Generating Expressions that Refer to Visible Objects

Margaret Mitchell
Johns Hopkins HLTCOE
m.mitchell@jhu.edu

Kees van Deemter
University of Aberdeen
k.vdeemter@abdn.ac.uk

Ehud Reiter
University of Aberdeen
e.reiter@abdn.ac.uk

Abstract

We introduce a novel algorithm for generating referring expressions, informed by human and computer vision and designed to refer to visible objects. Our method separates absolute properties like color from relative properties like size to stochastically generate a diverse set of outputs. Expressions generated using this method are often overspecified and may be underspecified, akin to expressions produced by people. We call such expressions *identifying descriptions*. The algorithm outperforms the well-known Incremental Algorithm (Dale and Reiter, 1995) and the Graph-Based Algorithm (Krahmer et al., 2003; Viethen et al., 2008) across a variety of images in two domains. We additionally motivate an evaluation method for referring expression generation that takes the proposed algorithm's non-determinism into account.

1 Introduction

Referring expression generation (REG) is the task of generating an expression that can identify a referent to a listener. These expressions generally take the form of a definite noun phrase such as "the large orange plate" or "the furry running dog". Research in REG primarily focuses on the subtask of selecting a set of properties that may be used to construct the final surface expression, e.g., ⟨color:orange, size:large, type:plate⟩. This property selection task is optimized to meet different goals: for example, to be identical to those a person would generate in the same situation, or to be unique to the intended referent and no other item in the discourse.

We focus on the task of generating referring expressions for visible objects, specifically with the goal of generating descriptive, human-like referring expressions. We are motivated by the desire to connect this algorithm to input from a computer vision system, and discuss how this may work throughout the paper. Computer vision (CV) does not yet reliably provide features for some of the most frequent properties that people use in visual description (in particular, size-based features), and so we use a gold-standard visual input, evaluating purely on REG. The proposed algorithm, which we call the Visible Objects Algorithm, is designed to approximate human variation identifying an object in a group of visible, real world objects.

Our primary contributions are the following. Background for each issue is provided in Section 2:

1. An approach accounting for overspecification, underspecification, and some of the known effects of vision on reference.
2. A function to approximate the stochastic nature of reference. This reflects that people will produce different references to the same object.
3. A separation between absolute properties like color, which may be detected directly by CV, from relative properties like size and location, which require reasoning over visual features to determine an appropriate form (e.g., height/width and distance features between pixels are available from a visual input; saying an object is "tall" requires further reasoning).
4. An evaluation method for non-deterministic REG that aligns generated and observed data and calculates accuracy over alignments.

2 Motivation & Overview

Most implemented algorithms for referring expression generation focus on unique identification of a referent, determining the set of properties that distinguish a particular target object from the other objects in the scene (the *contrast set*) (Dale, 1989; Reiter and Dale, 1992; Dale and Reiter, 1995; Krahmer et al., 2003; Areces et al., 2008). This view of reference was first outlined by Olson (1970), "the specification of an intended referent relative to a set of alternatives". A substantial body of evidence now shows that contrastive value relative to alternatives is not the only factor motivating speakers' property choices, specifically in visual domains. The phenomena of *overspecification* and *redundancy*, where speakers select properties that have little or no contrastive value, was observed in early developmental studies in visual domains (Ford and Olson, 1975; Whitehurst, 1976; Sonnenschein, 1985) as well as later studies on adult speakers in visual domains (Pechmann, 1989; Engelhardt et al., 2006; Koolen et al., 2011). The related phenomenon of *underspecification*, where speakers select a set of properties that do not linguistically specify the referent, has also received some attention, particularly in visual domains (Clark et al., 1983; Kelleher et al., 2005).

These findings make sense in light of visual evidence that some properties "pop out" in the scene (Treisman and Gelade, 1980), and speakers may begin referring before scanning the full set of scene objects (Pechmann, 1989), selecting those properties that are salient for them (Horton and Keysar, 1996; Bard et al., 2009) without spending a great amount of cognitive effort considering the perception of a hearer (Keysar and Henly, 2002).

We take this evidence to suggest an approach for a visual reference algorithm that generates natural, human-like reference by generating visual properties that are salient for a speaker.[1] We can understand what is salient visually (what does the visual system first respond to, what guides attention?), linguistically (what do people tend to mention in visual scenes?), and cognitively, which we will not have room to discuss in this paper (what is atypical for this object?); as well as in terms of broader notions of salience, e.g., discourse salience (Krahmer and Theune, 2002).

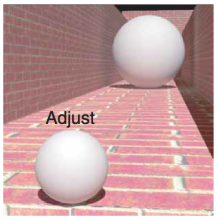

Figure 1: Relative properties, like size and location, are difficult to obtain from a two-dimensional image. We find it easy to perceive the background object as larger than the one in the front; but they are technically the same size in the image (from Murray et al. (2006)).

This suggests a paradigm shift in the generation task when referring to visible objects, if the goal is to produce human-like reference. In particular, this suggests moving from selecting properties that *rule out* other scene objects to selecting properties that are salient for the speaker (visually, conversationally, based on previous experiences, etc.). This mirrors related research on the tradeoff between audience design and egocentrism in language production (Clark and Murphy, 1982; Horton and Keysar, 1996; Bard et al., 2009; Gann and Barr, 2013). Under- and overspecification naturally fall out from such an approach, with no need to specifically model either phenomenon.

