Referring Expression Generation under Uncertainty: Algorithm and Evaluation Framework

Tom Williams and Matthias Scheutz
Human-Robot Interaction Laboratory
Tufts University
williams@cs.tufts.edu, matthias.scheutz@tufts.edu

Abstract

For situated agents to effectively engage in natural-language interactions with humans, they must be able to *refer* to entities such as people, locations, and objects. While classic *referring expression generation* (REG) algorithms like the Incremental Algorithm (IA) assume perfect, complete, and accessible knowledge of all referents, this is not always possible. In this work, we show how a previously presented *consultant* framework (which facilitates *reference resolution* when knowledge is uncertain, heterogeneous and distributed) can be used to extend the IA to produce *DIST-PIA*, a domain-independent algorithm for REG under uncertain, heterogeneous, and distributed knowledge. We also present a novel framework that can be used to evaluate such REG algorithms without conflating the performance of the algorithm with the performance of classifiers it employs.

1 Introduction

For situated agents to effectively engage in natural-language interactions with humans, they must be able to *refer* to those entities of interest to their interlocutors, such as people, locations, and objects. This task, known as *referring expression generation (REG)* is typically split into two sub-tasks: *content determination* (deciding which properties to use to describe a target), and *linguistic realization*, (choosing which words to use to communicate those properties) (Krahmer and Van Deemter, 2012). In keeping with tradition, we refer to content determination algorithms as REG algorithms.

Traditionally, REG algorithms make use of a *domain* (comprised of target referent m and a set of distractors X), where each entity in that domain is represented by an *attribute set* of properties and relations that hold for that entity (Dale and Reiter, 1995). The most traditionally successful such algorithm has been Dale and Reiter's *Incremental Algorithm (IA)*, which additionally takes a *preference ordering* P in which attributes are to be considered.

A variety of factors prevent many *situated* agents from using algorithms from this tradition. Crucially, to check whether an entity has a certain attribute, *IA* simply checks whether that attribute is a member of that entity's attribute set, producing a clear and unambiguous answer. But for many agents, it is imperative to represent the *uncertainty* of knowledge. The knowledge bases (KBs) of these agents may thus be *unable* to definitively state whether or not a given attribute holds for a given entity.

While there have been previous approaches to generating referring expressions (REs) under uncertainty, those algorithms have been explicitly designed to refer to *objects* in visual scenes, and as such are tightly integrated with visual classifiers (Zarrieß and Schlangen, 2016; Roy, 2002; Meo et al., 2014). This is problematic for least two reasons: First, intelligent agents may need to generate REs for a much wider class of entities than those appearing in a visual scene (e.g., agents, locations, ideas, utterances), which may not be possible if an REG algorithm is tightly coupled with *visual* classifiers. Second, due to this tight coupling, the evaluation of these algorithms conflates the performance of the REG algorithms themselves with the perfor-

Proceedings of The 10th International Natural Language Generation conference, pages 75–84,
Santiago de Compostela, Spain, September 4-7 2017.

mance of the visual classifiers they employ. For these reasons, previous algorithms for generating REs under uncertainty have only been evaluated relative to different versions of themselves, and not to other algorithms or to humans. We believe it is important to be able to talk separately about the design, efficacy, and integration of REG algorithms and the design, efficacy, and integration of property classifiers used by those algorithms. In this paper we present an REG algorithm that is *not* tightly integrated with specific property classifiers, but is easily extensible to allow for arbitrary property classifiers to be utilized within a general framework.

In addition to these two primary concerns, we raise a third, specific to the realities of modern integrated agent architectures. In many integrated agent architectures, such as DIARC (Scheutz et al., 2013) and ROS (Quigley et al., 2009), information may be distributed across a number of architectural components, rather than being stored in a single centralized KB, meaning that no central attribute set can be expected to be ready and available for use by an REG algorithm. While there has been much research on merging disparate knowledge bases (Lin, 1996; Liberatore and Schaerf, 1998; Konieczny, 2000), this is not always feasible or even desirable in integrated architectures (Williams and Scheutz, 2016).

In this paper, we address three main research challenges. First and most crucially, we address the need for REG algorithms that take into account the generator's uncertainty regarding entities' attributes, and which are not tied to a particular domain (e.g., visible objects). Second, we address the lack of a rigorous evaluation framework for systematically evaluating such algorithms, in a way that allows REG algorithms in this class to be compared to both each other and to humans. Finally, we address the need for such algorithms to take into account the realities of the distributed knowledge representation schemes used by modern integrated agent architectures.

To address these challenges, we present *DIST-PIA*: an *IA*-inspired REG algorithm designed to operate within our previously presented *consultant framework* (Williams and Scheutz, 2016), which provides access to uncertain, heterogeneous, and distributed knowledge. Furthermore, we present a novel two-stage evaluation framework in which human participants first assess the uncertainty that various attributes hold within a domain and to generate novel REs, and then evaluate the effectiveness of REs created from both human- and machine-generated sets of properties within that domain.

