Towards the Evaluation of Referring Expression Generation

Jette Viethen
Centre for Language Technology
Macquarie University
Sydney NSW 2109
jviethen@ics.mq.edu.au

Robert Dale
Centre for Language Technology
Macquarie University
Sydney NSW 2109
robert.dale@mq.edu.au

Abstract

The Natural Language Generation community is currently engaged in discussion as to whether and how to introduce one or several shared evaluation tasks, as are found in other fields of Natural Language Processing. As one of the most well-defined subtasks in NLG, the generation of referring expressions looks like a strong candidate for piloting such shared tasks. Based on our earlier evaluation of a number of existing algorithms for the generation of referring expressions, we explore in this paper some problems that arise in designing an evaluation task in this field, and try to identify general considerations that need to be met in evaluating generation subtasks.

1 Introduction

In recent years, the inclusion of an evaluation component has become almost obligatory in any publication in the field of Natural Language Processing. For complete systems, user-based and task-oriented evaluation are almost standard practice in both the Natural Language Understanding (NLU) and Natural Language Generation (NLG) communities. A third, more competitive, form of evaluation has become increasingly popular in NLU in the form of shared-task evaluation campaigns (STECs). In a STEC, different approaches to a well-defined problem are compared based on their performance on the same task. A large number of different research communities within NLP, such as Question Answering, Machine Translation, Document Summarisation, Word Sense Disambiguation, and Information Retrieval, have adopted a shared evaluation metric and in many cases a shared-task evaluation competition.

The NLG community has so far withstood this trend towards a joint evaluation metric and a competitive evaluation task, but the idea has surfaced in a number of discussions, and most intensely at the 2006 International Natural Language Generation Conference (see, for example, Bangalore et al. (2000), Reiter and Sripada (2002), Reiter and Belz (2006), Belz and Reiter (2006), Belz and Kilgarriff (2006), Paris et al. (2006), and van Deemter et al. (2006)).

Amongst the various component tasks that make up Natural Language Generation, the generation of referring expressions is probably the subtask for which there is the most agreement on problem definition; a significant body of work now exists in the development of algorithms for generating referring expressions, with almost all published contributions agreeing on the general characterisation of the task and what constitutes a solution. This suggests that, if formal shared tasks for NLG are to be developed, the generation of referring expressions is a very strong candidate.

In (Viethen and Dale, 2006), we argued that the evaluation of referring expression generation algorithms against natural, human-generated data is of fundamental importance in assessing their usefulness for the generation of understandable, natural-sounding referring expressions. In this paper, we discuss a number of issues that arise from the evaluation carried out in (Viethen and Dale, 2006), and consider what these issues mean for any attempt to define a shared task in this area.

The remainder of this paper has the following structure. In Section 2, we briefly describe the evaluation experiment we carried out for three well-established referring expression generation

algorithms, and report the performance of these algorithms in the chosen test domain. This leads us to identify three specific issues that arise for the evaluation of referring expression generation algorithms, and for NLG systems in general; we discuss these in the subsequent sections of the paper. Section 3 looks at the problem of input representations; Section 4 explores how the wide variety of acceptable outputs, and the lack of a single correct answer, makes it hard to assess generation algorithms; and Section 5 explores whether we can usefully provide a numeric measure of the performance of a generation algorithm. Finally, in Section 6 we point to some ways forward.

2 An Evaluation Experiment

In (Viethen and Dale, 2006), we observed that surprisingly little existing work in natural language generation compares its output with natural language generated by humans, and argued that such a comparison is essential. To this end, we carried out an experiment consisting of three steps:

1. the collection of natural referring expressions for objects in a controlled domain, and the subsequent analysis of the data obtained;
2. the implementation of a knowledge base corresponding to the domain, and the re-implementation of three existing algorithms to operate in that domain; and
3. a detailed assessment of the algorithms' performance against the set of human-produced referring expressions.

In the remainder of this section we briefly describe these three stages. As we are mainly concerned here with the evaluation process, we refer to (Viethen and Dale, 2006) for a more detailed account of the experimental settings and an in-depth discussion of the results for the individual algorithms.

2.1 The Human-Generated Data

Our test domain consists of four filing cabinets, each containing four vertically arranged drawers. The cabinets are placed directly next to each other, so that the drawers form a four-by-four grid as shown in Figure 1. Each drawer is labelled with a number between 1 and 16 and is coloured either blue, pink, yellow, or orange. There are four drawers of each colour distributed randomly over the grid.

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 human participants were given, on a number of temporally-separated occasions, a random number between 1 and 16, and then asked to provide a description of the corresponding drawer to an onlooker without using any of the numbers; this basically restricted the subjects to using either colour, location, or some combination of both to identify the intended referent. The characterisation of the task as one that required the onlooker to identify the drawer in question meant that the referring expressions produced had to be *distinguishing descriptions*; that is, each referring expression had to uniquely refer to the intended referent, but not to any of the other objects in the domain.

