Referring Expression Generation: What Can We Learn From Human Data?

Jette Viethen and Robert Dale
Centre for Language Technology
Macquarie University, Australia
[jviethen|rdale]@science.mq.edu.au

Abstract

In this paper, we reflect on what we can learn about the processes involved in the generation of referring expressions by looking at a corpus of human-produced data. We find that the data vastly underspecifies what might be involved algorithmically, but it does rule out a number of popular algorithms for referring expression generation as candidates for models of what people do. We posit an alternative algorithmic schema which forces us to focus more clearly on the questions that need to be answered experimentally if we are to develop algorithms that emulate human behaviour on this task.

Introduction

There are two distinct exercises we might be engaged in when we develop algorithms for content determination in referring expression generation. One is an exercise in engineering, where the goal is to develop algorithms that are able to effectively identify intended referents for hearers, for use in sophisticated applications of natural language generation. Alternatively, we might be engaged in an exercise in computational psycholinguistics, where our goal is to model what it is that a speaker does when he or she refers to an entity. Much existing work is ambivalent as to which of these two tasks is the primary concern; and while we might argue that the two exercises are not unrelated, there are also quite significant differences that follow from pursuing either goal. Our interest in the present paper is in the second of these exercises: how might we develop algorithms that emulate the referring behaviour of humans, and in so doing, perhaps begin to explain how humans carry out this task?

An obvious starting point for such an endeavour is to see what the nature of human data might tell us about possible algorithms. So, in this paper, we examine a collection of human-produced referring expressions, and make some observations based on this data. Then, we consider current algorithms, and establish that they do not serve as good models of what people do. Next, we attempt to identify some constraints that algorithms would have to meet in order to be compatible with the human data, and on the basis of this analysis we propose a general schema for the task of referring expression generation. Finally, we identify a number of crucial details this schema leaves unspecified. Our aim is to provoke questions that lead to experimental work that will help to illuminate how these details might be determined.

The Data

The data we explore here has been discussed in detail elsewhere (see (Dale & Viethen, 2009)), so we will provide here only a summary for present purposes.

The GRE3D3 corpus was gathered via a web-based experiment where subjects were asked to produce referring expressions that would enable a listener to identify one of a number of objects shown on the screen. Each participant was assigned one of two trial sets of ten scenes each; the two trial sets are superficially different, but the elements of the sets are pairwise identical in terms of the factors explored in the research. Scenes were presented one at a time. The complete set of 20 scenes is shown in Figure 1: Trial Set 1 consists of Scenes 1 through 10, and Trial Set 2 consists of Scenes 11 through 20. Each scene contains three objects, which we refer to as the *target* (the intended referent), the potential *landmark* (a nearby object), and the *distractor* (a further-away object). From the experiment we collected 623 descriptions of objects produced by 63 participants. Every one of these descriptions is an instance of one of the 18 *content patterns* shown in Table 1; for ease of reference, we label these patterns A through R. Each pattern indicates the collection of attributes used in the description, where each attribute is identified by the object it describes (tg for target, lm for landmark) and the attribute used (col, size and type for colour, size and type, respectively).

Table 2 shows the distribution of different patterns across the 10 different scenes.[1] The primary ob-

[1] Recall that Scene 1 is essentially the same as Scene 11, Scene 2 is essentially the same as Scene 12, and so on; so we consolidate the data for these pairwise similar scenes.

Label	Pattern	Example
A	⟨tg_col, tg_type⟩	*the blue cube*
B	⟨tg_col, tg_type, rel, lm_col, lm_type⟩	*the blue cube in front of the red ball*
C	⟨tg_col, tg_type, rel, lm_size, lm_col, lm_type⟩	*the blue cube in front of the large red ball*
D	⟨tg_col, tg_type, rel, lm_size, lm_type⟩	*the blue cube in front of the large ball*
E	⟨tg_col, tg_type, rel, lm_type⟩	*the blue cube in front of the ball*
F	⟨tg_size, tg_col, tg_type⟩	*the large blue cube*
G	⟨tg_size, tg_col, tg_type, rel, lm_col, lm_type⟩	*the large blue cube in front of the red ball*
H	⟨tg_size, tg_col, tg_type, rel, lm_size, lm_col, lm_type⟩	*the large blue cube in front of the large red ball*
I	⟨tg_size, tg_col, tg_type, rel, lm_size, lm_type⟩	*the large blue cube in front of the large ball*
J	⟨tg_size, tg_col, tg_type, rel, lm_type⟩	*the large blue cube in front of the ball*
K	⟨tg_size, tg_type⟩	*the large cube*
L	⟨tg_size, tg_type, rel, lm_size, lm_type⟩	*the large cube in front of the large ball*
M	⟨tg_size, tg_type, rel, lm_type⟩	*the large cube in front of the ball*
N	⟨tg_type⟩	*the cube*
O	⟨tg_type, rel, lm_col, lm_type⟩	*the cube in front of the red ball*
P	⟨tg_type, rel, lm_size, lm_col, lm_type⟩	*the cube in front of the large red ball*
Q	⟨tg_type, rel, lm_size, lm_type⟩	*the cube in front of the large ball*
R	⟨tg_type, rel, lm_type⟩	*the cube in front of the ball*