2 Previous Work

"Referring" has been referred to as the "fruit fly" of language due to the amount of study it has attracted (Van Deemter, 2016). The bulk of such study has focused on the content determination stage of REG. As previously noted, classic REG algorithms (e.g., *Full Brevity*, the *Greedy Algorithm* (Dale, 1989), and the aforementioned *Incremental Algorithm* (Dale and Reiter, 1995)) operate under a number of simplifying assumptions (such as certain knowledge on the part of both speaker and listener) that are not tenable in realistic interaction scenarios.

In this section, we will not attempt to survey the full scope of REG algorithms developed in the past few decades (for an excellent primer, we recommend Van Deemter (2016)'s recent book on the subject), but will instead focus on REG algorithms that have relaxed the constraint of certain knowledge.

Horacek (2005) presents an algorithm that reasons about the certainty that *a listener* will be able to recognize that a target referent has certain attributes, choosing an utterance that minimizes recognition failure. This is an example of *audience design* in which the *listener's* knowledge and capabilities are taken into account. Horacek's algorithm does not, however, take into account the uncertainty of the *agent's* knowledge, which we argue must be taken into account before audience design is considered.

In the graph-based approach presented by Sadovnik et al. (2013), computer vision classifiers are used to assess both the uncertainty of the agent's knowledge as well as for audience design. If Sadovnik's algorithm cannot generate an RE that is sufficiently likely to disambiguate the target, it is re-run using the attributes of both the target and one of its neighbors. While this approach relaxes the assumption of completely certain knowledge, it imposes others, as it is specifically tailored to use vision-based techniques alone. Furthermore, by giving equal weight to the attributes of target and anchors, the algorithm generates REs that curiously under-describe the target relative to anchors (e.g.,

"*The person* on the right of a person who is not Asian and has eye glasses and is smiling and has bangs and whose mouth is not closed").

Finally, Fang et al. (2013) (see also (Fang et al., 2015)) present an approach which more systematically handles attributes by expanding a hypergraph of properties until the selected properties disambiguate the target referent. When choosing how to expand this hypergraph, Fang chooses the attribute of minimal cost, taking into account the uncertainty of the agent's knowledge as well as the preference of that attribute (in the sense used by the IA). While this approach moves in the right direction, it once again assumes the exclusive use of computer vision classifiers, that all entities are objects in a visual scene, and that information about all such entities is stored in a single, centralized data structure.

3 Consultant Framework

We previously presented a framework of "consultants" that allows information about entities to be assessed when knowledge is uncertain, heterogeneous, and distributed (Williams and Scheutz, 2016). Specifically, each consultant c facilitates access to one KB k, and must be capable of at least four functions:

1. providing a set c_{domain} of atomic entities from k,
2. advertising a list $c_{constraints}$ of constraints that can be assessed with respect to entities from c_{domain},
3. assessing constraints from $c_{constraints}$ with respect to entities from c_{domain}, and
4. adding, removing, or imposing constraints from $c_{constraints}$ on entities from c_{domain}.

While these capabilities were designed to facilitate reference resolution, they can also facilitate REG, which requires a set of distractors to rule out (Capability 1), a list of constraints that can be used to rule out those distractors (Capability 2), and a means of checking whether those constraints do in fact rule out distractors (Capability 3).

Recall, however, that the IA considers constraints according to a *preference ordering*. For example, it may be more preferable to use an entity's *type* to describe it than to use its *color*, more preferable to use color rather than size, and so on. To use the aforementioned consultant framework for REG, we thus require a more specific second capability:

2. advertising a list $c_{constraints}$ of constraints that can be assessed with respect to entities from c_{domain}, *and that is ordered by descending preference*.

With this modification, we now have a framework which provides access to uncertain knowledge from different domains and of heterogeneous representation which is distributed throughout a robot architecture, and which is configured to interface well with the *IA*. In the next section, we describe how we have similarly modified the *IA* in order to take advantage of this framework.

4 Algorithm and Walkthrough

In this section, we present *DIST-PIA*, the Distributed, Probabilistic Incremental Algorithm[1], a modified version of the *Incremental Algorithm (IA)* (Dale and Reiter, 1995). For each property p in the list of properties attributed to the target referent, *IA* checks whether p is *not* attributed to any distractors; if so, p is added to the list of properties to communicate, and the ruled-out distractors are removed from the set of distractors. This process terminates when all distractors are eliminated or there are no properties left to consider.

When information is uncertain and distributed across multiple KBs, however, the assumptions made by the *IA* are unlikely to hold. It is unlikely, for example, that the set of properties that hold for each entity will have been helpfully precomputed. As such, an REG algorithm operating under uncertain and distributed knowledge cannot rely on simple set-membership checks, but must instead explicitly check how probable it is that an entity has a particular property.

In this section, we present *DIST-PIA*, the Distributed, Probabilistic Incremental Algorithm which uses the aforementioned consultant framework to do just that. And we provide a walkthrough of this algorithm in an example scenario. Each step of this walkthrough is summarized in a row of Tab. 1 and denoted in the walkthrough in bold (e.g., **(1)**).

