Natural Language Generation in Interactive Systems

Amanda Stent and Srinivas Bangalore
July 10, 2013

CAMBRIDGE UNIVERSITY PRESS
Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo

Cambridge University Press
The Edinburgh Building, Cambridge CB2 2RU, UK

Published in the United States of America by Cambridge University Press, New York

www.cambridge.org
Information on this title: www.cambridge.org/9780521XXXXXX

Printed in the United Kingdom at the University Press, Cambridge.

A catalogue record for this publication is available from the British Library

Library of Congress Cataloguing in Publication data

ISBN-13 978-0-521-XXXXX-X hardback
ISBN-10 0-521-XXXXX-X hardback

Contents

2 Referring expression generation in interaction: A graph-based perspective

Emiel Krahmer, Martijn Goudbeek and Mariët Theune

2.1 Introduction

Buying new chairs can be complicated. Many constraints have to be kept in mind, including your financial situation, the style and colour of the furniture you already own and possibly also the taste of your partner. But once you have made a tentative choice (say, the chair in Figure 2.1 on the left), there is one final hurdle: you have to inform the seller of your desire to buy it. Furniture stores tend to contain many chairs, so somehow you need to refer to your chair of choice, for example *I'd like to buy the wooden chair with the thin legs and solid seat, the red one with the open back*. It is hardly surprising that many people in this situation resort to pointing (*that one*). Of course, it would be helpful to know that salespeople usually refer to this chair as *the red Dutch lily chair*, because that would allow you to adapt to their way of referring.

This problem illustrates the importance of **reference** in everyday interactions: people can only exchange information about objects when they agree on how to refer to those objects. How this agreement may arise, and how we can model this in natural language generation, is the topic of this chapter. We argue that two possibly competing forces play a role. On the one hand, speakers may have inherent preferences for certain properties when referring to objects in a given domain. On the other, they may also have a tendency to adapt to the references produced by their dialogue partner. We describe how preferences can be determined, and how they interact with adaptation. We model this trade-off using a graph-based referring expression generation algorithm (Krahmer et al., 2003).

2.1.1 Referring expression generation

Given the centrality of reference in interaction, it is hardly surprising that one of the first things that children learn when acquiring language is how to refer to the objects surrounding them (Matthews et al., 2012). Similarly, when researchers develop a natural language generation (NLG) application, they typically also require a module that generates referring expressions (Mellish et al., 2006, Reiter and Dale, 2000). Such a referring expression generation (REG) module is typically dedicated to identifying **target** objects with respect to a set of **distractor** objects using natural language, and to do so the module needs to make a series

d_1 d_2 d_3

Figure 2.1: Three pieces of furniture

of related choices. First, it needs to select the form of reference, for example, deciding whether to refer to the chair using a deictic pronoun (*that one*) or a full description (*the chair with the armrests*). If the REG module decides to generate a description, two additional choices need to be made: which **properties** of the target should be included in the description, and how the selected property set can be expressed as a natural language description. These two processes are often referred to as **attribute selection** and **surface realisation**, respectively. Of these, attribute selection has received by far the most scholarly attention, perhaps because researchers tend to assume that a standard surface realiser for a given language can be used to express a set of selected properties.

Attribute selection is a complex balancing act (Reiter and Dale, 2000): we need to include enough properties that an addressee will be able to determine the target, but including all known properties of the target may be awkward or misleading. Hence a selection of properties needs to be made, and this selection should take as little time as possible. This is especially crucial in NLG for interactive settings, where a system needs to respond to a user in near real time.

2.1.2 Preferences versus adaptation in reference

Typically, a target can be distinguished using many different properties; for example, a chair can be referred to as wooden, having armrests, being Dutch, or facing right. Many REG algorithms, including Dale and Reiter's well-known Incremental Algorithm (Dale and Reiter, 1995), assume that some properties or attributes[1] are preferred, and will be selected first by the content determiner. This heuristic allows REG algorithms to ignore some potential property sets during attribute selection. It may also lead to **overspecified** referring expressions, *i.e.* those that contain more properties than are necessary to uniquely identify

[1] In this chapter, we use the term **attribute** to refer to concepts, such as size and colour, that can be used in referring expressions. When referring to an attribute-value pair, *e.g.* colour=blue, we use the term **property**. As we will see, the distinction is important, because some REG algorithms operate over attributes while others, in particular the graph-based algorithm, rank properties.

the target, but we have considerable evidence that humans also overspecify (*e.g.* Dale and Reiter (1995)). So how can we determine a preference ordering over a set of properties or attributes? Dale and Reiter stress that constructing a preference ordering is essentially an empirical question, which will differ from one domain to another, but they do point to psycholinguistic research (especially (Pechmann, 1989)) suggesting that, in general, absolute attributes (such as colour) are preferred over relative ones (such as size). After all, to determine the colour of an object we only need to look at the object itself, while to determine whether it is large or small all domain objects need to be inspected.

Even though the Incremental Algorithm is probably unique in assuming a complete preference ordering of attributes, many other REG algorithms rely on preferences as well. This became apparent, for example, during the REG Challenges (see (Gatt and Belz, 2010) for an overview); virtually all participating systems relied on training data to determine preferences of one form or another. However, relevant training data is hard to find. It has been argued that determining which properties to include in a referring expression requires a "semantically transparent" corpus (van Deemter et al., 2006): a corpus that contains the actual properties of all domain objects as well as the properties that were selected for each referring expression. Obviously, text corpora hardly ever meet this requirement. The few existing semantically transparent corpora were collected by the time-consuming exercise of asking human participants to produce referring expressions in a particular language (typically, English) for targets in controlled visual scenes for a particular domain (see *e.g.* (Gatt et al., 2007, Gorniak and Roy, 2004, Guhe and Bard, 2008, Viethen and Dale, 2006)). An important question therefore is how many human-produced references are needed to achieve a certain level of accuracy in preference ordering. One way to answer this question is by training a REG algorithm on subsets of a (semantically transparent) corpus of various sizes, and measuring the performance differences. This is precisely what we do in this chapter, in Section 2.3.

Another question is how stable preference orderings are in interactive settings, *e.g.* for applications such as spoken dialogue systems or interactive virtual characters. In these cases, it seems likely that referring expressions produced earlier in the interaction are also important. We know for instance that if one dialogue partner refers to a couch as a *sofa*, the other is more likely to use the word *sofa* as well (Branigan et al., 2010). This kind of micro-planning or **lexical entrainment** (Brennan and Clark, 1996) can be seen as a specific form of **alignment** (Pickering and Garrod, 2004) in interaction. But what if dialogue partners' preference orderings differ? Do they adapt to the other's preference ordering, or stick to their own? And what if one dialogue partner's preference ordering leads to an overspecified referring expression – will the other partner reproduce this overspecified form due to alignment? These questions are also addressed in this chapter, in Section 2.4, where we report on two experimental studies using an

interactive reference production paradigm and discuss how a REG algorithm could model our findings.