Table 1: The 18 different content patterns corresponding to the different forms of description that occur in the GRE3D3 corpus.

	Scene #									
Pattern	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
A	17	24			36	32	26			40
B	14	8	3		16	7	8	3		10
C		4		1			3		1	
D						1	1		1	
E	4		1			2				
F	2	1	15	44	5	3	2	25	40	8
G	1		14		2		1	14		1
H		1	1	13	2	1	2	1	17	2
I				3					1	
J			1			1				1
K			12					15		
L				1						
M	1		7					4		
N	11	13				14	14			
O		4					1			
P							1			
Q		3					2			
R	13	5	9			2	2	1		

Table 2: The number of content patterns per scene.

Pattern Sequence ⟨Scene #, Content Pattern⟩	No of participants
⟨1, A⟩, ⟨2, A⟩, ⟨3, G⟩, ⟨4, F⟩, ⟨5, A⟩, ⟨6, A⟩, ⟨7, A⟩, ⟨8, G⟩, ⟨9, F⟩, ⟨10, A⟩	2
⟨1, B⟩, ⟨2, B⟩, ⟨3, G⟩, ⟨4, H⟩, ⟨5, B⟩, ⟨6, B⟩, ⟨7, B⟩, ⟨8, G⟩, ⟨9, H⟩, ⟨10, B⟩	2
⟨1, N⟩, ⟨2, N⟩, ⟨3, K⟩, ⟨4, F⟩, ⟨5, A⟩, ⟨6, N⟩, ⟨7, N⟩, ⟨8, K⟩, ⟨9, F⟩, ⟨10, A⟩	6
⟨1, A⟩, ⟨2, A⟩, ⟨3, F⟩, ⟨4, F⟩, ⟨5, A⟩, ⟨6, A⟩, ⟨7, A⟩, ⟨8, F⟩, ⟨9, F⟩, ⟨10, A⟩	9

Table 3: Sequences of content patterns found more than once.