To illustrate the behavior of *DIST-PIA*, we will use an architectural configuration with three distributed consultants C for representing people (p), locations (l), and objects (o). If *DIST-PIA* (*DP* hereafter) is required, using this architecture, to refer to entity $m = p_5$, it will begin by creating an empty de-

[1] A preliminary description of this algorithm also appears in (Williams and Scheutz, 2017a).

Notation

C A set of *consultants* $\{c_0, \ldots, c_n\}$

c_m^Λ The set of formulae $\{\lambda_0, \ldots, \lambda_n\}$ advertised by the consultant $c \in C$ responsible for m.

M A robot's *world model* of entities $\{m_0 \ldots m_n\}$ found in the domains provided by C.

D The incrementally built up description, comprised of mappings from entities M to sets of pairs (λ, Γ) of formulae and bindings for those formulae.

D^M The set of entities $m \in M$ for which sub-descriptions have been created.

d^M The set of entities $m \in M$ involved in sub-description d.

P The set of candidate (λ, Γ) pairs under consideration for addition to a sub-description.

Q The queue of referents which must be described.

X The incrementally pruned set of distractors

Algorithm 1 *DIST-PIA*(m, C)

1: $D =$ new Map() // *The Description*
2: $Q =$ new Queue(m) // *The Referent Queue*
3: **while** $Q \neq \emptyset$ **do**
4: // *Consider the next referent*
5: $m\prime$ = pop(Q)
6: // *Craft a description d for it*
7: $d =$ *DIST-PIA-HELPER*$(m\prime, C)$
8: $D = D \cup \{m \rightarrow d\}$
9: // *Find all entities used in d*
10: **for all** $m\prime\prime \in d^M \setminus keys(D)$ **do**
11: // *And add undescribed entities to the queue*
12: $push(Q, m\prime\prime)$
13: **end for**
14: **end while**
15: **return** D

Algorithm 2 *DIST-PIA-HELPER*(m, C)

1: $d = \emptyset$ // *The Sub-Description*
2: $X = M \setminus m$ // *The Distractors*
3: // *Initialize a set of properties to consider: those advertised by the consultant c responsible for* m
4: $P = [\forall \lambda \in c_m^\Lambda : (\lambda, \emptyset)]$
5: // *While there are distractors to eliminate or properties to consider*
6: **while** $X \neq \emptyset$ and $P \neq \emptyset$ **do**
7: $(\lambda, \Gamma) = pop(P)$
8: // *Find all unbound variables in the next property*
9: $V = find\_unbound(\lambda, \Gamma)$
10: **if** $|V| > 1$ **then**
11: // *If there's more than one, create copies of that property under all possible variable bindings that leaving unbound exactly one variable of the same type as the target referent*
12: **for all** $\Gamma\prime \in cross\_bindings(\lambda, \Gamma, C)$ **do**
13: // *And push them onto the property list*
14: $push(P, (\lambda, \Gamma\prime))$
15: **end for**
16: // *Otherwise, if it is sufficiently probable that the property applies to the target referent...*
17: **else if** $apply(c_m, \lambda, \Gamma \cup (v_0 \rightarrow m)) > \tau_{dph}$ **then**
18: // *And it's sufficiently probable that it does* ***not*** *apply to at least one distractor...*
19: $\bar{X} = [x \in X \mid apply(c_x, \lambda, \Gamma \cup (v_0 \rightarrow x)) > \tau_{dph}]$
20: // *Then bind its free variable to the target referent, and add it to the sub-description...*
21: **if** $\bar{X} \neq \emptyset$ **then**
22: // *And remove any eliminated distractors*
23: $d = d \cup (\lambda, \Gamma \cup (v_0 \rightarrow m))$
24: $X = X \setminus \bar{X}$
25: **end if**
26: **end if**
27: **end while**
28: **return** d

scription $D = \emptyset$ and a queue of referents to describe $Q = \{p_5\}$ (Tab. 1 Row **(1)**; Algorithm 1, Lines 1-2). Because there are still referents left to describe (Line 5), *DP* calls on its helper function *DIST-PIA-HELPER* (*DPH* hereafter) to craft a sub-description for p_5, which is popped off of Q (Line 7).

DPH begins by asking the consultant responsible for p_5 for a set of distractors X (e.g., $\{p_1, p_2, p_3, p_4\}$) and a set of properties P to consider (e.g., c_m^Λ = jim(X-p), jill(X-p), man(X-p), woman(X-p), lives-in(X-p,Y-l)), each of which *DPH* pairs with an empty set of bindings (Algorithm 2, Lines 1- 4). From this list, *DPH* pops the first unconsidered property (i.e., $jim(X - p)$) and its (empty) set of bindings. $jim(X - p)$ has exactly one unbound variable, so *DPH* will use **(2)** consultant p's $apply$ method (as per Capability 3) to ask how probable it is that $jim(X - p)$ applies to p_5. Suppose the returned probability is above threshold τ_{dph} (e.g., 60%). Because the chosen property does indeed apply to the target referent, *DPH* uses the same method to determine whether it also applies to any distractor x in X. Suppose this is only the case for p_2. The remaining distractors $\{p_1, p_3, p_4\}$ **(3)** will be removed from X and **(4)** $jim(p_5)$ will be added to sub-description d (Lines 17- 26).