As our REG model of choice we use the graph-based algorithm, originally proposed by Krahmer, van Erk and Verleg (Krahmer et al., 2003), and described in Section 2.2. This algorithm models domain information about potential target objects in a graph structure and treats REG as a graph-search problem, where a cost function is used to prefer some solutions over others. The graph-based algorithm is a state-of-the-art REG algorithm; it was among the best scoring algorithms on attribute selection in the 2008 REG Challenge (Gatt et al., 2008), and emerged as the best performing algorithm in the most recent REG Generation Challenge (Gatt et al., 2009). In this chapter we argue that the use of cost functions makes the algorithm well-suited to deal with the trade-off between preference orderings and alignment.

2.2 Graph-based referring expression generation

2.2.1 Scene graphs

Figure 2.1 depicts an example **domain** with three potential referents or **objects**, ($\mathcal{D} = \{d_1, d_2, d_3\}$), a set of **properties** ($Prop = \{\text{chair}, \text{blue}, \text{facing-left}, \ldots\}$), and a set of **relations** ($Rel = \{\text{left-of}, \text{right-of}\}$). This domain can be modelled as the labelled directed **scene graph** shown in Figure 2.2. Properties are modelled as loops, *i.e.* edges that start and end in the same node, whereas relations are modelled as edges between nodes.

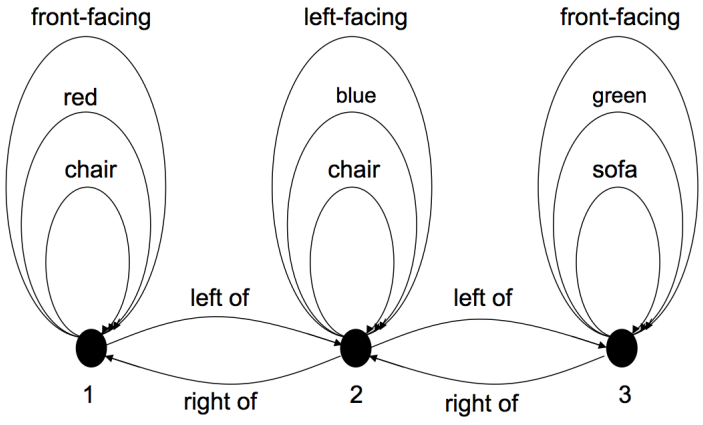

Figure 2.2: A simple scene graph

Formally, scene graphs are defined as follows. Let $\mathcal{D}$ be the **domain**, and $L = Prop \cup Rel$ the set of **labels**. Then, the scene graph $G = \langle V_G, E_G \rangle$ is a labelled directed graph, where $V_G \subseteq \mathcal{D}$ is the set of nodes or vertices (the objects)

and $E_G \subseteq V_G \times L \times V_G$ is the set of labelled directed edges (in this chapter, subscripts are omitted whenever this can be done without creating confusion).

2.2.2 Referring graphs

Now imagine that given our example domain we want to generate a **distinguishing description**, *i.e.* a referring expression that uniquely identifies d_1. We need to select a set of properties and relations that single out the **target** d_1 from the other two domain objects (the **distractors**). In the graph-based REG approach, this is done by constructing **referring graphs**. Each referring graph includes at a minimum a vertex representing the target. Referring graphs are defined in exactly the same way as scene graphs, which allows us to view REG as a graph construction exercise. Informally, a target node v in a referring graph refers to a node w in a scene graph if the referring graph can be "placed over" the scene graph in such a way that v can be placed over w, and each edge from the referring graph labelled with some property or relation can be placed over a corresponding edge in the scene graph with the same label. If there is only one way in which a referring graph can be placed over a scene graph, we have found a distinguishing description.

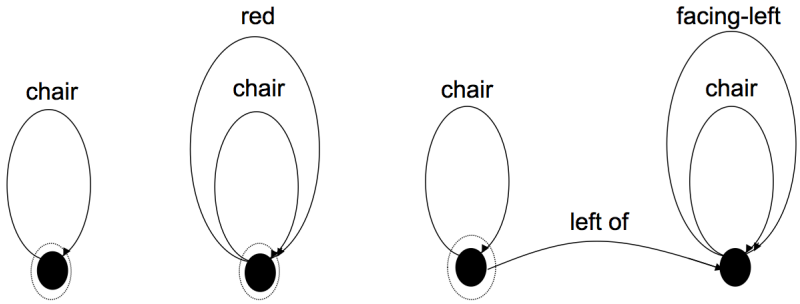

Figure 2.3: Three referring graphs

Figure 2.3 shows three potential referring graphs for d_1, with the target circled. The first, which could be realised as *the chair*, can be placed over node d_1, but also over d_2, and hence is not distinguishing. The other two, which could be realised as *the red chair* and *the chair to the left of the chair facing left* respectively, can only be placed over the scene graph in one way, and hence represent possible distinguishing descriptions for target d_1. Clearly, the second would be a more natural description for d_1 than the third; below we shall discuss how cost functions can be used to rank different descriptions such as these.

2.2.3 Formalizing reference in terms of subgraph isomorphism

Let us make the "placed over" notion a bit more precise. The informal notion of one graph being placed over another corresponds to a well-known construct in graph theory, namely subgraph-isomorphism.

A graph G' is a **subgraph** of G if and only if $V_{G'} \subseteq V_G$ and $E_{G'} \subseteq E_G$. A **subgraph isomorphism** between graphs H and G exists, if there is a subgraph G' of G such that H is isomorphic to G'. H is **isomorphic** to G' if and only if there exists a bijection $\pi : V_H \rightarrow V_{G'}$, such that for all vertices $v, w \in V_H$ and all labels $l \in L$:

$$(v, l, w) \in E_H \Leftrightarrow (\pi.v, l, \pi.w) \in E_{G'}.$$

Given a graph H and a vertex v in H, and a graph G with a vertex w in G, we will say that the pair (v, H) **refers to** the pair (w, G) if and only if (a) H is a connected graph (that is: each vertex in H has at least one edge that links it to another vertex in H), and (b) H is mapped to a subgraph of G by an isomorphism π with $\pi.v = w$. A vertex-graph pair (v, H) **uniquely refers to** (w, G) if and only if (v, H) refers to (w, G) and there is no other vertex w' in G such that (v, H) refers to (w', G).

2.2.4 Cost functions

As most REG algorithms, the graph-based algorithm requires a mechanism to give some solutions preference over others. It does so by using **cost functions**, which assign costs to the edges and nodes of a referring graph, and sums these:

$$\text{cost}(G) = \Sigma_{v \in G_V} \text{cost}(v) + \Sigma_{e \in G_E} \text{cost}(e)$$

The only a priori assumption that we make is that the cost function should be **monotonic**: extending a graph G with an edge e (notation: $G + e$) should never result in a graph which is cheaper than G. Formally,

$$\forall H \subseteq G, \forall e \in E_G : \text{cost}(H) \leq \text{cost}(H + e)$$

As we shall see below, cost functions can be defined in various ways, and this is one of the attractive properties of the graph-based REG algorithm.