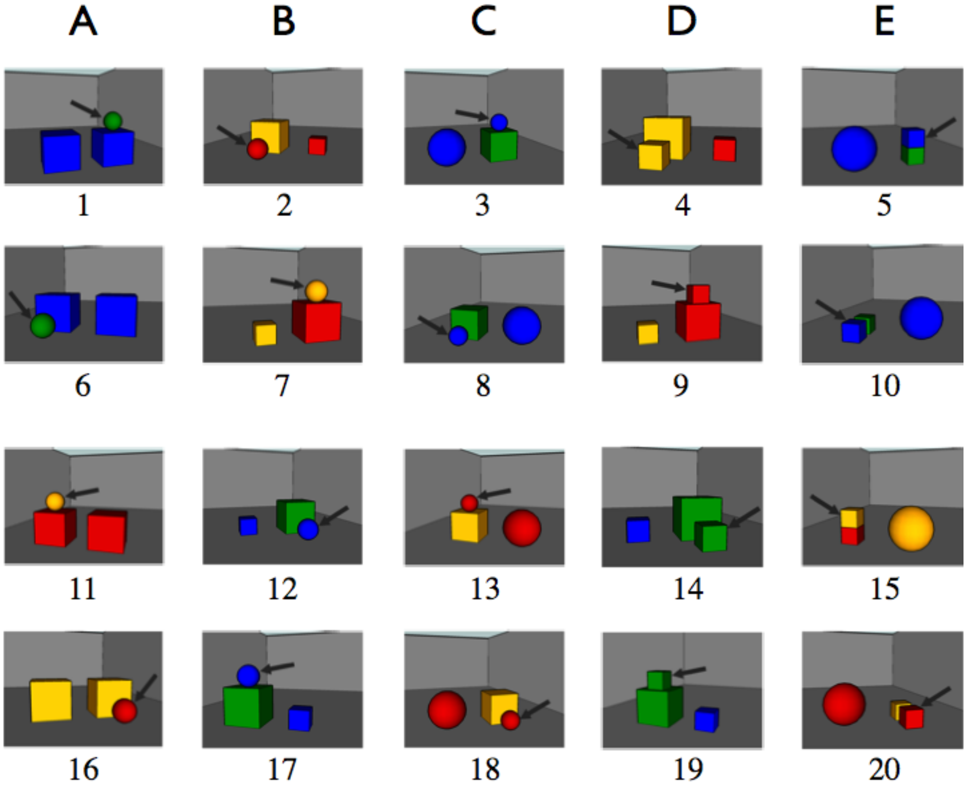

Figure 1: The stimulus scenes.

servation of relevance here is that there is no one 'correct' answer for how to refer to the target in any given scene: the number of semantically-distinct referring expressions varies from five (in Scenes 4, 5, 9 and 10) to 12 (in Scene 7). All of these are distinguishing descriptions, so all are acceptable forms of reference, although some contain more redundancy than others. Most obvious from the table is that, for many scenes, there is a predominant form of reference used; for example, pattern F (⟨tg_size, tg_col, tg_type⟩) accounts for 43 (68%) of the descriptions used in Scene 4, and pattern A (⟨tg_col, tg_type⟩) is very frequently used in a number of scenes.

All participants were presented with the scenes in the same order, so we can also look at variation across subjects in terms of the sequences of descriptions they provided. Across the 63 subjects, there are 47 different sequences; of these, only four occur more than once. The recurrent sequences, i.e. those used by at least two people, are shown in Table 3.

All of the above supports an observation which is hardly new, and yet appears to have been ignored until only recently: when it comes to reference, different people do different things and each person does different things in different situations.

A second observation from this data is that the referring expressions that people produce are often informationally redundant. 372 of the 623 descriptions in the corpus (59.71%) contained at least one property that could be dropped without rendering the description ambiguous. Of these, 217 contain at least two redundant properties.

Existing Algorithms

Existing algorithms have generally not been designed with variation in mind. As we noted earlier, much of the work on algorithms does not clearly state whether its objective is an engineering one or a psycholinguistic one, following the distinction we made at the beginning of this paper; and so, if these algorithms were actually developed with engineering objectives in mind, it would be unfair to criticise them for failing to meet the objectives of psycholinguistic modelling. Nonetheless, an examination of the ways in which these algorithms do not match human behaviour is insightful.

Given an intended referent R, a set of distractors C, a set of attributes L_R, and the set of properties to use in a description D:

```
Let $D = \emptyset$
repeat
    add one selected attribute $\in L_R$ to $D$
    recompute $C$ given $D$
until $C = \emptyset$
```

Figure 2: The structure of referring expression generation algorithms

We focus here on two algorithms that form the basis of many more recent approaches: the Greedy Algorithm (Dale, 1989) and the Incremental Algorithm (Dale & Reiter, 1995), the latter of which has served as a starting point for many other algorithms in the literature. The Greedy Algorithm targets the construction of *minimal distinguishing descriptions*, i.e. fully distinguishing descriptions which contain no informational redundancy and are as short as possible. The Incremental Algorithm drops the explicit aim of finding the shortest description, but, like the Greedy Algorithm, considers its work done as soon as it has constructed a distinguishing description. It selects properties from an ordered predefined list without backtracking, which can sometimes result in descriptions containing redundant properties.

The behaviour of both these algorithms is captured by the algorithmic schema shown in Figure 2. What makes one algorithm different from another, in terms of this schema, is the particular means by which the next attribute to include is selected: in the Greedy Algorithm the attribute which rules out most potential distractors is chosen next, whereas in the Incremental Algorithm the next attribute selected comes from a pre-determined preference order, and is included provided it adds some discrimination to the description. This embodies what we might call 'the assumption of serial dependence': both algorithms make the inclusion of a property dependent on its discriminatory power *given the properties that have been selected so far*.