DPH will then repeat this process with other properties. Suppose it is insufficiently probable that $jill(X - p)$ holds **(5)**: it will be ignored. Suppose it *is* sufficiently probable that $man(X - p)$ holds **(6)**, but that it is *also* sufficiently probable that it applies to the lone remaining distractor, p_2 **(7)**: it will

#	Act	Description	m	Sub-description	Distractors	Property	Property List
1	P	$\emptyset$	p_5	$\emptyset$	$\{p_1, p_2, p_3, p_4\}$	$\emptyset$	$\{jim(X\text{-}p), jill(X\text{-}p), man(X\text{-}p), wom(X\text{-}p), l\text{-}in(X\text{-}p, Y\text{-}l)\}$
2	A	$\emptyset$	p_5	$\emptyset$	$\{p_1, p_2, p_3, p_4\}$	$jim(X\text{-}p)$	$\{jill(X\text{-}p), man(X\text{-}p), wom(X\text{-}p), l\text{-}in(X\text{-}p, Y\text{-}l)\}$
3	E	$\emptyset$	p_5	$\emptyset$	$\{p_2\}$	$jim(X\text{-}p)$	$\{jill(X\text{-}p), man(X\text{-}p), wom(X\text{-}p), l\text{-}in(X\text{-}p, Y\text{-}l)\}$
4	d	$\emptyset$	p_5	$\{jim(p_5)\}$	$\{p_2\}$	$\emptyset$	$\{jill(X\text{-}p), man(X\text{-}p), wom(X\text{-}p), l\text{-}in(X\text{-}p, Y\text{-}l)\}$
5	A	$\emptyset$	p_5	$\{jim(p_5)\}$	$\{p_2\}$	$jill(X\text{-}p)$	$\{man(X\text{-}p), wom(X\text{-}p), l\text{-}in(X\text{-}p, Y\text{-}l)\}$
6	A	$\emptyset$	p_5	$\{jim(p_5)\}$	$\{p_2\}$	$man(X\text{-}p)$	$\{wom(X\text{-}p), l\text{-}in(X\text{-}p, Y\text{-}l)\}$
7	E	$\emptyset$	p_5	$\{jim(p_5)\}$	$\{p_2\}$	$man(X\text{-}p)$	$\{wom(X\text{-}p), l\text{-}in(X\text{-}p, Y\text{-}l)\}$
8	A	$\emptyset$	p_5	$\{jim(p_5)\}$	$\{p_2\}$	$wom(X\text{-}p)$	$\{l\text{-}in(X\text{-}p, Y\text{-}l)\}$
9	B	$\emptyset$	p_5	$\{jim(p_5)\}$	$\{p_2\}$	$l\text{-}in(X\text{-}p, Y\text{-}l)$	$\{l\text{-}in(X\text{-}p, l_1), l\text{-}in(X\text{-}p, l_2), l\text{-}in(X\text{-}p, l_3)\}$
10	A	$\emptyset$	p_5	$\{jim(p_5)\}$	$\{p_2\}$	$l\text{-}in(X\text{-}p, l_1)$	$\{l\text{-}in(X\text{-}p, l_2), l\text{-}in(X\text{-}p, l_3)\}$
11	E	$\emptyset$	p_5	$\{jim(p_5)\}$	$\emptyset$	$l\text{-}in(X\text{-}p, l_1)$	$\{l\text{-}in(X\text{-}p, l_2), l\text{-}in(X\text{-}p, l_3)\}$
12	d	$\emptyset$	p_5	$\{jim(p_5), l\text{-}in(p_5, l_1)\}$	$\emptyset$	$\emptyset$	$\{l\text{-}in(X\text{-}p, l_2), l\text{-}in(X\text{-}p, l_3)\}$
13	D	$\{jim(p_5), l\text{-}in(p_5, l_1)\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
14	P	$\{jim(p_5), l\text{-}in(p_5, l_1)\}$	l_1	$\emptyset$	$\{l_2, l_3\}$	$\emptyset$	$\{som(X\text{-}l), cam(X\text{-}l), mass(X\text{-}l), in(X\text{-}l, Y\text{-}l)\}$
15	A	$\{jim(p_5), l\text{-}in(p_5, l_1)\}$	l_1	$\emptyset$	$\{l_2, l_3\}$	$som(X\text{-}l)$	$\{cam(X\text{-}l), mass(X\text{-}l), in(X\text{-}l, Y\text{-}l)\}$
16	E	$\{jim(p_5), l\text{-}in(p_5, l_1)\}$	l_1	$\emptyset$	$\emptyset$	$som(X\text{-}l)$	$\{cam(X\text{-}l), mass(X\text{-}l), in(X\text{-}l, Y\text{-}l)\}$
17	d	$\{jim(p_5), l\text{-}in(p_5, l_1)\}$	l_1	$\{som(l_1)\}$	$\emptyset$	$\emptyset$	$\{cam(X\text{-}l), mass(X\text{-}l), in(X\text{-}l, Y\text{-}l)\}$
18	D	$\{jim(p_5), l\text{-}in(p_5, l_1), som(l_1)\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

