

Capturing Acceptable Variation in Distinguishing Descriptions

Jette Viethen and Robert Dale

Centre for Language Technology

Macquarie University, Sydney, Australia

{jviethen|rdale}@ics.mq.edu.au

Abstract

Almost all existing referring expression generation algorithms aim to find one best referring expression for a given intended referent. However, human-produced data demonstrates that, for any given entity, many perfectly acceptable referring expressions exist. At the same time, it is not the case that all logically possible descriptions are acceptable; so, if we remove the requirement to produce only one best solution, how do we avoid generating undesirable descriptions? Our aim in this paper is to sketch a framework that allows us to capture constraints on referring expression generation, so that the set of logically possible descriptions can be reduced to just those that are acceptable.

1 Introduction

The literature contains many algorithms for the generation of referring expressions: see, for example, (Dale, 1989; Dale and Haddock, 1991; Gardent, 2002; Varges and van Deemter, 2005; Gatt, 2006). These algorithms generally attempt to produce a single ‘best’ referring expression for a given intended referent. What counts as ‘best’ is generally defined in terms of minimality and the redundancy of information: the best referring expression is the shortest possible distinguishing description, usually defined in terms of the number of properties expressed. At the same time, some researchers (for example, (Dale and Reiter, 1995; Kraemer et al., 2003)) have noted that human-produced referring expressions are often not minimal in this sense, and so variations on these algorithms weaken this requirement, while still tending to embody a ‘shorter is better’ criterion.

This focus on minimality has the consequence that it allows us to ignore the abundant evidence that any intended referent can be successfully and appropriately referred to by a large number of referring expressions, many of which involve some

redundancy; see, for example, the data described in (Viethen and Dale, 2006). Once we remove the requirement of minimality, and admit that there are many possible acceptable solutions to the problem of how to refer to an intended referent, we are faced with a new problem: for any given entity, there are many logically possible distinguishing descriptions, and we need some way to navigate this space of possibilities, so that we can at least separate the acceptable from the less acceptable. This paper attempts to establish a framework for thinking about this problem.

In Section 2, we begin by first discussing the question of domain-specificity; our argument here is that we are at too early a stage to come up with a definition of ‘acceptable reference’ that is universally applicable, and so we must begin by acknowledging what we call *the hierarchy of referential domains*. We also introduce the domain that we use as the focus of discussion in this paper. Then, in Section 3, we discuss the idea of a *space of descriptions*, distinguishing the two notions of *logically possible description* and *acceptable description* and presenting some statistics that demonstrate the scale of the problem we face. In Section 4, we enumerate a collection of constraints on acceptable reference that we have identified as being applicable in the domains under consideration here. Finally, in Section 5, we draw some conclusions and consider how the ideas presented in this paper might be taken further.

2 Domain Specificity

2.1 On the Generality of Solutions

Early algorithms for referring expression generation (for example, (Dale, 1989)) attempted to provide domain-independent characterisations of what makes a good referring expression. Subsequent work, and specifically that in the spirit of the Incremental Algorithm (IA; (Dale and Reiter, 1995)) acknowledged that there were domain-specific aspects to the problem; in the case of the IA, this involved using a gen-

eral, domain-independent algorithm in conjunction with domain-specific preference orderings over the available properties to be used in descriptions. With the question of domain-specificity conveniently delegated to the subtask of determining appropriate preference orderings, research in this area typically then goes on to explore only a single chosen domain.¹

Our suspicion is that domain-specific characteristics have a much greater impact on the problem of referring expression generation than this simple view suggests. Dale et al. (2002) suggested that additional higher-level domain-specific *reference strategies* were required to provide more guidance over the process of referring expression generation. Here, we note that a monotonic combination of the costs or rankings of potential attributes, as suggested by Krahmer et al. (2003), is based on the assumption that adding another property to an already distinguishing referring expression will never make the expression better; but this is by no means always the case. In many instances the incorporation of another property — such as a visually very salient one — can increase the usefulness of a referring expression for the hearer, but this very much depends upon the domain. Research into how to effectively rank referring expressions for entities in a specific domain therefore first requires an exploration into the characteristics of the properties that occur in that domain and the dependencies between these properties. In short, we do not believe there is a short-cut around the problem of domain-specificity.