2.2.5 Algorithm

Figure 2.4 contains the sketch of a basic graph-based REG algorithm, called **makeReferringExpression**. It takes as input a target v in a scene graph G. The algorithm constructs a referring graph H, which is initialized as the graph consisting of only one node: the target v. In addition, a variable *bestGraph* is

```
makeReferringExpression(v, G) {
    bestGraph := ⊥
    H := ⟨{v}, ∅⟩
    return findGraph(v, bestGraph, H, G)
}

findGraph(v, bestGraph, H, G) {
    if [bestGraph ≠ ⊥ and cost(bestGraph) ≤ cost(H)]
    then return bestGraph
    C := {v' | v' ∈ V_G & (v, H) refers to (v', G)}
    if C = {v} then return H
    for each adjacent edge e do
        I := findGraph(v, bestGraph, H + e, G)
        if [bestGraph = ⊥ or cost(I) ≤ cost(bestGraph)]
        then bestGraph := I
    rof
    return bestGraph;
}
```

Figure 2.4: Sketch of the main function (**makeReferringExpression**) and the subgraph construction function (**findGraph**), based on Krahmer et al. (2003)

introduced, for the best solution found so far. Since none have been found at this stage, *bestGraph* is initialized as the empty graph $\perp$. In the **findGraph** function the algorithm systematically tries expanding H by adding adjacent edges (i.e, edges from v, or possibly from any of the other vertices added to the referring graph H under construction). For each H the algorithm finds the set of nodes $C \subseteq G$ to which H could refer. A successful distinguishing description is found if and only if H can only refer to the target (*i.e.* $C = \{v\}$). The first distinguishing description that is found is stored in *bestGraph* (best solution found so far). At that point the algorithm only looks for referring graphs that are cheaper than the best (cheapest) solution found so far, performing a complete, depth-first search[2]. It follows from the monotonicity requirement on cost functions that the algorithm outputs the cheapest distinguishing description graph, if one exists. Otherwise it returns the empty graph.

2.2.6 Discussion

The graph-based REG algorithm has a number of attractive properties. For example, graphs are a well understood mathematical formalism, and there are

[2] Naturally, graph-based generation is compatible with different search strategies as well.

many efficient algorithms for dealing with graph structures (see for instance (Chartrand and Oellermann, 1993, Gibbons, 1985)). In addition, because relational properties are handled in the same way as other properties (namely as edges in a graph), the treatment of relations between objects does not suffer from some of the problems of earlier REG approaches; there is, for instance, no need to make any *ad hoc* stipulations (*e.g.* that a property can only be attributed to a given object once per referring expression (Dale and Haddock, 1991)). Relational properties cause testing for a subgraph isomorphism to have exponential complexity (Garey and Johnson, 1979), but special cases are known in which the problem has lower complexity, for example when considering graphs that are **planar** (that is, drawable without crossing edges). Krahmer et al. (2003) sketch a greedy algorithm that can simplify any graph into a planar equivalent.

Variations and extensions

Van Deemter and Krahmer 2007 discuss how the basic graph-based algorithm can be extended with, for example, plurals (*e.g. the chairs*) and boolean expressions (*e.g. the blue chairs and the coach that is not red*).

Van der Sluis and Krahmer 2007 present an extension of the algorithm which is capable of generating multimodal referring expressions (*e.g. that chair* accompanied by a pointing gesture). Their account allows for pointing degrees of various precisions, where less precise pointing gestures are accompanied by more extensive verbal descriptions (see de Ruiter et al. (2012) for a discussion).

The approach to graphs outlined here offers an attractive account of binary relations, but not of more complex relations (*e.g. the chair given by John to Mary, the fan in between the chair and the couch*). Croitoru and van Deemter (2007) offer an alternative way of constructing graphs, using insights from conceptual graph theory, which can account for relations of arbitrary complexity. Their approach also allows for a logic-based interpretation of reference graphs, which enables more complex knowledge representation and reasoning.

Computing costs

The use of cost functions is an important ingredient of the graph-based algorithm. Various alternative ways of computing costs have been considered. Perhaps the most straightforward option is to assign a cost of 1 to each edge and vertex. It is easily seen that in this way the cheapest distinguishing graph will also be the smallest one; this would make the graph-based algorithm equivalent (in terms of its input-output behaviour) to the well-known Full Brevity algorithm (Dale, 1989). The Full Brevity algorithm has been criticised, however, for lacking "humanlikeness", since human speakers frequently produce overspecified referring expressions (Arts, 2004, Engelhardt et al., 2006, Olson, 1970, Pechmann, 1989, Sonnenschein, 1984). The graph-based algorithm can model overspecifica-

tion by allowing some properties to be included for free (Viethen et al., 2008)[3]. However, if a graph contains zero-cost edges, the order in which the graph-based algorithm tries to add properties to a referring expression must be explicitly controlled, to ensure that "free" distinguishing properties are included (Viethen et al., 2008).

Which properties should be available for free inclusion? One option is to link cost to frequency; properties that speakers use often should be cheap, while those that are less frequent should be more expensive. In other words, if we assume that $p(e)$ is the probability that an edge (property) will be used in a referring expression, then we could define $\text{cost}(e) = -\log(p(e))$. Properties that are very cheap (costs below a certain threshold) can manually be set to 0. Explicit control of property inclusion order can also be tied to frequency; edges can be tried in order of their corpus frequency.

The probabilities of properties can be estimated by counting frequencies of occurrence in a semantically transparent corpus such as the TUNA corpus (Gatt et al., 2007, van Deemter et al., 2012). The TUNA corpus consists of two domains: one containing pictures of people (all famous mathematicians), the other containing furniture items in different colours depicted from different orientations, such as those in Figure 2.1[4]. Viethen et al. (2008) compare a cost function where costs are directly derived from frequencies in the TUNA corpus (in terms of log probabilities) with a "Free Naïve" one that just assigns three costs based on the frequencies (with 0 = free, 1 = cheap, 2 = expensive), and found that the latter results in more human-like referring expressions. The graph-based algorithm with the Free Naïve cost function, combined with a dedicated linguistic realiser, was the best performing REG algorithm in the 2009 REG Challenge (Gatt et al., 2009, Krahmer et al., 2008): the referring expressions this algorithm produced were overall most similar to the referring expressions in the test set, were subjectively judged to be most adequate and most fluent, and resulted in the highest identification accuracies (ability of human readers to identify the target referent given the generated referring expression). Although it is still a matter of some debate how REG algorithms should best be evaluated (see Chapter **??**) and the test data used in the 2009 REG Challenge contains some idiosyncratic referring expressions (Gatt and Belz, 2010), it is clear that the graph-based REG algorithm produces results that are state-of-the-art, and that the cost function, giving preference to some properties over others, plays an important part in this. In the next section we will discuss in more detail how these cost functions can be determined automatically, and how much data is required to obtain accurate cost functions.

[3] Notice that this does not violate the monotonicity assumption.

[4] The pictures of furniture items were taken from the Object Databank, developed by Michael Tarr at Carnegie Mellon University and freely distributed at `www.tarrlab.org`.

2.3 Determining preferences and computing costs

In the work of Krahmer et al. (2008) and Viethen et al. (2008), cost functions for properties for the two TUNA domains were determined in two ways, by: (a) directly using frequencies from the TUNA corpus of about 450 human-produced referring expressions; and (b) manually clustering properties into cost clusters based on the corpus frequencies. This raises two related questions: how can we achieve an optimal cost clustering, and how much training data is necessary for accurate cost function estimation? In this section we address these two questions.