Perhaps unsurprisingly, the set of properties that are visually salient and the set of properties that are linguistically salient largely overlap. Color is the first property our visual system processes, followed soon after by size (Murray et al., 2006; Fang et al., 2008; Schwarzkopf et al., 2010); and people tend to use color (Pechmann, 1989; Viethen et al., 2012) and size when identifying objects, with size common when there is another object of the same type in the scene (Brown-Schmidt and Tanenhaus, 2006).

Following this, our algorithm gives a privileged position to these properties, processing them first. Using computer vision techniques to determine an object's color works reasonably well (Berg et al., 2011), and the relevant visual features for this task may be useful in future work to return several possible color labels that capture differences in lexical choice (cf. Reiter and Sripada (2002)).

Detecting size does not work well (Forsyth,

[1] We can also add functionality to ensure that a referent is uniquely identified against the contrast set (whether or not that reflects what a person would do), which we will describe.

2011); and when it does, it will likely not take the form supposed in recent generation work. Most REG algorithms use a predefined single-featured value, such as "big"; however, given an image-based input, obtaining such a value requires (1) determining how the object is situated in a three-dimensional space, difficult to obtain from a two-dimensional image (see Figure 1); and (2) determining what the value should be: object detectors currently can provide the height and width of the location where an object is likely to exist (its bounding box), as well as the x- and y-axis locations of the pixels within the object detection; but a value from these features like "big", "tall", or "long" requires further reasoning. As such, we incorporate the top-performing size algorithm introduced in Mitchell et al. (2011), which takes as input the height and widths of objects in the image and outputs a size value or NONE, indicating that size should not be used to describe the object.

In addition to color and size, location and orientation begin to be processed early on in the visual system (Treisman, 1985; Itti and Koch, 2001), with our first perception of location corresponding to basic cues of where an object is relative to our focus of attention. For an input image, this simple type of location corresponds to surface forms such as, e.g., "on the right of the image" or "at the top of the image". Along with size, location and orientation make up the three primary relative properties that we aim to generate language for.

After the simple forms for color, size, location, and orientation properties are processed, our visual system feeds forward to two parallel pathways, the so-called "what" and "where" pathways (Ungerleider and Mishkin, 1982), which process properties with growing complexity. The "what" pathway includes absolute properties like shape and material, which computer vision has had some success detecting (Ferrari and Zisserman, 2007; Farhadi et al., 2009) while the "where" pathway corresponds to more complex spatial orientation and location information, such as where objects are relative to one another and which way they are facing.

To begin connecting this process to the generation of human-like descriptions of visible objects, we start with the following simplification: Color and size have a privileged status, the first properties processed. These are followed by the relative properties

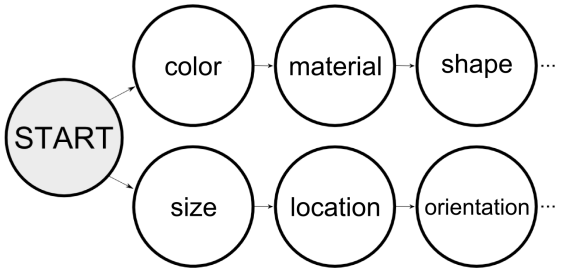

Figure 2: Initial model for generating visual reference.

of location and orientation, which may feed forward to more complex location and orientation properties in one pathway; and absolute properties following color, like material and shape, which may be processed in another pathway.

This gives us the basic model for generating reference to visible objects shown in Figure 2. To generate reference in this model, nodes correspond to general visual *attributes* and may generate forms for visual *properties* (attribute:value pairs). That is, a property such as color:red is generated from the attribute node color and a property such as size:tall is generated from the attribute node size. We are limited by existing REG corpora in which properties we can evaluate; in this paper, we examine the effect of the independent selection of color and size, followed by location and orientation.[2]

Generating human-like expressions in this setting begins to be possible by adopting recent proposals that REG handle speaker variation (Viethen and Dale, 2010) and the non-deterministic nature of reference (van Gompel et al., 2012; van Deemter et al., 2012b). We can capture such variation simply by estimating α_{att}, the likelihood that an attribute att generates a corresponding visual property. During generation, the algorithm passes through each attribute node, and uses this estimate to stochastically add each property to the output property set.

Such a non-deterministic process means that the algorithm will not return the same output every time, which offers new challenges for evaluation. If we run the algorithm 1,000 times, we have a distribution over several possible output property sets. From this we can obtain the majority set and check if it matches the majority observed set. Similarly, we can

[2]We have also built an algorithm and corpus with more complex properties in order to tease out further details of visual reference, but must leave these details for follow up work; for now, we focus on the properties common to REG corpora.

run the algorithm for as many instances as we have in our test data, and see how well the property sets it produces aligns to the observed property sets. We discuss evaluation using both methods in Section 6.