This does not mean that we are reduced to providing algorithms that are devoid of any generality. Rather, as we argue below, we may be able to determine a structure over the space of domains that makes it possible to provide solutions that apply at varying degrees of domain-specificity.

2.2 Grid-Based Domains

Our recent work on referring expression generation (Viethen and Dale, to appear) is based on a domain which has arguably very specific characteristics compared to many other application domains in NLG. It consists of 16 filing cabinet drawers which form a four-by-four grid. Four drawers each have one of four colours (yellow, orange, pink and blue); the colours are distributed randomly over

1 (blue)	2 (orange)	3 (pink)	4 (yellow)
8 (blue)	7 (blue)	6 (yellow)	5 (pink)
9 (orange)	10 (blue)	11 (yellow)	12 (orange)
16 (yellow)	15 (pink)	14 (orange)	13 (pink)

Figure 1: The filing cabinets

the grid, as shown in Figure 1.

In exploring the generation and evaluation of referring expressions in this domain, we determined that specific characteristics of the domain have a considerable influence on the performance of different algorithms. In (Viethen and Dale, 2006) we discuss how the very good performance of the IA, and the contrasting unexpectedly bad results for the Relational Algorithm (Dale and Haddock, 1991), are mainly due to the peculiarities of the drawer domain. In particular, the domain’s regular grid-like layout and the uniformity of the possible referents with regard to their types and properties put the Relational Algorithm, at least as originally specified, at a great disadvantage.

This does not mean, however, that more specifically-tailored solutions for reference in this domain are then so tied to the description of filing cabinets that they cannot be used for anything else. Rather, we take the view that our specific domain of enquiry is an instance of a more general class of domains, which we refer to as the *grid-based domains*. This class of domains shares a set of common characteristics:

Type homogeneity: All potential referents are of the same type; related objects of other types might be used to describe the referent, but these objects are not considered distractors.

Positional precision: Every entity has an exact position in the grid, allowing a unique description using only positional information.

Connectedness: Every entity is connected to each of its neighbours in the grid by a spatial relation, such as above or left-of.

Also, in domains like this, the properties of the entities fall into two categories: they are either *positional* (for example, row or column, and corner-

¹Interestingly, as far as we are aware, no significant studies have been carried out to determine how such preference orderings might actually be identified in any given domain.

hood or other special positional information such as on-an-edge or in-the-middle) or *intrinsic* (for example, colour, size, or shape).

Many domains share these properties: for example, blocks or buildings on a street map with a grid layout, as can be found in many inner cities of Australia or the United States; cubicle layouts in open-plan offices; cells in spreadsheets or tables of many different kinds; the boards of many games such as checkers; windows in large building blocks; and cars in car parks, perhaps as imagined from an aerial view. We might consider these all to be *regular* grid-based domains; we can then also identify somewhat less regular grid-based domains, such as map references, books in bookshelves, bricks in a wall, or pictures on display.

Our proposal here is that, while we carry out our experimentation in one particular grid-based domain, we can still identify strategies for reference that work across all grid-based domains; consequently, we can provide solutions that, while they are clearly not domain-independent, are of more value than strategies tied to a single very specific domain.

2.3 A Hierarchy of Referential Domains

The basic idea underlying the above proposal is that we can impose some structure over the space of possible *referential domains* (domains in which we want to refer to things) that allows us to determine different degrees of domain-specificity. It is too early to determine what such a hierarchy of domains would look like, but there are some reasonable suggestions we might make. For example, it is plausible that, at the top level, we can partition the space of all referential domains into those that are physical or concrete, on the one hand, and those that are abstract (such as the domains of beliefs, or mathematical objects) on the other hand, with different strategies for reference appropriate in each.