Mapping frequencies to cost clusters

One way to identify an optimal clustering of frequencies is to systematically compare the performance of cost functions derived from various clusterings on a held out test set. We use k-means clustering (Hartigan and Wong, 1979), which partitions n data points into k clusters (S_1 to S_k), with $k \leq n$ by assigning each point to the cluster with the nearest mean. The total intra-cluster variance V is minimised using the function

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} (x_j - \mu_i)^2$$

where μ_i is the centroid of all the points $x_j \in S_i$. In our case, the points n were properties, and μ_i is the average frequency of the properties in S_i. The cluster-based costs were defined as follows:

$$\forall x_j \in S_i, \text{cost}(x_j) = i - 1$$

where S_1 is the cluster with the most frequent properties, S_2 is the cluster with the next most frequent properties, and so on. Using this approach, properties from cluster S_1 get cost 0 and thus can be added for free to a referring expression.

The training and test data on which we performed our experiment were taken from the TUNA corpus (Gatt et al., 2007, van Deemter et al., 2012). For training, we used the -LOC data from the REG Challenge 2009 training data (Gatt et al., 2009): 165 Furniture referring expressions and 136 People referring expressions[5]. For testing, we used the -LOC data from the TUNA 2009 development set: 38 Furniture referring expressions and 38 People referring expressions.

We clustered the training data repeatedly using $k = 2$ to $k = 6$. Then we evaluated the performance of the graph-based algorithm with the resulting cost functions on the TUNA 2009 development data. We used two metrics: Dice (overlap between sets of properties) and Accuracy (perfect match between sets of

[5] The -LOC data was collected by explicitly instructing participants not to use locative information (*e.g. in the top left corner*) when referring to targets in the grid.

Table 2.1: Cost clustering results for 2-means costs and the Free Naïve costs of Krahmer et al. (2008)

	Furniture		**People**	
Costs	**Dice**	**Acc.**	**Dice**	**Acc.**
k-means	0.810	0.50	0.733	0.29
Free Naïve	0.829	0.55	0.733	0.29

properties) as evaluation metrics. For comparison, we also ran the graph-based algorithm on this data set using the Free Naïve cost function of Viethen et al. (2008). In all our tests, we used decreasing frequency for explicit control of property inclusion order to ensure that free properties would be considered, *i.e.* the algorithm always examined more frequent properties first.

The best results were achieved with $k = 2$, for both TUNA domains. Interestingly, this is the coarsest possible k-means function: with only two costs (0 and 1) it is even less fine-grained than the Free Naïve cost functions. The results for the k-means costs with $k = 2$ and the Free Naïve costs of (Krahmer et al., 2008) are shown in Table 2.1. A repeated measures analysis of variance (ANOVA) on the Dice and Accuracy scores, using cost function as a within-subjects variable (with levels Free Naïve and 2-means) revealed no statistically significant differences between the two cost functions. This suggests that k-means clustering offers a good and systematic alternative to manual clustering of frequency-based costs.

Varying training set size

To find out how much training data is required to achieve an acceptable property selection performance, we derived cost functions and preference orderings from different sized training sets, and evaluated them on our test data.

For training data, we used randomly selected subsets of the -LOC data from the REG Challenge 2009 training data (Gatt et al., 2009), with set sizes of 1, 5, 10, 20 and 30 items. Because the accidental composition of a training set may strongly influence the results, we created 5 different sets of each size. The training sets were built up in a cumulative fashion: we started with five sets of size 1, then added 4 items to each of them to create five sets of size 5, etc. This resulted in five series of increasingly sized training sets. As before, for testing we used the -LOC data from the TUNA 2009 development set and the Dice and Accuracy metrics.

We derived cost functions (using k-means clustering with $k = 2$) and preference orderings for each of the training sets, following the method outlined earlier in this chapter. In doing so, we had to deal with missing data: not all properties were present in all data sets. This problem mostly affected the smaller training sets. By set size 10 only a few properties were missing, while by set size 20, all properties were present in all sets. For the cost functions, we simply assigned

the highest cost (1) to the missing properties. For the sake of comparison, we listed properties with the same frequency (0 for missing properties) always in alphabetical order.

To determine significance, we calculated the means of the scores of the five training sets for each set size, so that we could compare them with the scores of the entire set. We applied repeated measures analyses of variance (ANOVA) to the Dice and Accuracy scores, using set size (1, 5, 10, 20, 30, entire set) as a within-subjects variable. The mean results for the different training set sizes are shown in Table 2.2. The general pattern is that the scores increase with the size of the training set, but the increase gets smaller as the set sizes become larger.

Table 2.2: Mean results for different training set sizes

	Furniture		**People**	
Set size	**Dice**	**Acc.**	**Dice**	**Acc.**
1	0.693	0.25	0.560	0.13
5	0.756	0.34	0.620	0.15
10	0.777	0.40	0.686	0.20
20	0.788	0.41	0.719	0.25
30	0.782	0.41	0.718	0.27
Entire set	0.810	0.50	0.733	0.29

In the Furniture domain, we found a main effect of set size (Dice: $F_{(5,185)} = 7.209$, $p < .001$; Accuracy: $F_{(5,185)} = 6.140$, $p < .001$). To see which set sizes performed differently as compared to the entire set, we conducted Tukey's HSD post hoc comparisons. For Dice, the scores of set size 10 ($p = .141$), set size 20 ($p = .353$), and set size 30 ($p = .197$) did not differ significantly from the scores of the entire set of 165 items. The Accuracy scores show a slightly different pattern: the scores of the entire training set were still significantly higher than those of set size 30 ($p < .05$).

In the People domain we also found a main effect of set size (Dice: $F_{(5,185)} = 21.359$, $p < .001$; Accuracy: $F_{(5,185)} = 8.074$, $p < .001$). Post hoc pairwise comparisons showed that the scores of set size 20 (Dice: $p = .416$; Accuracy: $p = .146$) and set size 30 (Dice: $p = .238$; Accuracy: $p = .324$) did not significantly differ from the performance of the full set of 136 items.

The results suggest that small data sets can be sufficient for training the graph-based REG algorithm. However, domain differences play a role as well in how much training data is needed: using Dice as an evaluation metric, training sets of 10 sufficed in the relatively simple Furniture domain, while in the People domain it took a set size of 20 to achieve similar results as when using the full training set. Using the full training sets does give numerically higher scores, but the differences were not statistically significant. Furthermore, the accidental composition of the training sets may strongly influence the attribute selection performance. In the Furniture domain, there were clear differences between the results of specific training sets, with "bad sets" pulling the overall performance

down. This affected Accuracy but not Dice, perhaps because the latter is a less strict metric.

Of course, Dice and Accuracy are not the only evaluation metrics. It would be particularly interesting to see how the use of small training sets affects effectiveness and efficiency of target identification by human subjects; as shown by Belz and Gatt (Belz and Gatt, 2008), task-performance measures do not necessarily correlate with similarity measures such as Dice (although the graph-based algorithm scored well on both dimensions).

It is interesting to see which preferences were learned using the graph-based algorithm with corpus-based cost functions. If we focus on attributes, we find that generally colour is preferred and orientation and size less so (in the Furniture domain), while having glasses is highly preferred, and for example, wearing a tie or a suit is not (in the People domain). Colour and glasses are good examples of attributes that can be added for free. It also interesting to observe that orientation is far less dispreferred than wearing a tie; in fact, hasTie = 1 is *never* used in the TUNA data. Many of these distinctions can already be observed in set sizes as small as 5.