3 The State of the Art in REG

3.1 Algorithms

In order to understand how this approach compares to the state of the art in REG, we evaluate against two of the most well-known algorithms, the Incremental Algorithm (Dale and Reiter, 1995) and the Graph-Based Algorithm (Krahmer et. al, 2003, as implemented in Viethen et al., 2008). Details on these algorithms are available in their corresponding papers. As a brief summary, both algorithms formalize the objects in the discourse as a set of properties (attribute:value pairs). For example, one object may be represented as ⟨type:box, color:red, size:large⟩. The task is to find the set of properties that uniquely specify the referent. This is known as a content selection problem, and the set of properties chosen by the algorithm is called a *distinguishing description*.

The Incremental Algorithm (IA) proceeds by iterating through attributes in a predefined order (a preference order), and for each attribute, it checks whether specifying a value would rule out at least one item in the contrast set that has not already been ruled out. If it will, the attribute:value is added to the distinguishing description. This process continues until all contrast items (distractors) are ruled out or all available properties have been checked. We use the implementation of the IA available from the NLTK (Bird et al., 2009).[3]

In the Graph-Based Algorithm (GB), the objects in the discourse are represented within a labeled directed graph, and content selection is a subgraph construction problem. Each object is represented as a vertex, with properties for an object represented as self-edges on the object vertex, and spatial relations between objects represented as edges between vertices. The algorithm seeks to find the cheapest subgraph, calculated from the edge costs. We use the implementation available from Viethen et al. (2008), which adds a preference order to decide between edges with the same cost during search. This has been one of the best-performing systems in recent generation challenges (Gatt and Belz, 2008; Gatt et al., 2009).

An important commonality between these algorithms, and much of the work on REG that they have influenced, is the focus on *unique identification* and operating *deterministically*. Both produce one property set (and only one), and stop once a target item has been uniquely identified (or else fail). Their driving goal is to rule out distractor objects.

In the approach introduced here, the algorithm produces a distribution over several possible outputs, and the initial driving mechanism is based on likelihood estimates for each attribute independent of the other objects in the scene, rather than ruling out all distractors. This offers a way to capture some aspects of human-like reference, including under- and overspecification and speaker variation. Due to the fundamentally different objective of this algorithm, we will call the kind of expression it generates an *identifying description*, following Searle (1969). This is a description that the system finds (1) useful to describe the referent and (2) true of the referent.

4 The Algorithm

The Visible Objects Algorithm iterates through lists of visible attributes, stochastically adding properties to the property set it will generate. After this initial search, the algorithm then scans through the objects in the scene, roughly corresponding to how people scan a scene when referring (Pechmann, 1989). The target referent type, corresponding to the head noun in the final generated description, is added to the property set at the end of the algorithm.

We represent the basic components of the algorithm graphically in Figure 3. Full code is available online.[4] After START, the algorithm proceeds in parallel through a list of absolute attributes and a list of relative attributes. The likelihood of generating a property is a function of the prior likelihood α_{att} and γ, a penalty on the length of the constructed property set up to that point. This ensures that only a few properties are generated for a referent, and the expression will not be too complex. This is also in line with recent research suggesting that there are rarely more than three adjectives in a visual noun phrase

[3] https://github.com/nltk/nltk_contrib/blob/master/nltk_contrib/referring.py retrieved 1.Aug.2012.

[4] https://github.com/itallow/VisibleObjectsAlgorithm.

(Berg et al., 2011). Once the algorithm hits END, it scans through the objects in the scene. If it finds an object that is the same type as the referent object, the algorithm checks through the attributes again in a preference order akin to the IA, comparing the object's properties against the referent's and generating properties as a function of the length penalty alone. If the algorithm does not find an object that it is the same type, no further properties are added.

4.1 Requirements

The algorithm requires the following:

1. Prior likelihood estimates on the inclusion of different attributes. Represented as α_{att}.
2. Ordered list of absolute attributes beyond color. Represented as `AP`.
3. Ordered list of relative attributes beyond size. Represented as `RP`.
4. Ordered list of all attributes. Represented as `P`.
5. Ordered list specifying the order in which to scan through other scene objects. The current implementation uses the order in which the objects are listed in the corpora it is run on.

(1) is similar to the cost functions for GB, but attributes are selected non-deterministically using prior likelihoods. (2), (3), and (4) are similar to the IA's and GB's preference order. For our evaluation corpora, `AP` is empty and `RP` contains location and orientation. (5) is novel to this algorithm, defining an order in which to compare the target object against other objects in the scene. This is inspired by the process of incremental speech production (Pechmann, 1989), where speakers scan objects during naming, incrementally producing properties.