Similarly, we can hypothesise some domain structure from the bottom-up. Above, we identified a subset of domains that we referred to as grid-based; we might consider the set of grid-based domains to be a subtype of the set of ‘physical layout’ domains, such as a collection of objects on a table or plants in a garden. Such domains clearly share some of the characteristics of grid-based domains that will be relevant for reference, but at the same time embody significant differences that

will render some specifically grid-based reference strategies inappropriate.

These are only suggestions; our view is that detailed consideration of quite different domains is required before we can properly establish the relationships between these domains. The main point we are arguing for here is that some degree of domain specificity cannot be ignored. To put the point more strongly: we argue that further progress on the development of algorithms for referring expression generation will only result if we move away from a focus on domain-independent aspects of the problem.

2.4 Reference in Grid-Based Domains

The special characteristics of grid-based domains noted above have an impact on the kinds of referring expressions that we might generate.

First, as already noted, the property of *positional precision* means that for every entity in a grid-based domain, there exists a unique description which only uses the row and column properties of that object, such as *the drawer in the bottom row, third column* for d14 in Figure 1.

The property of *type homogeneity* means that the entities in the domain often share a set of grid-independent properties. In the case of the drawer domain, this set only consists of colour; in the car park domain, this would include properties such as make and model, colour, or modifications such as spoilers. This characteristic adds to the uniformity of descriptions across objects, and therefore allows for reference strategies to be quite generally applicable in the domain.

The *positional precision* and *connectedness* characteristics mean that it is possible to infer the exact grid location of one entity from that of another whose location is specified. For example, if we know that one object is in row 2 and column 4, then we know that the object to the right of it is in row 2 and column 3. As we will see in Section 4, this has an influence on the usefulness of grid-dependent properties of objects that are spatially related to the referent.

Another consequence of *connectedness* is that the normal transitivity of spatial relations such as above, below, and left-of does not play a role in these domains. Of course, it is logically true that if object2 is below object1 and object3 is below object2, then object3 is also below object1. However, this relation between object1 and object3 would never be

used in a referring expression describing either of these two entities. In the drawer domain, a hearer would always understand *the drawer below the orange drawer in the top row* to be d7, the drawer *directly* below d2, not d10 or d15.

Our point here is that the nature of the domain has an impact on how the properties available in that domain might be used to refer to entities. Most importantly, these domain characteristics impact on the applicability of ‘general purpose’ algorithms, as we saw in (Viethen and Dale, 2006) with regard to the poor performance of the relational algorithm of Dale and Haddock (1991) in a new domain.

3 The Space of Descriptions

3.1 The Possible and the Acceptable

In any domain where entities are described via a finite set of properties and relations, there are a finite number of possible descriptions of a given entity. We might think of these as the set of *logically possible referring expressions*. This set will of course be combinatorial in the number of properties and related entities. Even if we limit the set to those which are *distinguishing* descriptions, we are still faced with a large set of descriptions to choose from.

Existing algorithms effectively provide ways to search this space, generally oriented towards finding shorter solutions before longer ones are considered. These strategies favour referring expressions which avoid or minimise redundancy and stop as soon as one referring expression is found, but as we know, human descriptions are often redundant and there is usually more than one acceptable solution; these algorithms therefore will fail to find many descriptions which are in fact quite acceptable from the point of view of readers or listeners. This is in effect taking an engineering perspective: find one good solution that can do the job, then stop exploring other possibilities. A more interesting approach to the problem might be from the speaker’s and the listener’s perspectives: identify all the acceptable referring expressions for an object and then rank them by usefulness to the parties involved.

These observations then raise some questions. In particular, are all logically possible distinguishing descriptions also *acceptable* to the speaker or listener? And if not, how do we rule out those which are not acceptable descriptions? It seems

likely that many of these descriptions—especially the very long, multiply redundant ones—will not be particularly useful.