This is the situation when we look at the level of attributes. The graph-based REG algorithm, however, operates with preferences on the level of properties (i.e, attribute-value combinations). The potential advantage of this is that it becomes possible to prefer some colours (*e.g.* red) and disprefer other (mauve, taupe); the intuition is that it may be simpler to describe a mauve chair in terms of its size (certainly when assuming that the addressee may not know what colour mauve is). Indeed, if we look at the preferences that were learned from the data, we see that for instance having glasses (hasGlasses = 1) is strongly preferred (costs 0), while not having glasses (hasGlasses = 0) is not (costs 1). Of course, it can be conjectured that learning preferred attributes will require less data (fewer referring expressions) than learning preferred properties.

So far, we have been working on the assumption that some properties are preferred over others, and we have just shown that a limited set of referring expressions may be enough to determine these preferences. However, is this basic assumption tenable when we consider the production of referring expressions in interaction? Unfortunately, data for this has been lacking. We now describe two experiments looking at the relation between adaptation and interaction.

2.4 Adaptation and interaction

In this section we report on two experiments that study the trade-off between domain-dependent preferences and adaptation to referring expressions that have been used earlier in an interaction. Experiment I studies what speakers do when referring to a target that can be distinguished in a preferred (*e.g. the red chair*) or a dispreferred way (*e.g. the left-facing chair*), when earlier in the interaction either the preferred or the dispreferred variant was **primed**, or used by a dialogue

partner. Experiment II studies overspecification, where participants were again asked to refer to a target, which can be distinguished using a minimal referring expression containing only including a preferred attribute (*e.g. the red chair*), while earlier overspecified references (*e.g. the red front-facing chair*) were primed. Both studies use a novel interactive reference production paradigm, applied to the Furniture and People domains of the TUNA corpus, to see to what extent adaptation may be domain dependent.

2.4.1 Experiment I: adaptation and attribute selection

This experiment studies whether and how adaptation influences attribute selection in REG in interactive settings.

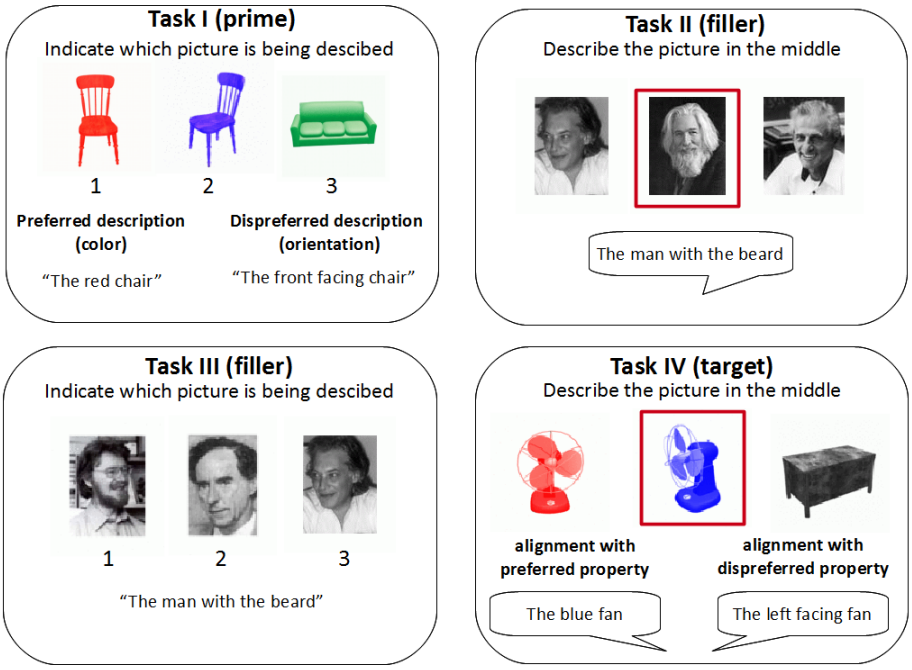

Figure 2.5: The 4 turns that constitute a trial in experiment I. This figure shows a Furniture trial; People trials have an identical structure.

Method

Participants Participants were 26 native speakers of Dutch (two males, mean age 20 years, 11 months) who participated in the experiment in exchange for partial course credit.

Materials The stimulus pictures were taken from the TUNA corpus (Gatt, 2007). We relied on a Dutch version of the TUNA corpus (Koolen et al., 2009) to determine which properties our participants would prefer and disprefer (*i.e.* which properties they use frequently and infrequently). It was found that Dutch speakers, like English ones, have a preference for colour in the Furniture domain and wearing glasses in the People domain, and disprefer orientation of a furniture piece and wearing a tie, respectively. These properties were used as primes.

Procedure Each experimental trial consisted of four turns in an interactive reference understanding and production experiment: a prime, two fillers and the experimental referring expression (see Figure 2.5 for an example of an experimental trial). In each trial, the prime and final turns were from one domain (Furniture or People), while the filler turns were from the other domain. The two filler turns were intended to prevent a too direct connection between the prime and the target. In the prime, the participant listened to a referring expression pre-recorded by a female voice and had to select a referent from three possibilities in the trial domain. In this turn, referring expressions used either a preferred or a dispreferred property; each property alone would be enough to uniquely identify the referent. In the two filler turns, the participant him/herself first produced a referring expression for a target given three objects in the other domain, and then had to select, from three possibilities in the other domain, the referent for a spoken referring expression. Finally, the participant produced a referring expression for a target object in the trial domain, which could always be distinguished from its two distractors using a preferred (*e.g. The blue fan*) or a dispreferred (*e.g. The left-facing fan*) property. Note that *attributes* were primed, not properties; a participant may have heard *front-facing* in the prime turn, while the target referent had a different value for the orientation attribute in the experimental turn (as in Figure 2.5). In addition, in the Furniture domain but not in the People domain, the type values could differ; for example, when primed with a (preferred or dispreferred) referring expression for a chair, participants did not necessarily have to describe a chair in the experimental turn.

For each domain, there were 20 preferred and 20 dispreferred trials, resulting in 2 x (20 + 20) = 80 critical trials. These were presented in counter-balanced blocks, and within blocks each participant received a different random order. In addition, there were 80 filler trials (each following the same structure as outlined in Figure 2.5); filler trials never involved the attributes of interest. During debriefing, none of the participants indicated they had been aware of the experiment's true purpose.

Results and discussion

The proportions of preferred and dispreferred attributes used by participants as a function of prime and domain are shown in Figure 2.6. The black bars

indicate use of the preferred attribute and the white bars indicate use of the dispreferred attribute. In both domains, the preferred attribute is used more frequently than the dispreferred attribute with the preferred prime, which serves as a manipulation check (our participants indeed overall preferred the preferred attributes to the dispreferred ones). The results show a clear effect of prime for the Furniture domain: participants used the preferred attribute (colour, as in *the red fan*) more when they were primed with it, and the dispreferred attribute (orientation, as in *the fan seen from the front*) more when it was in the prime. The results for the People domain reveal a similar picture (when participants were primed with the dispreferred attribute, they used it more often), but much less pronounced.