4.2 The Stochastic Process

Generally speaking, we want to penalize longer descriptions and encourage the attributes that we know people are likely to use. We can encourage a likely attribute by using its prior likelihood as an estimate of whether to include it. We can penalize longer descriptions with a penalty proportional to the length of the property set under construction. In other words, given a prior likelihood estimate for including an attribute att, α_{att}, and the property set constructed so far A, we compute whether to add a property for att to A as a function of α_{att} and the length-based penalty γ:

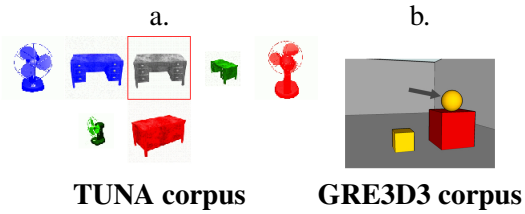

Figure 4: Example scenes from corpora.

$$f(A \cup \{x\}) = \gamma \alpha_{att}$$

where

$$\gamma = \begin{cases} \frac{1}{\lambda |A|} & \text{if } |A| > 0 \\ 1 & \text{otherwise} \end{cases}$$

and λ is an empirically determined weight. The algorithm then chooses a random number n, $0 \leq n \leq 1$. If $n < f(A \cup \{x\})$, it adds the property.

4.3 Scanning Through Objects

After the initial pass through the properties, the algorithm compares each object in the scene that is the same type as the target. If the values for an attribute are different, then the corresponding property is added to the property set based on the length penalty alone; when the goal is unique identification, the algorithm can use no penalty. In development, we found that incrementally scanning through objects after initially adding properties resulted in better performance than an algorithm that did not contain this step.

4.4 Worked Example

Suppose the input in Figure 6 (visualized in Figure 4a), with the goal of referring to obj_1 by producing a property set A. First, the algorithm scans through color and size in parallel. For color, it finds the corresponding value grey; with a computer vision input, this would be possible using the object pixels as features. There is no length penalty at this point ($|A|$=0), so it adds the property color:grey to A with likelihood α_{color}. For our evaluation domains, α_{color} is around .90 across folds, and so a color property is usually added.

For size, the algorithm finds an appropriate value using the Size Algorithm from Mitchell et al. (2011). The Size Algorithm uses the average height and

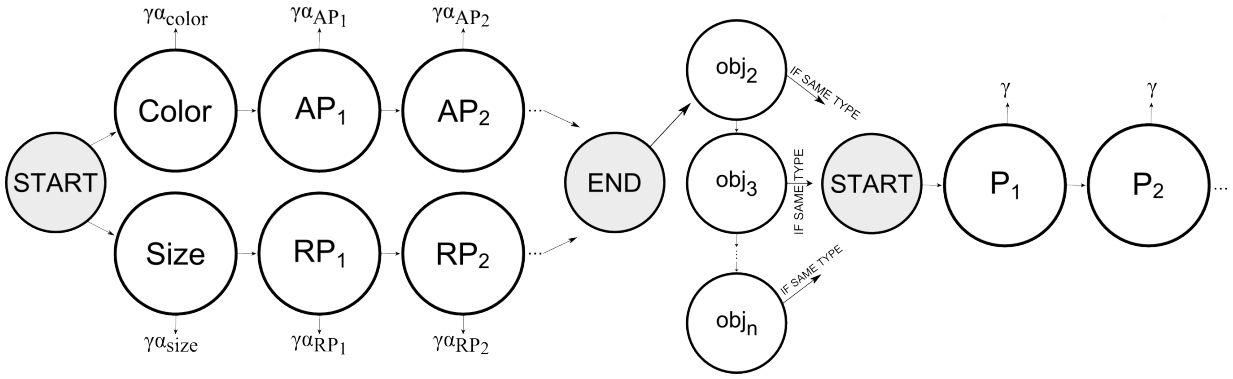

Figure 3: Basic model for generating visual reference.

width of all objects that are the same type as the referent object; in this case, obj_2, obj_3, obj_4. This returns a size value large, and so the property size:large is added to A with likelihood α_{size} (around .40 to .70 across folds, depending on the domain).

The most likely property set at this point is simply ⟨color:grey⟩. The next most likely is ⟨color:grey, size:large⟩, then ⟨size:large⟩. There are no further absolute properties in this example, but there are values for the relative attributes loc (location) and ori (orientation). Assuming RP=⟨location, orientation⟩, the algorithm first analyzes location, then orientation. A location property is added to A with likelihood α_{loc} multiplied by the length penalty γ=$\frac{1}{(\lambda\times1)}$ if A=⟨color:grey⟩; γ=$\frac{1}{(\lambda\times2)}$ if A=⟨color:grey, size:large⟩, etc.; and an orientation property is added to A with likelihood α_{ori} multiplied by the length penalty γ=$\frac{1}{(\lambda\times1)}$ if the property set is ⟨color:grey⟩, etc. At this point, the likelihood of adding further properties quickly diminishes.

Once all properties have been analyzed, the algorithm scans through the objects in the scene. For each object $\text{obj}_2 \ldots \text{obj}_n$, if the object is the same type as the target object obj_1, then any different property of the target referent is added to A with a likelihood based on the length penalty alone γ. ⟨type:desk⟩ is added at the end.