Below, we provide some basic statistics that demonstrate the size of the space of possible descriptions that are available in even a simple domain like the one focussed on here. Then, in Section 4, we propose some plausible constraints on this space of possible descriptions.

3.2 The Size of the Space

In our present work, we use an algorithm which is based on the graph-based framework for referring expression generation described in Krahrmer et al. (2003) to generate all logically possible distinguishing descriptions for a target referent. We concentrate on referring expressions containing simple attributive properties and binary relations between pairs of entities. More complex expressions involving plurals, and Boolean combinations of properties or quantifiers (see among others (Gatt, 2006), Varges and van Deemter (2005) and van Deemter and Krahrmer (2007 to appear)) are not included in our current investigation; but even without these, the number of possible descriptions we have to consider is very large.

The attributive properties we encode for the drawer domain are colour, row and column for all drawers, and position with the value corner for the four corner drawers. Of the relational properties, only above and left-of are explicitly encoded, so as to avoid circularity. It is left to the realisation level to determine whether to realise the edge (a above b) as *A which is above B* or *B which is below A*, depending on whether A or B is the entity being described.²

We use a parameter `maxNodes` to delimit the set of descriptions produced by our algorithm; this determines how many entities can maximally be included in each referring expression. We currently investigate referring expressions involving no more than two entities. Three reasons justify this seemingly low cut off: firstly, as each entity in our domain has 3–4 attributive properties, it seems unnecessary to consider descriptions containing long chains of relations such as *the blue drawer above the blue drawer left of the drawer in the third row below the yellow drawer*; secondly, our set of 140 human-produced descriptions for

²For representational issues in the drawer domain and the graph-based framework, see (Viethen and Dale, 2006) and (Krahrmer et al., 2003) respectively.

this domain contains only four relational descriptions with more than one relation; and finally and most practically, allowing more than two entities in a description extends the set of descriptions to be considered from hundreds to thousands.

If we allow maximally one related entity to be included alongside the intended referent, it turns out that we have an average of about 212 possible distinguishing descriptions per drawer. For example, for drawer d1 in Figure 1, the number of candidate descriptions consists of the following sets, where ‘DD’ means ‘distinguishing description’:

- $|\{\text{DDs of d1}\}| = 8$
- $|\{\text{DDs of d1}\} \times \{\text{DDs of d2 or d8}\}| = 8 \times (4+2) = 48$
- $|\{\text{DDs of d1}\} \times \{\text{non-DDs of d2 or d8}\}| = 8 \times (4+6) = 80$
- $|\{\text{non-DDs of d1}\} \times \{\text{DDs of d2 or d8}\}| = 8 \times (4+2) = 48$
- $|\{\{\text{non-DDs of d1}\} \times \{\text{non-DDs of d2 or d8}\}, \text{ where the resulting description is distinguishing}\}| < 8 \times (4+6) = 80$. In fact, for this drawer the number is 44.

This results in a total of 228 candidate descriptions for drawer d1 using at most one relational property.

Taking into account that, depending on a drawer’s position in the grid, there are between 4 and 12 distinct possible combinations of two neighbours in a relational description; that each candidate description only needs to describe one of the contained drawers distinctly to distinguish the intended referent; and that there are a large number of non-distinguishing descriptions that become distinguishing when combined with a non-distinguishing description for one or more other drawers, it becomes clear that the number of candidate descriptions involving three drawers lies in the thousands. In fact, there are between 5136 and 8834 relational descriptions with up to two relatives for each drawer, or about 6764 on average.

4 Constraints on Acceptability

Of course, no sensible algorithm would generate the set of all possible descriptions and then attempt to select from amongst these. However, as we noted above, generate-and-test search strategies like those present in existing algorithms will fail to discover many acceptable descriptions. We would like, therefore, to see if we can identify ways of

constraining this space of possible descriptions to a more manageable set; any constraints so determined might then form the basis of a revised generate-and-test search strategy that does not focus on minimality, but rather on acceptability, and is able to find all acceptable solutions, not just one.