Table 1: WALKTHROUGH SUMMARY.
Column Two summarizes action taken: **P**repare, **A**ssess, **E**liminate, **B**ind, **d**-append, or **D**-append.
Some predicates are abbreviated, and predicate/binding tuples are rewritten as bound predicates.

be ignored. Suppose it is insufficiently probable that $woman(X-p)$ holds **(8)**: it will be ignored. Finally, *DPH* will consider $lives\text{-}in(X-p, Y-l)$. Unlike the previous properties, this has two unbound variables. *DPH* will thus use **(9)** $cross\_bindings$ to create a set of candidate variable bindings for this property, each of which leaves exactly one variable for which p is responsible unbound. Suppose l knows of locations l_1, l_2, and l_3: $cross\_bindings$ will return $(lives\text{-}in(X-p, Y-l), \{Y \rightarrow ls_1\})$, $(lives\text{-}in(X-p, Y-l), \{Y \rightarrow ls_2\})$, and $(lives\text{-}in(X-p, Y-l), \{Y \rightarrow ls_3\})$, each of which will be added onto P for *DPH* to consider. (Lines 10- 15).

Suppose it is sufficiently probable **(10)** that $(lives\text{-}in(X-p, l_1))$ applies to p_5 but not to the lone remaining distractor (p_2), allowing p_2 to be ruled out **(11)**. $lives\text{-}in(X-p, Y-l)$ will be added **(12)** to d and $\{p_2\}$ will be removed from X. Since X is now empty, the sub-description $p_5 \rightarrow \{jim(p_5), lives\text{-}in(p_5, l_1)\}$ will be returned **(13)** to *DP* (Line 28). Notice that this sub-description refers to an entity (l_1) which itself needs to be described. Accordingly, l_1 will be added to Q **(14)**, and because it is the only entity on the queue, immediately popped and sent back to DPH.

As before, *DPH* begins by asking the consultant responsible for l_1 for a set of distractors X (e.g., $\{p_2, l_3\}$) and a set of properties P to consider (e.g., c_m^Λ = somerville(X-l), cambridge(X-l), massachusetts(X-l), in(X-l,Y-l)), which it considers one at a time. Suppose it is sufficiently probable that $somerville(X-l)$ applies to l_1 **(15)** but not to distractors l_2 or l_3 **(16)**: it will be added **(17)** to d, $\{l_2, l_3\}$ will be removed from X, and $l_1 \rightarrow \{somerville(l_1)\}$ will be returned **(18)** to *DP*. Finally, since Q is empty, *DP* will return $\{p_5 \rightarrow \{jim(p_5), lives\text{-}in(p_5, l_1)\}, l_1 \rightarrow \{somerville(l_1)\}\}$ (Algorithm 1, Line 15). It will be the responsibility of the next component of the natural language pipeline to translate this into an RE along the lines of "Jim, who lives in Somerville"[2].

[2]The integration of *DIST-PIA* with the remaining natural language components of our robot architecture is described in our other recent work (Williams and Scheutz, 2017b).

5 Evaluation

Traditional REG evaluation metrics (e.g., *Dice* (Gatt et al., 2007) and *MASI* (Passonneau, 2006)) compare algorithm- and human-chosen attributes by measuring the distance (e.g., set difference) between machine- and human-generated attribute sets. Recently, however, this methodology has come under criticism, as the semantic similarity of two attribute sets does not imply similarity between the *effectiveness* of those two sets. That is, this methodology does not necessarily assess how well a generated RE actually allows a target to be picked out by a hearer – the presumed purpose of REG algorithms (Van Deemter and Gatt, 2009). Recently, there has been a shift towards *task-based* evaluations (e.g., (Byron et al., 2009; Koller et al., 2010; Viethen and Dale, 2006)), in which algorithms are compared by how well they allow some task to be achieved.

The previously discussed uncertainty-handling REG algorithms have mainly used task-based evaluations in which an image provided to participants is also provided to the algorithm. This necessarily conflates the evaluation of the algorithm with the evaluation of the visual classifiers used to process that image. Furthermore, it prevents direct comparison between the algorithm and both other algorithms (unless they use identical classifiers) and humans. It is thus imperative to develop a *new* evaluation framework that allows an REG algorithm to receive information about attribute uncertainty without visually processing the scene.

We will now present an evaluation framework that achieves this goal through two stages. In Stage One, participants are shown an environment, and are asked to provide (1) an RE referring to an indicated entity, and (2) probability judgments that particular attributes hold for indicated entities. The probability judgments can be used to train REG algorithms to assess whether various attributes hold without committing to particular classifiers. In Stage Two, new participants are shown the same environments, along with either human- or machine-generated REs, and asked to indicate the described entity. This framework thus allows REG algorithms to be compared to both other algorithms and humans under uncertainty.