For our statistical analysis, we use the proportion of attribute **alignment** as the dependent measure. Alignment occurs when a participant uses the same attribute in the target as occurred in the prime. Table 2.3 displays the alignment mean and standard deviation per prime (preferred versus dispreferred) for the Furniture and People domains. We conducted a 2 x 2 repeated measures analysis of variance (ANOVA) with alignment as the dependent variable, and domain (Furniture versus People) and prime (preferred versus dispreferred) as independent variables. A statistically significant main effect was found for prime ($F(1, 25) = 6.43, p < .05$), showing that the prime influenced the selection of attributes in the experimental turn: when primed with dispreferred attributes, our participants used the dispreferred attributes more often than when they were primed with preferred attributes. A statistically significant main effect was found for domain ($F(1, 25) = 10.88, p < .01$), confirming that there is significantly more alignment in the Furniture domain. Finally, a statistically significant interaction was found ($F(1, 25) = 5.74, p < .05$), confirming our observation that the effect of the prime was less pronounced in the People domain. Interestingly, this is very much in line with the observations made in the previous section, where we saw that orientation is less dispreferred than wearing a tie.

Table 2.3: Alignment means (and standard deviations) as a function of domain (Furniture and People) and prime (preferred and dispreferred)

Domain	**Prime**	**Alignment mean (SD)**
Furniture	Preferred	0.89 (0.32)
	Dispreferred	0.60 (0.49)
People	Preferred	0.97 (0.16)
	Dispreferred	0.25 (0.43)

2.4.2 Experiment II: Adaptation and overspecification

This experiment looks at overspecification: participants were primed with overspecified referring expressions that included both preferred and dispreferred attributes, and were then asked to produce a referring expression for a target

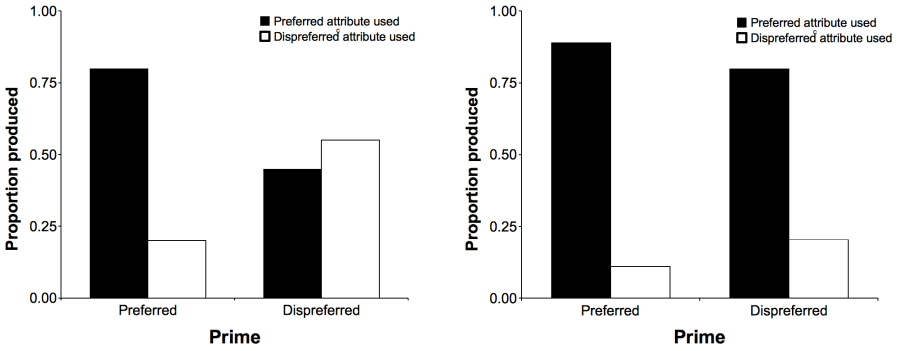

Figure 2.6: Proportions of preferred and dispreferred attributes in the Furniture (left) and People (right) domains

which could be distinguished using a minimal referring expression including only a preferred attribute.

Method

Participants Particpants were 28 native speakers of Dutch (8 males, mean age 20 years, six months) who participated in exchange for partial course credit. None had participated in experiment I.

Materials and procedure The materials and procedure were identical to those in experiment I, with the exception of the referring expressions in the prime turn (see Figure 2.5). In experiment II, the referring expressions in the prime turn were always overspecified. Thus, in the Furniture domain participants heard referring expressions such as *the red chair seen from the front* and in the People domain they heard referring expressions such as *the man with the glasses and the tie*. All these referring expressions were overspecified in that they use two attributes (in addition to the type attribute), including a preferred and a dispreferred one, while either attribute would be sufficient to uniquely identify the referent. All referring expressions in the prime turns were produced by the same speaker as in experiment I.

Results and discussion

Figure 2.7 and Table 2.4 show the proportions of overspecified references in experiment I ("single prime") and experiment II ("dual prime") for both domains. A referring expression was considered overspecified when both the preferred and the dispreferred attribute were used. The results show that when participants were primed with both the preferred and the dispreferred attribute, 52% of the

Furniture trials and 57% of the People trials were produced with both attributes, even though the preferred attribute would be sufficient to distinguish the target. By contrast, in experiment I speakers produced overspecified referring expressions in only 11% to 15% of the experimental turns.

To analyse these results, we combined the data for experiment I (single prime) and experiment II (dual prime) and conducted a mixed effects ANOVA with proportion of overspecification as the dependent variable, domain (Furniture versus People) as a within-subjects variable, and prime (single prime versus dual prime) as a between-subjects variable. The results show a statistically significant effect for prime ($F(1, 52) = 32.50, p < 0.001$); the dual primes result in more overspecified referring expressions (and thus a more frequent use of the dispreferred property) than the single primes. There was no statistically significant effect for domain, and no statistically significant interaction between domain and prime.

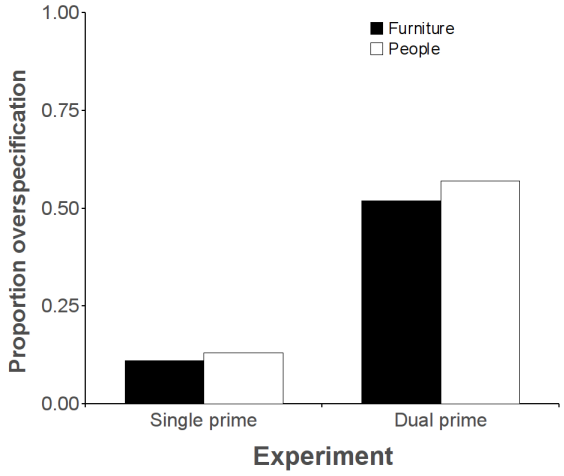

Figure 2.7: Proportions of overspecification with single (*the chair seen from the front / the man with the tie*) and dual primes (*the blue chair seen from the front / the man with the glasses and the tie*) in the People and Furniture domains

Table 2.4: Overspecification means (and standard deviations) for experiment I (selection) and experiment II (overspecification) per domain (Furniture and People) and prime (preferred and dispreferred).

	Experiment I		**Experiment II**
	Preferred	**Dispreferred**	
Furniture	0.13 (0.34)	0.11 (0.31)	0.52 (0.37)
People	0.15 (0.36)	0.13 (0.33)	0.57 (0.34)

2.5 General Discussion

In this chapter we showed that REG in interactive settings is a balancing act between relatively stable domain-dependent preferences and relatively dynamic interactive factors. We first asked how much data is required to determine the preference ordering for a domain. Our experiment in Section 2.3 showed that with 20 or fewer training instances, acceptable attribute selection results can be achieved; that is, results that do not differ significantly from those obtained using many more training instances. This is good news, because collecting such small amounts of training data should not take too much time and effort, making it relatively easy to apply the graph-based REG algorithm for new domains and languages. Next, we examined on the relation between preferences and adaptation, describing two experiments in Section 2.4. Experiment I (looking at attribute selection) showed that speakers were more likely to include a (preferred or dispreferred) attribute in a referring expression when this attribute was primed. Experiment II (looking at referential overspecification) revealed that alignment and overspecification are closely related. While some participants were reluctant to select a dispreferred attribute in experiment I, participants in experiment II aligned frequently with an overspecified referring expression that contained *both* a preferred and a dispreferred attribute, even though including only the preferred one would have been sufficient to produce a distinguishing description.