Now, neither of these algorithms were necessarily developed as models of the human production of referring expressions. However, it is instructive to review how they behave differently from humans.

First, while these algorithms strive to avoid redundancy in the descriptions they produce, as we have already noted, people on the whole do not. The Greedy Algorithm's prioritisation of minimality above all else makes it a poor model of human behaviour. The lack of backtracking in the Incremental Algorithm can sometimes result in descriptions that contain redundant properties, and this might lead us to think that it is a better model of human behaviour. But our data contains some redundant expressions that the algorithm will never generate, such as *the top yellow cube* for Scene 15, where either *the top cube* or *the yellow cube* would have sufficed to describe the target.

The algorithms also fail as models of the variation found in human data. In the case of the Greedy Algorithm, the only scope for variation in output arises from random choices to be made whenever multiple properties have the same discriminatory power. In earlier work (Viethen & Dale, 2006), we demonstrated that the Incremental Algorithm could account for *most* variation in a different corpus by changing the preference ordering; but in order to do this for either corpus, a given speaker's preference ordering has to change from one scene to the next, which does not seem an entirely plausible characterisation of what people do.

What The Data Supports

So: existing algorithms don't seem to serve as good explanations of the human data. Can the data tell us what a more explanatory algorithm would have to be like?

In (Dale & Viethen, 2009), we noted that, while there appears to be great variety at the level of the descriptions the participants in our experiment produced, there is less variety, and more commonality, when we look instead at the extent to which the use of particular properties is shared by different speakers on the basis of what we called *characteristics of scenes*.[2] We used Weka (Witten & Frank, 2005) to train decision trees on the scene characteristics and participant IDs to predict whether each property, taken individually, should be included in a description; we also tried to learn the full content pattern. The learner's success rate was compared to the baseline of Weka's 0-R classifier, which simply chooses the majority class.[3]

Table 4 demonstrates that the data allows us to

[2] See (Dale & Viethen, 2009) for a full description of what counts as a characteristic of a scene; in essence, these are primarily binary properties of the scene as a whole, such as whether or not the target object and the landmark object share colour.

[3] The results reported are for the accuracy of a pruned J48 decision tree, under 10-fold cross-validation.

		Using Scene	Using Scene	Number
Learned Item	Baseline	Characteristics	Characteristics	of
	(0-R)	Only	and Participant	Heuristics
tg_col inclusion	78.33%	78.33%	**89.57%**	5
tg_size inclusion	57.46%	**90.85%**	90.85%	1
rel inclusion	64.04%	65.00%	**81.22%**	17
lm_col inclusion	74.80%	**87.31%**	**93.74%**	15
lm_size inclusion	88.92%	**95.02%**	95.02%	1
full content pattern	28.01%	**47.99%**	**57.62%**	37

Table 4: Accuracy of learning attribute inclusion and full content patterns. Statistically significantly increases (p<0.01) are marked in bold.

learn whether an individual property should be included with a much higher success rate than it allows us to learn which content pattern to use, no matter which set of features we look at. The last column in Table 4 shows the number of different scene-based heuristics across speakers for each learned item. To predict the full content pattern for each person, the learner requires 37 different heuristics, of which 32 are only used for one participant each. The numbers of different heuristics required to predict the inclusion of individual properties are much lower, and, as we discuss in (Dale & Viethen, 2009), there is quite a lot of overlap between participants in terms of the heuristics used.

The 'context-independent' strategies (the 0-R baselines) work surprisingly well for predicting the inclusion of some attributes in the human data. As has been noted in other work, colour is often included in referring expressions irrespective of its discriminatory power, and this is borne out by the data here. Perhaps more suprising is the large degree to which the inclusion of landmark size is captured by a context-independent strategy.

Improvement on all attributes other than target colour increases when we take into account the characteristics of the scenes, confirming the widely-held view that the context of reference does indeed make a difference. When we add participant ID to the features used in the learner, performance improves further still, indicating that there are speaker-specific consistencies across contexts.

In our earlier work, we used these findings to argue for a notion of *speaker profiles*, by means of which the referring behaviour of a given speaker can be captured as a collection of attribute-specific heuristics. These heuristics are exactly what is learned in the machine learning experiments for each individual property.