On the basis of an examination of the kinds of referring expressions produced by our graph-based algorithm for this domain, we first define in Section 4.1 two ‘whitelist’ rules that allow us to identify a number of referring expressions that are considered acceptable under all circumstances. We then go on in Section 4.2 to describe a collection of ‘blacklist’ constraints which exclude certain types of referring expressions as unacceptable. Finally, in Section 4.4 we provide three examples of rules that can be used for ranking the remaining referring expressions in this domain and, we believe, other grid-based domains.

4.1 Whitelist rules

Minimality: In any domain, if the description is a minimal distinguishing description, it is considered acceptable, where a *minimal* referring expression is defined as a *shortest possible* one.³

Non-relationality: In domains with only a small number of available attributive properties for each entity, as is the case in the drawer domain, we will consider all non-relational distinguishing referring expressions as acceptable, regardless of any redundancy they contain. In other domains, descriptions using long lists of attributive properties may become too long and cumbersome, and may lead to false implicatures.

4.2 Blacklist constraints

After applying these two white-list rules, we can direct our focus towards the remaining descriptions, which will either be non-minimal (i.e., overspecified), relational, or both. We make two general observations regarding constraints on the acceptability of these types of referring expressions.

Firstly, the two categories of properties in grid-based domains discussed in Section 2.2 have an impact on the acceptability of overspecification in referring expressions. We observe that it seems to be the case that informationally redundant *intrinsic* properties are more useful—or even desirable— than redundant *positional* properties.

³See (Dale, 1989). For present purposes, we will assume that all descriptions are ‘accessible’, in that they can be determined by the listener to be true of the intended referent.

For example, while the mention of the relatee and its colour in *the pink drawer in the far right that's below the yellow drawer* is unnecessary, this description is clearly more acceptable than *the pink drawer in the far right that's below the drawer in the top row*, where instead of colour, the row property is included for the relatee.

Secondly, we observe that, in our human-produced data, relations are only used under two circumstances:

- (a) The attributive properties of the relatee are more visually salient than those of the target. So, the target gets described mainly in terms of its relation to a more salient entity, as for example in *the book left of the huge black lexicon in the bottom shelf*.
- (b) The combination of properties of target and relatee, including the relation holding between them, is more or as visually salient as the target alone. An example of this, taken from our human-produced data set, is *the yellow drawer that's above another yellow drawer* for drawer d6.

Note also, as discussed in Section 3.1, that we have already ruled out those descriptions which are not distinguishing descriptions. We then propose the following constraints to further reduce the space of acceptable referring expressions.

C1: No relatees without attributes: A referring expression should at least contain one attributive property for each relatee to the referent. In most domains the minimum requirement is that the type of each entity is included. This is a commonly observed phenomenon, of course, and is often catered for in algorithms by means of a special case ‘necessary-inclusion’ rule. Note, however, that for highly connected domains where all the referents are of the same type (as in our drawer domain), this constraint also excludes descriptions containing only type as the attributive property used for a relatee. Consequently, Examples (1) and (2) are excluded by this constraint, while Example (3) is not:

- (1) the blue drawer in the first column below another drawer
- (2) the thick book in the third shelf from the top, left of another book
- (3) the bush under the tree

This constraint reduces the average number of descriptions per drawer from about 212 to 200.

C2: No relatee without salient properties: Either the relatee itself or the combination of the relatee and the intended referent need to be as readily locatable as the referent described in only attributive terms. If this is not the case, the mention of the relatee renders the expression more informative than required and adds the potential for confusion. While the visual salience of an entity or specific properties is difficult to determine for most domains, in the drawer domain it is straightforward that the two properties that can contribute to the visual salience of an entity are its colour and its being in the corner position. We therefore exclude descriptions containing relatees without either of these two properties, such as:

- (4) the yellow drawer in the second row, third column that's left of the drawer in the second row, fourth column
- (5) the pink drawer left of the drawer in the bottom row, third column.