For this example scene, the algorithm will generate the property sets ⟨color:grey, type:desk⟩, ⟨color:grey, size:large, type:desk⟩, ⟨size:large, type:desk⟩, ⟨color:grey, ori:front, type:desk⟩, ⟨color:grey, loc:(3,1), type:desk⟩, etc., with different frequencies. Due to the length penalty, generated property sets will almost never have more than 3 properties.

tg	color:yellow	size:(63,63)	type:ball	loc:right-hand
lm	color:red	size:(345,345)	type:cube	loc:right-hand
obj3	color:yellow	size:(70,70)	type:cube	loc:left-hand

Figure 5: Example input scene: GRE3D3 corpus. For IA And GB, gold-standard size values are provided rather than measurements (small, large).

obj1	colour:grey	size:(454,454)	type:desk	loc:(3,1) ori:front
obj2	colour:blue	size:(454,454)	type:desk	loc:(2,1) ori:front
obj3	colour:red	size:(454,454)	type:desk	loc:(3,2) ori:back
obj4	colour:green	size:(254,254)	type:desk	loc:(4,1) ori:left
obj5	colour:blue	size:(454,454)	type:fan	loc:(1,1) ori:front
obj6	colour:red	size:(454,454)	type:fan	loc:(5,1) ori:back
obj7	colour:green	size:(254,254)	type:fan	loc:(2,2) ori:left

Figure 6: Example input scene: TUNA corpus. For IA And GB, gold-standard size values are provided rather than measurements (small, large).

As such, although ⟨color:grey, type:desk⟩ would sufficiently distinguish the intended referent, we instead produce a variety of sets, overspecifying in some instances (e.g., ⟨color:grey, ori:front, type:desk⟩), and with a small chance of underspecifying in others (e.g., ⟨size:large, type:desk⟩).

5 Evaluation Algorithms & Corpora

5.1 Corpora

We evaluate on two well-known REG corpora, the GRE3D3 corpus (Viethen and Dale, 2008) and the singular furniture section of the TUNA corpus (van Deemter et al., 2006). Both corpora contain expressions elicited to computer-generated objects, and so provide a reasonable starting point for evaluating reference to visible objects. For all algorithms, we evaluate on the selection of referent attributes. Lexical choice and word order are not taken into account. Example images from GRE3D3 and TUNA are shown in Figure 4, and example algorithm input

from these corpora are shown in Figures 5 and 6.

In GRE3D3, we evaluate on the selection of type, color, size, and location, but leave aside properties of relatum objects, which are not currently addressed by this algorithm or the IA. In TUNA, we evaluate on the selection of type, color, size and orientation.[5]

5.2 Algorithms

5.2.1 The Incremental Algorithm

The Incremental Algorithm requires a preference order list (PO) specifying the order to iterate through scene attributes. We determine the preference order from corpus frequencies using cross-validation to hold out a test scene and list attributes from the training scenes in descending order. We find that color precedes size in the preference orders, in line with recent research showing that this allows the algorithm to perform optimally on the TUNA corpus (van Deemter et al., 2012a). In development, we find that IA performs best with type as the last attribute in the PO, and report on numbers with this approach.

5.2.2 The Graph-Based Algorithm

The version of the Graph-Based Algorithm that we use is available from Viethen et al. (2008). This algorithm requires (1) a set of cost functions for each edge, and (2) a PO for deciding between properties in the case of a tie. For (1), we use the method from Theune et al. (2011) to assign two costs (0, 1) to the edges. We first determine the relative frequency with which each property is mentioned for a target object, and then create costs for each property using k-means clustering (k=2) in the Weka toolkit (Hall et al., 2009). We refer interested readers to the Theune et al. paper for further details. For (2), we follow the same method as for the Incremental Algorithm.

5.2.3 The Visual Objects Algorithm

The proposed algorithm requires α_{att}, which we estimate as the relative frequency of each attribute att in the training data. The ordered attribute lists for the algorithm (`AP`, `RP` and `P`) are built in the same way as the preference order list for the IA and GB, listing attributes from the training data in order of descending frequency. For these corpora, there are not absolute properties beyond color, so `AP` is empty.

6 Evaluation

Previous evaluation of REG algorithms have used measurements such as Uniqueness, Minimality, Dice (Belz and Gatt, 2008), and Accuracy (Gatt et al., 2009; Reiter and Belz, 2009). *Uniqueness* is the proportion of outputs that identify the referent uniquely, and *Minimality* is the proportion of outputs that are both minimal and unique. As our goal is to mimic human reference, these metrics are not as useful for the evaluations as the others.

The *Dice* metric provides a value for the similarity between a generated description and a human-produced description, and therefore serves as a reasonable objective measure for how human-like the produced sets are. Given the generated property set (D_S) and the human-produced property set (D_H), Dice is calculated as:

$$\frac{2 \times |D_S \cap D_H|}{|D_S| + |D_H|}$$

For each input domain, we evaluate over boolean values (included or excluded) for the attributes D (see Table 1). Note that this means the specific values for the attributes are not compared. In this formulation based on boolean values, $|D_S|$=$|D_H|$=$|D|$ and Dice reduces to:

$$\frac{|D_S \cap D_H|}{|D|}$$

Calculating Dice over the same number of attributes for both the observed and generated data has the nice mathematical property of making Dice equal to other common metrics for evaluating a model, including Accuracy, Precision, and Recall.[6]

Since the proposed algorithm is stochastic, this introduces a problem in using a metric that compares single expressions. We therefore seek to find the best alignment between the set of expressions produced by the algorithm and the set of expressions produced by people. We formulate this alignment as an assignment problem weighted by Dice. For the corpus of observed property sets H and the corpus of generated property sets S, we find the best align-

[5] We remove location from evaluation in this corpus. Location is not annotated directly, but split such that only x-dimension or y-dimension may be marked for a reference.