It could be argued that the interactive nature of our experimental paradigm in Section 2.4 is limited, in that participants did not truly interact with the speaker of the referring expressions they had to comprehend. Rather, participants interacted with an imaginary dialogue partner, which allowed us to guarantee that all participants were primed in exactly the same way. Using an imaginary audience is a standard experimental procedure to study interactive communication, and recent studies have shown that the differences between a real audience and an imagined audience are small (Van Der Wege, 2009, Ferreira et al., 2005).

It is also worth emphasizing that our experimental results in Section 2.4 cannot readily be explained in terms of well-understood phenomena such as lexical or syntactic alignment. In experiment I, what is primed are not lexical items, but attributes. A prime in the Furniture domain may be *the front-facing chair*, where *front-facing* is the relevant value of the orientation attribute, while in the experimental turn participants should produce a referent for, say, a fan whose orientation is to the left. Arguably, what is being primed is a way to look at an object, thereby making certain attributes of the object more salient.

The graph-based algorithm, as described in Section 2.2, does not yet capture the alignment effects we found in experiments I and II (and the same applies to other state-of-the-art REG algorithms such as the Incremental Algorithm (Dale and Reiter, 1995)). The graph-based algorithm, as it stands, predicts that a dispreferred property would never be used if a preferred property would be sufficient to uniquely characterise a target. And while the algorithm can account

for some amount of overspecification (by allowing some properties to be included for free), it would never redundantly use a dispreferred (expensive) property, even though our participants did this in over half of the cases in experiment II.

What is missing in the algorithm is a sensitivity to the references produced during an earlier interaction. In fact, the use of cost functions offers an elegant way to account for this. In interactive settings, costs could be thought of as a composition of relatively stable domain-dependent preferences (as formalized above) combined with relatively dynamic costs, modelling the activation of properties in the preceding interaction. The latter costs can be derived from Buschmeier et al. (2010) who study alignment in micro-planning. Inspired by Buschmeier and colleagues, we can make the costs of an attribute a much (cheaper) when it is mentioned repeatedly, after which the costs gradually increase as the activation of a decays. The net result is that dispreferred properties become relatively cheap when they have been used in the previous interaction, and hence are more likely to be selected by the graph-based algorithm.

Gatt et al. (2011) go one step further than we have here, proposing a new model for alignment in referring expression production that integrates alignment and preference-based attribute selection. This model consists of two parallel processes: a preference-based search process, and an alignment-based process. These two processes run concurrently and compete to contribute attributes to a limited capacity working memory buffer that will produce the referring expression. This model was tested against the data of experiment II and showed a similar amount of overspecification as the human participants produced.

Use of the graph-based algorithm in an interactive setting has a number of other theoretical and practical advantages. First of all, alignment may reduce the search space for the algorithm; not all alternatives need to be explored, because the search process can be driven by the edges that were used previously in the interaction. In addition, as we have seen, preference orders need to be empirically determined for each new domain. But what to do when your REG algorithm is applied in a domain for which the preference order is unknown? Our experiments suggest that a good strategy might be to simply model alignment.

Various studies have demonstrated both the existence of and the benefits of alignment in human-computer interaction (for a recent survey, see Branigan et al. (2010)). Branigan et al. (2010) argue that a lot is to be gained from computers that align: "Speakers should also feel more positive affect when interacting with a computer that aligns with them than with one that does not."

The approach outlined in this chapter is limited to generation of distinguishing descriptions, which have identification of a target as their main function. Even though this kind of referring expression has received the most attention in the REG literature, they are certainly not the only kind of referring expression that occur. Various studies (Di Eugenio et al., 2000, Gupta and Stent, 2005, Jordan and Walker, 2005, Passonneau, 1996) confirm that references in interactive settings may serve other functions besides identification. The Coconut corpus (Di

Eugenio et al., 2000), for example, is a set of task-oriented dialogues in which participants negotiate the furniture items they want to buy on a fixed, shared budget. Referring expressions in this corpus (*e.g. a yellow rug for 150 dollars*) not only identify a particular piece of furniture, but also include properties that directly pertain to the task (*e.g.* the amount of money that is still available and the state of agreement between the negotiators). More recently, other researchers have started exploring the generation of referring expressions in interactive settings as well. Stoia et al. (2006), for example, presented a system that generates referring expressions in situated dialogue, taking into account both the dialogue history and the visual context (defined in terms of which distractors are in the current field of vision of the speakers and how distant they are from the target). Janarthanam and Lemon (2010) present an REG algorithm that automatically adapts to the expertise level of the intended addressee (for example, using *the router* when communicating with an expert user, and *the black block with the lights* when communicating with a novice). These lines of research fit in well with another, more general, strand of research concentrating on choice optimisation during NLG based on user data (Walker et al., 2007, White et al., 2009).

2.6 Conclusion

When speakers want to identify a target, such as a chair in a furniture shop, using a distinguishing description, they tend to prefer certain properties over others. We have shown that only a limited number of (semantically transparent) example descriptions is required to be able to determine these preferences, although this also depends on the size and complexity of the domain. In interactive settings, however, the generation of distinguishing descriptions not only depends on preferences, but also on the descriptions that were produced earlier in the interaction, as we have shown in two experiments, one dedicated to attribute selection and the other to overspecification. We argue that the graph-based REG algorithm is a suitable candidate to model this balancing act, since its use of cost functions enables us to weigh the different factors in a dynamic way.

Acknowledgments

Emiel Krahmer and Martijn Goudbeek thank The Netherlands Organisation for Scientific Research (NWO) for the VICI grant "Bridging the Gap between Computational Linguistics and Psycholinguistics: The Case of Referring Expressions" (277-70-007). The research described in Section 2.3 is based on (Theune et al., 2011); for more information on the research presented in Section 2.4 we refer the reader to (Goudbeek and Krahmer, 2010). We would like to thank Ruud Koolen and Sander Wubben for their help with the preferences experiment (Section 2.3) and Albert Gatt for discussions on REG and adaptation.

References

Arts, A. (2004). *Overspecification in Instructive Texts*. PhD thesis, Tilburg University.

Belz, A. and Gatt, A. (2008). Intrinsic vs. extrinsic evaluation measures for referring expression generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (henceforth 'ACL')*.

Branigan, H. P., Pickering, M. J., Pearson, J., and McLean, J. F. (2010). Linguistic alignment between people and computers. *Journal of Pragmatics*, 42(9):2355–2368.

Brennan, S. E. and Clark, H. H. (1996). Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology*, 22(6):1482–1493.

Buschmeier, H., Bergmann, K., , and Kopp, S. (2010). Modelling and evaluation of lexical and syntactic alignment with a priming-based microplanner. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (henceforth 'EMNLP')*.

Chartrand, G. and Oellermann, O. R. (1993). *Applied and Algorithmic Graph Theory*. McGraw-Hill, New York, NY.

Croitoru, M. and van Deemter, K. (2007). A conceptual graph approach to the generation of referring expressions. In *Proceedings of the International Joint Conference on Artificial Intelligence (henceforth 'IJCAI')*.