This constraint reduces the average number of descriptions per drawer by another 65 to about 135.

C3: No grid properties for less salient relatees: In regular grid-based domains, it appears there is no good reason to include row and column properties for relatees. This information can either be inferred from the intended referent, or forces the listener to perform the opposite inference to find the location of the intended referent. This constraint excludes descriptions such as the following:

- (6) the big red book next to the little booklet, which is the fourth book from the left in the second shelf from the bottom
- (7) the blue drawer in the second row left of the yellow drawer in the third column

The only cases where grid information for a relatee might be useful are (i) situations where the relatee is more easily locatable than the intended referent (as in Example 8), and (ii) situations where the relatee only has the bit of grid property it shares with the intended referent and no other grid information is contained in the referring expression (as in Example 9).⁴

⁴Note that, in the second case, the PP attachment ambiguity means that we can see in *the leftmost column* to be either

- (8) the blue book below the fat brown dictionary in the third shelf from the top
- (9) the blue drawer above the orange drawer in the leftmost column

4.3 Status of the Constraints

In the previous section, we offered three domain-specific constraints on acceptable reference that substantially reduce the number of logically possible descriptions that we might want to consider. The constraints are based on our observations on the data; while the real status of the constraints requires more rigorous experimental testing, we would suggest that they do seem intuitively plausible when expressed as general rules, and they also succeed in ruling out specific referring expressions that are plausibly dispreferred.

To back up our intuition, we conducted a small evaluation exercise where seven native English speakers were asked to directly compare pairs of randomly chosen descriptions. One description in every pair was from the set of left over acceptable descriptions for one drawer; the other one was taken from the set of descriptions for the same drawer that were excluded by the constraints. Each participant could choose to compare between 5 and 20 pairs for 4 drawers each, which resulted in 361 comparisons.

Overall, the participants preferred the description from the set deemed acceptable in 69% of the cases. The descriptions considered acceptable by C1 and C3 were chosen in 64% of the cases, while the hypothesis of C2 was supported in 78% of the cases. Interestingly, there were two participants who overall slightly preferred the descriptions excluded by the constraints.

While these numbers are mildly encouraging in the characterisation of the constraints presented here, we do not claim that it is ultimately the correct one. Our main point here is that we need to identify such constraints for any given domain or class of domains.

4.4 Rules for Ranking

After applying the black-list constraints, we are still left with a fairly large number of possible descriptions. We are less confident of ruling subsets of these out as unacceptable, but it does seem to us that some are more acceptable than others.

a property of the intended referent or of the relatee. This constraint results in an average of about 66 descriptions per drawer.

We suggest, therefore, that we may also require a number of general ranking rules along the following lines.

R1: Not too many positional properties: In small grid domains, it seems desirable to either exclude all positional properties, or to use at most one bit of grid information. The smaller a domain, the less need there is to direct the hearer's focus towards the intended referent by the use of more and more precise position information.

Most relational descriptions in our human-produced set for the drawer domain are of the type <colour–relation–colour>:

- (10) the yellow drawer that's above another yellow drawer
- (11) the blue drawer above the pink drawer

However, there are three examples of relational descriptions containing one bit of grid information in our human-produced data, and other examples with more position information than this cannot be excluded on the grounds of not being contained in a data set of only 140 items.

If we choose to limit relational descriptions in the drawer domain to at most one bit of grid information, the average number of descriptions after applying the blacklist constraints decreases by 35 to about 31.

As this is a fairly large reduction, we included this rule in our evaluation exercise and found that only 54% of the comparisons turned out in its favour. However, without the vote of the two participants who overall preferred the excluded descriptions, the hypothesis of R1 was supported in 70% of the cases.