5.1 Stage One

In the first stage of the evaluation, participants were each shown three randomly-ordered scenes (a kitchen, an office, and a near-featureless room, as seen in Fig. 1)[3], one of which contained a red bounding box around an object. This object was either an object that only appeared in that scene, an object for which an identical object appeared in a different scene, or an object for which one of the same type (but, for example, of a different color) appeared in a different scene, yielding five task-relevant objects in each scene. Each image also contained around five salient irrelevant objects. Finally, because each participant was simultaneously shown three scenes, the rooms themselves also serve as anchors with respect to which referents could be described.

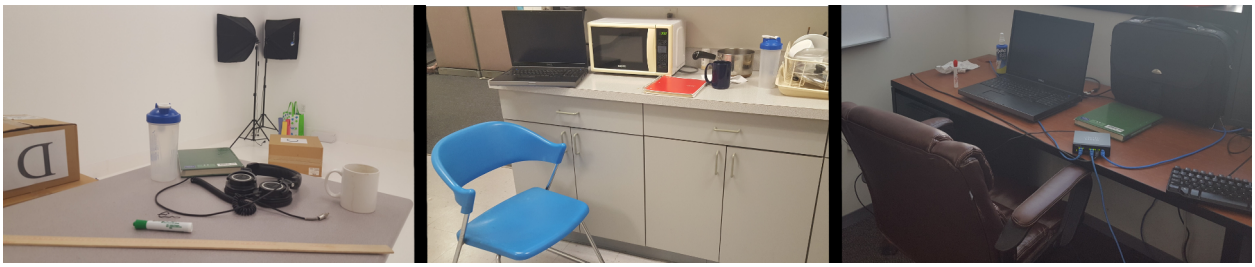
Participants were told to imagine that a future participant would tour the three rooms shown in the pictures in a random order, and in one of the rooms, receive a description of an object. Participants were told to write the description that the future participant should receive. After each participant provided a description, they were asked to evaluate how well the target object matched each of twenty attributes (randomly selected from a total of 52 informally collected attributes such as "is blue", "is a marker", and "is in the kitchen" by re-positioning [0-100] sliders from an initial position of 50.

Participants (56 male, 33 female; mean age 35 (sd=11.65)) were recruited through Amazon Mechanical Turk and paid small cash. Each participant was shown a random target object, providing us with an average of 10 REs per target object and 3.8 probability judgments for each attribute for each object. Gold standard semantic parses (i.e., sets of logical formulae representing properties and relations) were then crafted for each human-generated RE.

Next, two consultants were created which were provided with, respectively, a subset of the objects and locations found in the three scenes. When these consultants are asked for the probability that an entity has a particular attribute, they return the mean probability judgment provided by participants. Finally, each consultant was provided with a preference ordering over properties. While this ordering

[3] In future work it would of course be valuable to perform a more comprehensive evaluation with a larger variety of scenes.

Figure 1: Scenes Shown to Participants



In the scene to the left, the possible target referents in the two evaluation stages were the waterbottle, headphones, and mug; in the middle scene, these were the laptop, chair, and notebook; in the right scene, these were the briefcase, book, and marker.

was hand constructed, we would eventually like to learn similar orderings from data.

DIST-PIA was then used to generate attribute sets for each of the nine target referents, as shown in Tab. 2. We then combined these with the attribute sets derived from human utterances, yielding an average of 9.56 (sd=3.13) unique sets of attributes per target object. For each attribute set, we crafted one RE using a predefined template. This conversion from REs to logical form and back allows us to control for consistent phrasing. Because one RE was produced for each unique property set, an average of 9.56 (sd=3.13) REs were created per target object.

5.2 Stage Two

In the second stage, a new set of participants were shown the same images, but without bounding boxes, shown a randomly selected human- or machine-driven RE for that referent, and asked to click on the described object. After each image, participants were notified as to whether they had clicked on the correct object.

Participants were recruited through Amazon Mechanical Turk (62 male, 46 female; mean age 35.07 (sd=10.14)) and paid small cash. Each of the 85 unique REs was thus shown to an average of 11.44 participants. Recall, however, that these utterances were crafted based on property sets either chosen by *DIST-PIA* or extracted from the utterances collected from participants in Stage One. Because some of these property sets were identical, each of the unique REs in this section really corresponds to a *cluster* of human- or machine-driven property sets. We thus computed how accurately each property set allowed the true target referent to be picked out. Ranking clusters by accuracy, we can then compute an overall *accuracy percentile* for *DIST-PIA*, i.e., the percent of RE generators (in this case, humans) compared to which *DIST-PIA* achieved higher accuracy.