Dale, R. (1989). Cooking up referring expressions. In *Proceedings of ACL*.

Dale, R. and Haddock, N. (1991). Content determination in the generation of referring expressions. *Computational Intelligence*, 7(4):252–265.

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

de Ruiter, J. P., Bangerter, A., , and Dings, P. (2012). The interplay between gesture and speech in the production of referring expressions: Investigating the tradeoff hypothesis. *Topics in Cognitive Science*, 4(2):232–248.

Di Eugenio, B., Jordan, P. W., Thomason, R. H., and Moore, J. D. (2000). The agreement process: An empirical investigation of human-human computer-mediated collaborative dialogue. *International Journal of Human-Computer Studies*, 53(6):1017–1076.

Engelhardt, P. E., Bailey, K. G. D., and Ferreira, F. (2006). Do speakers and listeners observe the Gricean maxim of quantity? *Journal of Memory and*

Language, 54(4):554–573.

Ferreira, V. S., Slevc, L. R., and Rogers, E. S. (2005). How do speakers avoid ambiguous linguistic expressions? *Cognition*, 96(3):263–284.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, NY.

Gatt, A. (2007). *Generating coherent references to multiple entities*. PhD thesis, Department of Computing Science, University of Aberdeen.

Gatt, A. and Belz, A. (2010). Introducing shared tasks to NLG: the TUNA shared task evaluation challenges. In Krahmer, E. and Theune, M., editors, *Empirical methods in natural language generation*, pages 264–293. Springer-Verlag, Berlin, Heidelberg.

Gatt, A., Belz, A., and Kow, E. (2008). The TUNA challenge 2008: Overview and evaluation results. In *Proceedings of the International Workshop on Natural Language Generation (henceforth 'INLG')*.

Gatt, A., Belz, A., and Kow, E. (2009). The TUNA-REG challenge 2009: overview and evaluation results. In *Proceedings of the European Workshop on Natural Language Generation (henceforth 'ENLG')*.

Gatt, A., Goudbeek, M., and Krahmer, E. (2011). Attribute preference and priming in reference production: Experimental evidence and computational modelling. In *Proceedings of the Annual Conference of the Cognitive Science Society (henceforth 'CogSci')* .

Gatt, A., van der Sluis, I., and van Deemter, K. (2007). Evaluating algorithms for the generation of referring expressions using a balanced corpus. In *Proceedings of ENLG*.

Gibbons, A. M. (1985). *Algorithmic Graph Theory*. Cambridge University Press, Cambridge, UK.

Gorniak, P. J. and Roy, D. (2004). Grounded semantic composition for visual scenes. *Journal of Artificial Intelligence Research*, 21:429–470.

Goudbeek, M. and Krahmer, E. (2010). Preferences versus adaptation during referring expression generation. In *Proceedings of ACL*.

Guhe, M. and Bard, E. G. (2008). Adapting referring expressions to the task environment. In *Proceedings of CogSci*.

Gupta, S. and Stent, A. (2005). Automatic evaluation of referring expression generation using corpora. In *Proceedings of the Workshop on Using Corpora for Natural Language Generation*.

Hartigan, J. and Wong, M. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.

Janarthanam, S. and Lemon, O. (2010). Learning adaptive referring expression generation policies for spoken dialogue systems. In *Proceedings of EMNLP*.

Jordan, P. W. and Walker, M. A. (2005). Learning content selection rules for generating object descriptions in dialogue. *Journal of Artificial Intelligence Research*, 24(1):157–194.

Koolen, R., Gatt, A., Goudbeek, M., and Krahmer, E. (2009). Need i say more? on factors causing referential overspecification. In *Proceedings of the CogSci workshop on the Production of Referring Expressions*.

Krahmer, E., Theune, M., Viethen, J., and Hendrickx, I. (2008). GRAPH: the costs of redundancy in referring expressions. In *Proceedings of the International Natural Language Generation Conference (henceforth 'INLG')*.

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

Matthews, D., Butcher, J., Lieven, E., and Tomasello, M. (2012). Two- and four-year-olds learn to adapt referring expressions to context: Effects of distracters and feedback on referential communication. *Topics in Cognitive Science*, 4(2):184–210.

Mellish, C., Scott, D., Cahill, L., Paiva, D., Evans, R., and Reape, M. (2006). A reference architecture for natural language generation systems. *Natural Language Engineering*, 12(1):1–34.

Olson, D. R. (1970). Language and thought: Aspects of a cognitive theory of semantics. *Psychological Review*, 77(4):257–273.

Passonneau, R. (1996). Using centering to relax Gricean informational constraints on discourse anaphoric noun phrases. *Language and Speech*, 39(2–3):229–264.

Pechmann, T. (1989). Incremental speech production and referential overspecification. *Linguistics*, 27(1):89–110.

Pickering, M. and Garrod, S. (2004). Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, 27(2):169–226.

Reiter, E. and Dale, R. (2000). *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge, UK.

Sonnenschein, S. (1984). The effect of redundant communication on listeners: Why different types may have different effects. *Journal of Psycholinguistic Research*, 13(2):147–166.

Stoia, L., Shockley, D. M., Byron, D. K., and Fosler-Lussier, E. (2006). Noun phrase generation for situated dialogs. In *Proceedings of INLG*.

Theune, M., Koolen, R., Krahmer, E., and Wubben, S. (2011). Does size matter: how much data is required to train a REG algorithm? In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (henceforth 'ACL-HLT')*.

van Deemter, K., Gatt, A., van der Sluis, I., and Power, R. (2012). Generation of referring expressions: Assessing the incremental algorithm. *Cognitive Science*, 36.

van Deemter, K. and Krahmer, E. (2007). Graphs and booleans: On the generation of referring expressions. In Bunt, H. and Muskens, R., editors, *Computing Meaning*, volume 3, pages 397–422. Springer, Dordrecht, The Netherlands.

van Deemter, K., van der Sluis, I., and Gatt, A. (2006). Building a semantically transparent corpus for the generation of referring expressions. In *Proceedings*

of INLG.

van der Sluis, I. and Krahmer, E. (2007). Generating multimodal referring expressions. *Discourse Processes*, 44(3):145–174.

Van Der Wege, M. (2009). Lexical entrainment and lexical differentiation in reference phrase choice. *Journal of Memory and Language*, 60(4):448–463.

Viethen, J. and Dale, R. (2006). Algorithms for generating referring expressions: do they do what people do? In *Proceedings of INLG*.

Viethen, J., Dale, R., Krahmer, E., Theune, M., and Touset, P. (2008). Controlling redundancy in referring expressions. In *Proceedings of the International Conference on Language Resources and Evaluation (henceforth 'LREC')*.

Walker, M. A., Stent, A., Mairesse, F., and Prasad, R. (2007). Individual and domain adaptation in sentence planning for dialogue. *Journal of Artificial Intelligence Research*, 30(1):413–456.

White, M., Rajkumar, R., Ito, K., and Speer, S. R. (2009). Eye tracking for the online evaluation of prosody in speech synthesis: Not so fast! In *Proceedings of the International Conference on Spoken Language Processing (henceforth 'ICSLP')*.