[6] A false positive is a false negative, and there are no true negatives, so all four metrics are equivalent.

Example Expression	Corresponding Property Set	Evaluated Property Set
the red ball	⟨color:red, type:ball⟩	type:1 color:1 size:0 loc:0

Table 1: Example human expression and corresponding boolean-valued property set for evaluation in GRE3D3, with D={type, color, size, and location}.

ment x out of all possible alignments X between the corpora:

$$\arg\max_{x \in X} \sum_{(S,H) \in x} \text{Dice}(D_S, D_H)$$

This may be solved in polynomial time using the Hungarian method (Kuhn, 1955; Munkres, 1957). Note that because IA and GB are deterministic, finding an optimal alignment is trivial. We call this method ALIGNED DICE.

It is an open question whether an alignment-based evaluation is fair: the proposed algorithm has more than one chance to match the human descriptions. In the second evaluation method (MAJORITY) we address this issue, comparing how often the *most frequent* generated set compares with the most frequent observed set. We run the proposed algorithm 1,000 times, and the generated property sets are ordered by frequency. The most frequent generated set is compared against the most frequent human-produced set. The majority score is the percentage of folds where these two sets match. For IA and FB, the most frequent generated set is the only generated set. This is a simple way to fairly compare the output of deterministic and non-deterministic algorithms. There are no ties in the generated sets, but in the case of a tie in the observed data, we count a match if any match the most frequent generated set.

6.1 GRE3D3

We randomly select two scenes (7, 9) from Set 1 and their mirrored counterparts in Set 2 (17, 19) for development. We empirically determine λ=5 for the length-based penalty γ in the proposed algorithm.

We use the eight remaining scenes in each Set for eight-fold cross-validation, estimating parameters for the algorithms on the seven training scenes in each fold, as discussed in Section 5.2.

For ALIGNED DICE, we run the proposed algorithm five times in each fold and report the average score. Results are shown in Table 2.[7]

Algorithm	ALIGNED DICE		MAJORITY	
	Set 1	Set 2	Set 1	Set 2
Proposed Alg.	**88.23**	**90.06**	62.50	**50.00**
IA	87.71	85.13	62.50	25.00
GB	87.71	88.73	62.50	**50.00**

Table 2: GRE3D3: Results (in %).

Algorithm	ALIGNED DICE		MAJORITY	
	+LOC	-LOC	+LOC	-LOC
Proposed Alg.	**88.75**	**86.07**	**40.00**	40.00
IA	81.79	81.55	0.00	**100.00**
GB	75.36	66.04	20.00	20.00

Table 3: TUNA: Results (in %).

The proposed Visible Objects Algorithm achieves higher accuracy than either version of the Incremental Algorithm or the Graph-Based Algorithm using ALIGNED DICE. In MAJORITY, the Graph-Based and the Visible Objects Algorithm both predict the majority property set in this evaluation at least 50% of the time. The algorithm is competitive with the state of the art on this corpus.

6.2 TUNA

TUNA is split into two conditions: subjects discouraged to use location (-LOC) or not (+LOC). We randomly hold out two scenes from both conditions (1 and 2), and find a value of λ=5 again works well on the development data.

As in the GRE3D3 corpus, we use the TUNA scenes in five-fold cross-validation, estimating parameters on the four training scenes in each fold. For ALIGNED DICE, we average over five runs of the algorithm, and for MAJORITY, we run the proposed algorithm 1,000 times for each test scene.

Results are shown in Table 3. Again we see that the proposed Visible Objects Algorithm is competitive with the IA and GB for both ALIGNED DICE and MAJORITY. GB performs poorly here, and this may be due to the data sparsity issue that arises when requiring the algorithm to train on properties.[8] In

[7] We do not report statistical significance; the proposed algorithm produces several possible outputs for one input, while the IA and GB produce only one.

[8] The original property-based weighting approach (Theune et al., 2011; Koolen et al., 2012, see Section 5.2) trained on object collections that were identical to their test data in all properties except x- and y-dimension, and so this was less of an issue. We hope to explore whether basing weights on attributes alone

MAJORITY, the Visible Objects Algorithm is relatively stable across conditions, generating the majority property set in 40% of the test scenes. It does not outperform the IA in the -LOC condition, but the IA has a large range across the two conditions (0% and 100%).

7 Conclusions and Future Work

We have introduced a new algorithm for generating referring expressions, inspired by human and computer vision and aiming to refer in a human-like way to visible objects. The algorithm successfully generates the most common attributes that people choose for different objects, and offers a varied output to capture speaker variation. In contrast to most algorithms for the generation of referring expressions, which have aimed to produce distinguishing descriptions when these exist (Krahmer and van Deemter, 2012), the core idea behind this algorithm is to generate what is likely for a speaker in a visual domain. Since the driving mechanism behind the algorithm is not to uniquely identify the object, but rather to pipeline the analysis of properties in a way similar to human visual processing, the generated expression may be overspecified or underspecified.