5.3 Results and Discussion

Overall, *DIST-PIA* allowed successful identification in 91.4% of cases. On average, the REs crafted using *DIST-PIA*-chosen properties were as or more successful than those crafted using the properties chosen by 45.7% (sd=23.9) of human participants, suggesting that *DIST-PIA* was nearly as effective as humans in choosing properties. *DIST-PIA's* performance would further improve if given more sophisticated consultants. To fairly evaluate *DIST-PIA*, we provided it with consultants that only made judgments based on the attributes used in our pilot study. Even simple knowledge of what relations were *symmetric* would have increased performance.

It is important to note that because *DIST-PIA* is domain independent, it does not *compete* with the classifiers used by previous approaches, which could be integrated into our framework as consultants, allowing for direct comparison of disparate classifiers. And, while our evaluation *presented* all information visually (as we needed participants to be able to unambiguously select referents in an intuitive, static, online environment) it is critical to point out that our evaluation avoided *reliance* on specific visual classifiers: the generality of our consultant framework means that, like the *IA* (but unlike previous REG algorithms which have handled uncertainty), we could easily use consultants that assess non-visual traits (e.g., object costs, utterance sentiments, people professions, room temperatures).

Finally, *DIST-PIA* did not "nicely overspecify" in some conditions where humans did. For example,

Table 2: Properties Chosen by *DIST-PIA*

ID	Properties	Translation	Acc.	Rank
1	notebook(X)	The notebook	80%	11 of 11
2	Dell(X),laptop(X),blue(Y), chair(Y),in-front-of(Y,X)	The Dell laptop that the blue chair is in front of.	85.7%	5* of 10
3	blue(X),chair(X)	The blue chair.	97.8%	2 of 3
4	laptop-bag(X)	The laptop-bag.	100%	1 of 5
5	textbook(X),laptop-bag(Y), behind(Y,X)	The textbook that the laptop-bag is behind.	63.6%	5 of 10
6	red(X),whiteboard-marker(X)	The red whiteboard-marker.	86.7%	6 of 10
7	headphones(X)	The headphones.	100%	1* of 13
8	shaker-bottle(X), headphones(Y),next-to(X,Y)	The shaker-bottle next to the headphones.	85.7%	3* of 13
9	coffee-mug(X)	The coffee-mug.	100%	1* of 10

"Accuracy" denotes percent of participants who chose the correct object when given the machine-driven RE. "Rank" compares this with human-driven REs. * denotes a tie. For example, when "The notebook" was used, 80% of participants chose the correct object, but all other REs for that object yielded higher accuracy. In contrast, "The textbook that the laptop-bag is behind" had only 63.6% accuracy, but this was a higher than was achieved by all but four of the unique human-driven REs for that object.

DIST-PIA's choice of simply $notebook(X)$ for the first target object achieved 80% success rate, but had the lowest ranking for that object, in part because most humans used descriptions involving $red(X)$ to draw the eye away from distractors like the green book. The traditional *IA* captures this effect by placing colors at a high priority. However, unlike the traditional *IA*, we chose to handle the object's "type" (e.g., "bottle") and variants thereof (e.g., "waterbottle") just like any other properties, so that we would not need to specify an additional mandatory consultant capability (i.e., the ability to provide the "type" of a candidate object). This required us to place these type-like properties at the top of the preference orderings, to make sure that *type* was always used. When presented the trade-off, we chose generality of architectural mechanisms over performance gain.

6 Conclusion

In this paper, we make three main contributions. First, we presented a domain independent algorithm for REG under uncertainty, which separates the problems of referring expression generation and reference resolution from the task of *property assessment*. Second, we presented a novel evaluation framework which allows REG algorithms designed for uncertain contexts to be evaluated without conflating the performance of the *algorithm* with the performance of the *classifiers used by the algorithm*, and which uses human probability judgments to better facilitate comparison to human performance. Using this framework, we showed that *DIST-PIA* exhibited REG capabilities comparable to those of humans. Finally, we have taken the realities of modern integrated agent architectures into account through our *consultant framework*, which allows information to be distributed across multiple heterogeneous KBs (Williams and Scheutz, 2016).

In future work, our thresholds should be learned from data, and Dempster-Shafer Theoretic uncertainty representations should be used to better handle ignorance (Williams et al., 2015). *DIST-PIA* should also be modified to incorporate audience design considerations, similar to Horacek (2005). Finally, *DIST-PIA* should be modified to use use Givenness-Hierarchy Theoretic mechanisms (Williams et al., 2016) in conjunction with a multi-modal reference model to generate deictic and anaphoric REs.

Acknowledgments

This research was in part funded by grant N00014-14-1-0149 from the US Office of Naval Research. We would also like to thank Lars Kunze for constructive conversations which contributed to our evaluation framework.

References

Donna Byron, Alexander Koller, Kristina Striegnitz, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. 2009. Report on the first NLG challenge on generating instructions in virtual environments (GIVE). In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG)*, pages 165–173. Association for Computational Linguistics.

Robert Dale and Ehud Reiter. 1995. Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.

Robert Dale. 1989. Cooking up referring expressions. In *Proceedings of the 27th annual meeting on Association for Computational Linguistics (ACL)*, pages 68–75.