R2: Always use location: The opposite effect emerges in larger domains: the bigger the search space, the more helpful positional information is in directing the hearer's focus into the relevant area of the domain. This is especially true for uniform domains, where most entities have the same type and other attributive properties of similar salience. For example, in a large bookshelf containing only randomly-ordered small paperbacks in many different colours, a referring expression only containing *intrinsic* properties such as colour, size, or title is unlikely to be very helpful.

R3: No relational descriptions: In very small domains, it can be best to exclude all relational descriptions, given that each object can be described

uniquely using only attributive properties. In a 3×3 grid, descriptions such as *the small blue object left of the red object in the top right corner* or even *the blue object left of the red object* will always be significantly more cumbersome than non-relational descriptions.

5 Conclusions and Future Work

In this paper, we have taken the view that the process of referring expression generation fundamentally involves domain-specific principles. This position does not need to result in chaos, with every domain having its own hand-crafted rules for reference; rather, we propose that referential domains are ordered in a subsumption hierarchy, which allows us to group domains according to common characteristics. These characteristics can help us define what counts as an acceptable referring expression for domains of a given type. To illustrate our claim, we explore the use of constraints in grid-based domains, using our drawer domain as a specific example, and show how this allows us to reduce the very large set of logically possible descriptions to a more manageable set of acceptable descriptions.

Evidence from human-produced referring expressions shows that for any given object a variety of acceptable descriptions exist. This runs contrary to the prevailing assumption that the aim of a referring expression generation system should be to find only one best description for a target referent. We have provided an approach to determining acceptability that denies this assumption, and is more in line with real human behaviour.

There are some clear steps forward from the position taken in this paper.

- First, we need to develop algorithms that can use the kinds of constraints we have discussed, so that we don't have to generate all the bad referring expressions to find the good ones.
- There are a range of psycholinguistic experiments that could be carried out both to validate the constraints and rules we have identified, and to test algorithms based on these constraints.
- More generally, the notion of a hierarchy of referential domains requires further exploration.

In conclusion, it is clear that humans can produce many acceptable referring expressions for a given intended referent. We have argued that a focus on minimality and single 'best' solutions has allowed the field to avoid the fact that the range of acceptable descriptions can only be characterised by reference to characteristics of the domain in question.

References

- Dale, R. and Haddock, N. 1991. Generating referring expressions involving relations. In *Proceedings of the 5th Conference of the European Chapter of the ACL*, 161–166, Berlin, Germany.
- Dale, R. and Reiter, E. 1995. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.
- Dale, R., Geldof, S., and Prost, J. 2002. Generating more natural route descriptions. In *Proceedings of the Australasian Natural Language Processing Workshop*, Canberra, Australia.
- Dale, R. 1989. Generating recipes: An overview of EPICURE. In *Proceedings of the 2nd European Workshop on Natural Language Generation*, Edinburgh, UK.
- Gardent, C. 2002. Generating minimal definite descriptions. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA, USA.
- Gatt, A. 2006. Generating collective spatial references. In *Proceedings of the 28th Annual Meeting of the Cognitive Science Society*, Vancouver, Canada.
- Krahmer, E., van Erk, S., and Verleg, A. 2003. Graph-based generation of referring expressions. *Computational Linguistics*, 29(1):53–72.
- van Deemter, K. and Krahmer, E. 2007, to appear. Graphs and Booleans: On the generation of referring expressions. In Bunt, H. and Muskens, R. (Eds.), *Computing Meaning*, vol. 3, 17–53. Kluwer, Dordrecht, The Netherlands.
- Varges, S. and van Deemter, K. 2005. Generating referring expressions containing quantifiers. In *Proceedings of the 6th International Workshop on Computational Semantics*, Tilburg, The Netherlands.
- Viethen, J. and 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*, 63–70, Sydney, Australia.
- Viethen, J. and Dale, R. to appear. Evaluation in natural language generation: Lessons from referring expression generation. *Traitement Automatique des Langues*, 47(3).