightarrow obj_5, X \rightarrow obj_4) \rightarrow 0.7)\}$ will be returned. Once all partitions have been processed in this way, the resulting sets of bindings are combined into a comprehensive set of candidate binding hypotheses.

Conclusions and Future Work

We have proposed an open world reference resolution algorithm which uses the Givenness Hierarchy to handle definite noun phrases, indefinite noun phrases, and pronominal expressions. This allows our algorithm to handle a wider range of linguistic expressions than previous approaches. And, unlike previous approaches, our algorithm is able to handle open world and uncertain contexts. We thus argue that it is better suited for human-robot interaction scenarios than are previous

GH-enabled reference resolution algorithms. We have not yet evaluated the proposed algorithm on a robot as some components still remain to be adapted for use on robots. And, as seen in Table 4, the FOC, ACT, and FAM data structures are thus far only populated with directly referenced entities; future work will involve populating them with those entities observed visually or *inferred* from incoming utterances. Future work may also investigate better ways to determine whether a referential expression is cueing the *FAMILIAR* and *REFERENTIAL* levels.

We must also determine the best way to calculate activation and focus scores. While there have been many previous approaches towards measuring the salience of observed entities, we will need to develop a scoring system which is able to balance visual salience with a variety of other factors, such as duration of gaze and recency of mention in dialogue.

Future work will also include integration of commonsense reasoning capabilities: We do not currently consider the verb used in an utterance when resolving the references contained in that utterance. Common-sense reasoning may allow the robot to determine that some referents are less likely given, e.g., the action it is being asked to perform on that referent.

Finally, we must better integrate the proposed algorith with the rest of our robotic architecture. Currently, the most probable resolution hypothesis is bound to the set of semantics S, which is then sent to DIARC's pragmatic analysis component (Williams et al. 2015). However, if the set of resolution hypotheses produced by GH-RESOLVE has more or less than one member, it would be prudent to ask for clarification: If the set of hypotheses H is empty, the robot must alert its interlocutor that it doesn't know what entities he or she was refering to; If H has more than one member, the robot should ask its interlocutor which of those hypotheses is correct. After completing the implementation of our algorithm, this will be the immediate focus of our work.

Acknowledgments

This work was in part funded by grant N00014-14-1-0149 from the US Office of Naval Research.

References

Chai, J. Y.; Prasov, Z.; and Qu, S. 2006. Cognitive principles in Robust multimodal interpretation. *Journal of Artificial Intelligence Research* 27:55–83.

Grice, H. P. 1970. Logic and conversation. In et al., C., ed., *Syntax and Semantics 3: Speech Acts*. Elsevier.

Gundel, J. K.; Hedberg, N.; Zacharski, R.; Mulkern, A.; Custis, T.; Swierzbin, B.; Khalfoui, A.; Humnick, L.; Gordon, B.; Bassene, M.; and Watters, S. 2006. Coding protocol for statuses on the givennes hierarchy. unpublished manuscript.

Gundel, J. K.; Bassene, M.; Gordon, B.; Humnick, L.; and Khalfaoui, A. 2010. Testing predictions of the Givenness Hierarchy framework: A crosslinguistic investigation. *Journal of Pragmatics* 42(7):1770–1785.

Gundel, J. K.; Hedberg, N.; and Zacharski, R. 1993. Cognitive status and the form of referring expressions in discourse. *language* 274–307.

Gundel, J. K.; Hedberg, N.; and Zacharski, R. 2012. Underspecification of cognitive status in reference production: Some empirical predictions. *Topics in cognitive science* 4(2):249–268.

Gundel, J. 2010. Reference and Accessibility from a Givenness Hierarchy Perspective. *International Review of Pragmatics* 2(2):148–168.

Kehler, A. 2000. Cognitive Status and Form of Reference in Multimodal Human-Computer Interaction. In *Proceedings of the 14th AAAI Conference on Artificial Intelligence*.

Scheutz, M.; Schermerhorn, P.; Kramer, J.; and Anderson, D. 2007. First steps toward natural human-like HRI. *Autonomous Robots* 22(4):411–423.

Scheutz, M.; Briggs, G.; Cantrell, R.; Krause, E.; Williams, T.; and Veale, R. 2013. Novel mechanisms for natural human-robot interactions in the diarc architecture. In *Proceedings of AAAI Workshop on Intelligent Robotic Systems*.

Scheutz, M. 2006. ADE - steps towards a distributed development and runtime environment for complex robotic agent architectures. *Applied Artificial Intelligence* 20(4-5):275–304.

Schreitter, S., and Krenn, B. 2014. Exploring inter- and intra-speaker variability in multi-modal task descriptions. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 43–48. IEEE.

Williams, T., and Scheutz, M. 2015a. A domain-independent model of open-world reference resolution. In *Proceedings of the 37th annual meeting of the Cognitive Science Society*.

Williams, T., and Scheutz, M. 2015b. POWER: A domain-independent algorithm for probabilistic, open-world entity resolution. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Williams, T.; Cantrell, R.; Briggs, G.; Schermerhorn, P.; and Scheutz, M. 2013. Grounding natural language references to unvisited and hypothetical locations. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*.

Williams, T.; Briggs, G.; Oosterveld, B.; and Scheutz, M. 2015. Going beyond command-based instructions: Extending robotic natural language interaction capabilities. In *Proceedings of Twenty-Ninth AAAI Conference on Artificial Intelligence*.