The most important observation here is that, in order to capture the variation in the data, we considered the properties used in descriptions *independently of one another*. As we saw in the previous section, this is not what the Greedy Algorithm and the Incremental Algorithm do. Both of these algorithms make the inclusion of each property dependent on which other properties have already been chosen. Our analysis here suggests that if we abandon this assumption of serial dependency, we may get closer to modelling the human data; it seems plausible that humans, at least sometimes, select properties independently of one another.

The second key element of the schema presented in Figure 2 is a check to see if the content of the description chosen so far is sufficient for the task. People's high success rate in identification tasks makes it implausible to suggest that the world just happens to be set up so that 'unconsidered references', as we might call them, tend to work. Examples like that cited by (Sluis, 2005) suggest strongly that speakers do indeed self-monitor to determine whether the descriptions they are constructing achieve their intended goals:

> Uhm, I'm gonna transfer to the phone on the table by the red chair ... [points in the direction of the phone] the ... the red chair, against the wall, uh the little table, with the lamp on it, the lamp that we moved from the corner? ... the black phone, not the brown phone ...

This leads us to propose a more general schema for referring expression generation that is a generalisation of the previous one; this is presented in Figure 3. It differs from the earlier schema in two regards. First, to break the assumption of serial dependency, we allow any number of properties to be included without checking for satisfaction after each one. Second, we generalise the satisfaction check so

Given an intended referent R, a set of requirements R_R, a set of attributes L_R, and the set of properties to use in a description D:

```
Let $D = \emptyset$
repeat
    add some attributes $\in L_R$ to $D$
until $D$ satisfies $R_R$
```

Figure 3: A proposed structure for referring expression generation

that, rather than this just being concerned with the question of whether or not the description is distinguishing, any number of requirements can play into the decision. In particular, we expect factors such as person-specific preferences for or against redundancy, akin to Carletta's 'cautiousness' (Carletta, 1992), or for and against certain types of properties, will play a role here, as well as external factors such as task-criticalness and knowledge about the listener's conceptualisation of the task environment.

One Step Back, Two Steps Forward

It may seem like we have gone backwards here: the algorithm schema in Figure 3 leaves more details unspecified than did the earlier schema. But our point is that, in fact, this is as it should be: *the data does not warrant any stronger claims about the nature of the process of referring expression generation.* This is not a bad thing; it forces us to more clearly identify what it is that we don't know, and to begin to explore how we might fill those gaps in our knowledge. In particular, from Figure 3, we can now see that questions such as the following should serve as a focus of experimental investigation:

1. When are properties included in a description independently of one another?

2. What scene-specific, task-specific, and person-specific aspects result in particular properties being chosen?

3. What factors are involved in determining whether a description is satisfactory?

As our schema is more general than the schema that underlies existing algorithms, these questions are more general than the questions that had to be answered in order to make existing algorithms work.

Conclusions

Based on the analysis of a data set of human-produced distinguishing descriptions, we have shown that the classic algorithms for content selection for referring expressions are inadequate as explanations of how humans describe objects. While these algorithms might not have been designed with this aim in mind, it is clear that if our aim is to model human referring behaviour, we need to follow a different approach.

We have presented the most specific model that is supported by the data, which turns out to leave many details unspecified. Finding ways to fill out the details in this general model is the necessary next step for those who are interested in modelling the human production of referring expressions.

References

Carletta, J. C. (1992). *Risk-taking and Recovery in Task-Oriented Dialogue.* Unpublished doctoral dissertation, University of Edinburgh.

Dale, R. (1989). Cooking up referring expressions. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics* (pp. 68–75). Vancouver, Canada.

Dale, R., & Reiter, E. (1995). Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, *19*(2), 233–263.

Dale, R., & Viethen, J. (2009). Referring expression generation through attribute-based heuristics. In *Proceedings of the 12th European Workshop on Natural Language Generation* (pp. 58–65). Athens, Greece.

Sluis, I. van der. (2005). *Multimodal reference, studies in automatic generation of multimodal referring expressions.* Unpublished doctoral dissertation, Tilburg University, Tilburg, The Netherlands.

Viethen, J., & Dale, R. (2006). Algorithms for generating referring expressions: Do they do what people do? In *Proceedings of the 4th International Conference on Natural Language Generation* (pp. 63–70). Sydney, Australia.

Witten, I. H., & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques.* San Francisco CA, USA: Morgan Kaufmann.