We are limited by available REG corpora to reliably assess methods for generating more complex absolute properties like shape and material, but adding such properties would help advance the generation of human-like reference in visual scenes and offers further points of connection between the generation process and computer vision property detection. Models for generating more complex spatial relations are currently available, and are a natural extension to this framework (e.g., those of Kelleher and Costello (2009)) as object detection becomes more robust.

We may also be able to build more sophisticated graphical models as larger corpora become available. For example, modeling the conditional probability of generating reference for a property v_n given the previously generated context $p(v_n|v_1 \dots v_{n-1})$ may bring us closer to human-like output.

There are several additional issues that do not arise in this evaluation, but we expect must be accounted for when referring to naturalistic objects in visual domains. These include:

- The interconnected nature of properties, where some properties entail others; for example, a wooden object is likely to be called *wooden*, referring to its material, rather than *tan* or *brown*.
- The role of typicality, where properties are selected because they are atypical for the object.
- Referring to more complex properties, e.g., material, texture, etc., and object parts.
- Better methods for determining the length penalty and attribute likelihoods.

We hope to discuss extensions to this algorithm covering these aspects of reference in future work.

improves performance.

Acknowledgments

Funding for this research has been provided by SICSA and ORSAS. We thank the anonymous reviewers for useful comments on this paper.

References

Carlos Areces, Alexander Koller, and Kristina Striegnitz. 2008. Referring expressions as formulas of description logic. *Proceedings of the 5th International Natural Language Generation Conference (INLG 2008)*, pages 42–29.

Ellen Gurman Bard, Robin Hill, Manabu Arai, and Mary Ellen Foster. 2009. Accessibility and attention in situated dialogue: Roles and regulations. *Proceedings of the Workshop on the Production of Referring Expressions (PRE-CogSci 2009)*.

Anja Belz and Albert Gatt. 2008. Intrinsic vs. extrinsic evaluation measures for referring expression generation. *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL 2008)*, pages 197–200.

Alexander C. Berg, Tamara L. Berg, Hal Daumé III, Jesse Dodge, Amit Goyal, Xufeng Han, Alyssa Mensch, Margaret Mitchell, Karl Stratos, and Kota Yamaguchi. 2011. An exploration of how to learn from visually descriptive text. *JHU-CLSP Summer Workshop Whitepaper*.

Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc., Sebastopol, CA.

Sarah Brown-Schmidt and Michael K. Tanenhaus. 2006. Watching the eyes when talking about size: An investigation of message formulation and utterance planning. *Journal of Memory and Language*, 54:592–609.

Herbert H. Clark and Gregory L. Murphy. 1982. Audience Design in Meaning and Reference. In J. F. LeNy and W. Kintsch, editors, *Language and Comprehension*, volume 9 of *Advances in Psychology*, pages 287–299. North-Holland, Amsterdam.

Herbert H. Clark, Robert Schreuder, and Samuel Buttrick. 1983. Common ground and the understanding of demonstrative reference. *Journal of Verbal Learning and Verbal Behavior*, 22:245–258.

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

Robert Dale. 1989. Cooking up referring expressions. *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics (ACL 1989)*.

P. E. Engelhardt, K. Bailey, and F. Ferreira. 2006. Do speakers and listeners observe the gricean maxim of quantity? *Journal of Memory and Language*, 54:554–573.

Fang Fang, Huseyin Boyaci, Daniel Kersten, and Scott O. Murray. 2008. Attention-dependent representation of a size illusion in human V1. *Current biology*, 18(21):1707–1712.

Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. 2009. Describing objects by their attributes. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*.

V. Ferrari and A. Zisserman. 2007. Learning visual attributes. *Advances in Neural Information Processing Systems (NIPS 2007)*.

William Ford and David Olson. 1975. The elaboration of the noun phrase in children's description of objects. *The Journal of Experimental Child Psychology*, 19:371–382.

David A. Forsyth. 2011. Personal communication. Video clip of communication available from: http://vimeo.com/40553150. At 1:06:46.

T. M. Gann and D. J. Barr. 2013. Speaking from experience: Audience design as expert performance. *Language and Cognitive Processes*. In press.

Albert Gatt and Anja Belz. 2008. Attribute selection for referring expression generation: New algorithms and evaluation methods. *Proceedings of 5th International Natural Language Generation Conference (INLG 2008)*, pages 50–58.

Albert Gatt, Anja Belz, and Eric Kow. 2009. The TUNA REG challenge 2009: Overview and evaluation results. *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 174–182.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: An update. *SIGKDD Explorations*, 11(1).

William S. Horton and Boaz Keysar. 1996. When do speakers take into account common ground? *Cognition*, 59(1):91–117.

Laurent Itti and Christof Koch. 2001. Computational modelling of visual attention. *Nature Reviews Neuroscience*, 2:194–203.

John Kelleher and Fintan Costello. 2009. Applying computational models of spatial prepositions to visually situated dialog. *Computational Linguistics*, 35(2):271–306.

John Kelleher, Fintan Costello, and Josef van Genabith. 2005. Dynamically structuring, updating and interrelating representations of visual and linguistic discourse context. *Artificial Intelligence*, 167:62–102.