Rui Fang, Changsong Liu, Lanbo She, and Joyce Y Chai. 2013. Towards situated dialogue: Revisiting referring expression generation. In *Proceedings of Empirical Methods on Natural Language Processing (EMNLP)*, pages 392–402.

Rui Fang, Malcolm Doering, and Joyce Y Chai. 2015. Embodied Collaborative Referring Expression Generation in Situated Human-Robot Interaction. In *Proceedings of the 10th International Conference on Human-Robot Interaction (HRI)*, pages 271–278.

Albert Gatt, Ielka van der Sluis, and Kees van Deemter. 2007. Evaluating algorithms for the generation of referring expressions using a balanced corpus. In *Proceedings of the Eleventh European Workshop on Natural Language Generation (ENLG)*, pages 49–56, Stroudsburg, PA, USA. Association for Computational Linguistics.

Helmut Horacek. 2005. Generating referential descriptions under conditions of uncertainty. In *Proceedings of the 10th European Workshop on Natural Language Generation (ENLG)*, pages 58–67.

Alexander Koller, Kristina Striegnitz, Andrew Gargett, Donna Byron, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. 2010. Report on the second NLG challenge on generating instructions in virtual environments (GIVE-2). In *Proceedings of the 6th International Conference on Natural Language Generation (INLG)*, pages 243–250. Association for Computational Linguistics.

Sébastien Konieczny. 2000. On the difference between merging knowledge bases and combining them. In *Proceedings of Knowledge Representation (KR)*, pages 135–144.

Emiel Krahmer and Kees Van Deemter. 2012. Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1):173–218.

Paolo Liberatore and Marco Schaerf. 1998. Arbitration (or how to merge knowledge bases). *IEEE Transactions on Knowledge and Data Engineering*, 10(1):76–90.

Jinxin Lin. 1996. Integration of weighted knowledge bases. *Artificial Intelligence*, 83(2):363–378.

Timothy Meo, Brian McMahan, and Matthew Stone. 2014. Generating and resolving vague color references. *Proceedings of The 18th Workshop on the Semantics and Pragmatics of Dialogue (SemDial)*, pages 107–115.

Rebecca Passonneau. 2006. Measuring agreement on set-valued items (MASI) for semantic and pragmatic annotation. In *In Proceedings of the International Conference on Language Resources and Evaluation (LREC)*.

Morgan Quigley, Josh Faust, Tully Foote, and Jeremy Leibs. 2009. ROS: an open-source robot operating system. In *ICRA Workshop on Open Source Software*.

Deb K Roy. 2002. Learning visually grounded words and syntax for a scene description task. *Computer Speech & Language*, 16(3):353–385.

Amir Sadovnik, Andrew Gallagher, and Tsuhan Chen. 2013. Not everybody's special: Using neighbors in referring expressions with uncertain attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*, pages 269–276.

Matthias Scheutz, Gordon Briggs, Rehj Cantrell, Evan Krause, Tom Williams, and Richard Veale. 2013. Novel mechanisms for natural human-robot interactions in the DIARC architecture. In *Proceedings of the AAAI Workshop on Intelligent Robotic Systems*.

Kees Van Deemter and Albert Gatt. 2009. Beyond DICE: measuring the quality of a referring expression. In *n Proceedings of the Workshop on Production of Referring Expressions: Bridging Computational and Psycholinguistic Approaches (preCogSci-09)*.

Kees Van Deemter. 2016. *Computational Models of Referring: A Study in Cognitive Science*. MIT Press.

Jette Viethen and Robert Dale. 2006. Towards the evaluation of referring expression generation. In *Proceedings of the 4th Australasian Language Technology Workshop*, pages 115–122, Sydney, Australia.

Tom Williams and Matthias Scheutz. 2016. A framework for resolving open-world referential expressions in distributed heterogeneous knowledge bases. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, pages 3958–3964.

Tom Williams and Matthias Scheutz. 2017a. Referring expression generation under uncertainty in integrated robot architectures. In *Proceedings of Robotics: Science and Systems Workshop on Human-Centered*

Robotics: Interaction, Physiological Integration and Autonomy.

Tom Williams and Matthias Scheutz. 2017b. Resolution of referential ambiguity in human-robot dialogue using dempster-shafer theoretic pragmatics. In *Proceedings of Robotics: Science and Systems (RSS)*.

Tom Williams, Gordon Briggs, Bradley Oosterveld, and Matthias Scheutz. 2015. Going beyond command-based instructions: Extending robotic natural language interaction capabilities. In *Proceedings of 29th AAAI Conference on Artificial Intelligence*, pages 1387–1393.

Tom Williams, Saurav Acharya, Stephanie Schreitter, and Matthias Scheutz. 2016. Situated open world reference resolution for human-robot dialogue. In *Proceedings of the 11th ACM/IEEE International Conference on Human-Robot Interaction*, pages 311–318.

Sina Zarrieß and David Schlangen. 2016. Towards generating colour terms for referents in photographs: Prefer the expected or the unexpected? In *Proceedings of the 9th International Natural Language Generation conference (INLG)*.