Boaz Keysar and Anne S. Henly. 2002. Speakers' overestimation of their effectiveness. *Psychological Science*, 13(3):207–212.

Ruud Koolen, Martijn Goudbeek, and Emiel Krahmer. 2011. Effects of scene variation on referential overspecification. *Proceedings of the 33rd Annual Meeting of the Cognitive Science Society (CogSci 2011)*.

Ruud Koolen, Emiel Krahmer, and Mariët Theune. 2012. Learning preferences for referring expression generation: Effects of domain, language and algorithm. *Proceedings of the 7th International Workshop on Natural Language Generation (INLG 2012)*.

Emiel Krahmer and Mariët Theune. 2002. Efficient context-sensitive generation of referring expressions. *Information Sharing: Reference and Presupposition in Language Generation and Interpretation*, 143:223–263.

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

Emiel Krahmer, Sebastiaan van Erk, and André Verleg. 2003. Graph-based generation of referring expressions. *Computational Linguistics*, 29(1):53–72.

H. W. Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.

Margaret Mitchell, Kees van Deemter, and Ehud Reiter. 2011. Two approaches for generating size modifiers. *Proceedings of the 13th European Workshop on Natural Language Generation (ENLG 2011)*.

James Munkres. 1957. Algorithms for the assignment and transportation problems. *Journal of Industrial and Applied Mathematics*, 5(1):32–38.

Scott O. Murray, Huseyin Boyaci, and Daniel Kersten. 2006. The representation of perceived angular size in human primary visual cortex. *Nature Neuroscience*, 9(3):429–434.

David R. Olson. 1970. Language and thought: Aspects of a cognitive theory of semantics. *Psychological Review*, 77:257–273.

Thomas Pechmann. 1989. Incremental speech production and referential overspecification. *Linguistics*, 27:89–110.

Ehud Reiter and Anja Belz. 2009. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35(4):529–558.

Ehud Reiter and Robert Dale. 1992. A fast algorithm for the generation of referring expressions. *Proceedings of the 14th International Conference on Computational Linguistics (COLING 1992)*, 1:232–238.

Ehud Reiter and Somayajulu Sripada. 2002. Human variation and lexical choice. *Computational Linguistics*, 28:545–553.

D. Samuel Schwarzkopf, Chen Song, and Geraint Rees. 2010. The surface area of human V1 predicts the subjective experience of object size. *Nature Neuroscience*, 14(1):28–30.

J. R. Searle. 1969. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge.

Susan Sonnenschein. 1985. The development of referential communication skills: Some situations in which speakers give redundant messages. *Journal of Psycholinguistic Research*, 14:489–508.

Mariët Theune, Ruud Koolen, Emiel Krahmer, and Sander Wubben. 2011. Does size matter – how much data is required to train a REG algorithm? *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*.

Anne M. Treisman and Garry Gelade. 1980. A feature integration theory of attention. *Cognitive Psychology*, 12:97–13.

Anne Treisman. 1985. Preattentive processing in vision. *Computer Vision, Graphics, and Image Processing*, 31:156177.

L. G. Ungerleider and M. Mishkin. 1982. Two Cortical Visual Systems. In D. J. Ingle, M. Goodale, and R. J. W. Mansfield, editors, *Analysis of Visual Behaviour*, chapter 18, pages 549–586. The MIT Press.

Kees van Deemter, Ielka van der Sluis, and Albert Gatt. 2006. Building a semantically transparent corpus for the generation of referring expressions. *Proceedings of the 4th International Conference on Natural Language Generation (INLG 2006)*.

Kees van Deemter, Albert Gatt, Ielka van der Sluis, and Richard Power. 2012a. Generation of referring expressions: Assessing the incremental algorithm. *Cognitive Science*, 36(5):799–836.

Kees van Deemter, Emiel Krahmer, Roger van Gompel, and Albert Gatt. 2012b. Towards a computational psycholinguistics of reference production. *TopiCS: Production of Referring Expressions - Bridging the Gap between Computational and Empirical Approaches to Reference*.

Roger P. G. van Gompel, Albert Gatt, Emiel Krahmer, and Kees van Deemter. 2012. PRO: A computational model of referential overspecification. *Architectures and Mechanisms for Language Processing (AMLaP 2012)*.

Jette Viethen and Robert Dale. 2008. The use of spatial relations in referring expression generation. *Proceedings of the 5th International Natural Language Generation Conference (INLG 2008)*, pages 59–67.

Jette Viethen and Robert Dale. 2010. Speaker-dependent variation in content selection for referring expression generation. *Proceedings of the 8th Australasian Language Technology Workshop (ALTW 2010)*, pages 81–89.

Jette Viethen, Robert Dale, Emiel Krahmer, Mariët Theune, and Pascal Touset. 2008. Controlling redundancy in referring expressions. *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*.

Jette Viethen, Martijn Goudbeek, and Emiel Krahmer. 2012. The impact of colour difference and coloure codability on reference production. *Proceedings of the 34th Annual Meeting of the Cognitive Science Society (CogSci 2012)*.

G. J. Whitehurst. 1976. The development of communication: Changes with age and modeling. *Child Development*, 47:473–482.