The set of natural data we obtained from this experiment contains 140 descriptions. We filtered out 22 descriptions that were (presumably unintentionally) ambiguous or used reference to sets of drawers rather than only single drawers. As none of the algorithms we wanted to test aims to produce ambiguous referring expressions or handle sets of objects, it is clear that they would not be able to replicate these 22 descriptions. Thus the final set of descriptions used for the evaluation contained 118 distinct referring expressions.

Referring expression generation algorithms typically are only concerned with selecting the semantic content for a description, leaving the details of syntactic realisation to a later stage in the language production process. We are therefore only interested in the semantic differences between the descriptions in our set of natural data, and not in superficial syntactic variations. The

primary semantic characteristics of a referring expression are the properties of the referent used to describe it. So, for example, the following two referring expressions for drawer d3 are semantically different:

(1) The pink drawer in the first row, third column.

(2) The pink drawer in the top.

For us these are distinct referring expressions. We consider syntactic variation, on the other hand, to be spurious; so, for example, the following two expressions, which demonstrate the distinction between using a relative clause and a reduced relative, are assumed to be semantically identical:

(3) The drawer that is in the bottom right.

(4) The drawer in the bottom right.

We normalised the human-produced data to remove syntactic surface variations such as these, and also to normalise synonymic variation, as demonstrated by the use of the terms *column* and *cabinet*, which in our context carry no difference in meaning.

The resulting set of data effectively characterises each human-generated referring expression in terms of the semantic attributes used in constructing those expressions. We can identify four absolute properties that the human participants used for describing the drawers: these are the colour of the drawer; its row and column; and in those cases where the drawer is located in one of the corners of the grid, what we might call cornerhood. A number of participants also made use of relations that hold between two or more drawers to describe the target drawer. The relational properties that occurred in the natural descriptions were: above, below, next to, right of, left of and between. However, relational properties were used a lot less than the other properties: 103 of the 118 descriptions (87.3%) did not use relations between drawers.

Many referring expression generation algorithms aim to produce minimal, non-redundant descriptions. For a referring expression to be minimal means that all of the facts about the referent that are contained in the expression are essential for the hearer to be able to uniquely distinguish the referent from the other objects in the domain. If any part of the referring expression was dropped, the description would become ambiguous; if any other information was added, the resulting expression would contain redundancy.

Dale and Reiter (1995), in justifying the fact that their Incremental Algorithm would sometimes produce non-minimal descriptions, pointed out that human-produced descriptions are often not minimal in this sense. This observation has been supported more recently by a number of other researchers in the area, notably van Deemter and Halldórsson (2001) and Arts (2004). However, in the data from our experiment it is evident that the participants tended to produce minimal descriptions: only 24.6% of the descriptions (29 out of 118) contain redundant information.

2.2 The Algorithms

Many detailed descriptions of algorithms are available in the literature on the generation of referring expressions. For the purpose of our evaluation experiment, we focussed here on three algorithms on which many subsequently developed algorithms have been based:

- The Full Brevity algorithm (Dale, 1989) uses a greedy heuristic for its attempt to build a minimal distinguishing description. At each step, it always selects the most discriminatory property available.

- The Relational Algorithm from (Dale and Haddock, 1991) uses constraint satisfaction to incorporate relational properties into the framework of the Full Brevity algorithm. It uses a simple mechanism to avoid infinite regress.

- The Incremental Algorithm (Reiter and Dale, 1992; Dale and Reiter, 1995) considers the available properties to be used in a description via a predefined preference ordering over those properties.

We re-implemented these algorithms and applied them to a knowledge base made up of the properties evidenced collectively in the human-generated data. We then analysed to which extent the output of the algorithms for each drawer was semantically equivalent to the descriptions produced by the human participants. The following section gives a short account of this analysis.

2.3 Coverage of the Human Data

Out of the 103 natural descriptions that do not use relational properties, the Full Brevity Algorithm is able to generate 82 by means of at least one preference ordering over the object properties, providing a recall of 79.6%. The recall achieved by the Incremental Algorithm is 95.1%: it generates 98 of the 103 descriptions under at least one preference ordering. The relational descriptions from the natural data are not taken into account in evaluating the performance of these two algorithms, since they are not designed to make use of relational properties.

Both the Full Brevity Algorithm and the Incremental Algorithm are able to replicate all the minimal descriptions found in the natural data. Against its specification to avoid all redundancy, the Full Brevity Algorithm also generates nine of the redundant descriptions; the Incremental Algorithm replicates 24 of the 29 redundant descriptions produced by humans.

Perhaps surprisingly, the Relational Algorithm does not generate *any* of the human-produced descriptions. The particular strategy adopted by this algorithm is quite at odds with the human-generated descriptions in our data; we refer the reader to (Viethen and Dale, 2006) for a discussion of this failure, since it does not have a direct bearing on the present topic.

We now go on to discuss some of the key issues for NLG evaluation that became evident in this experiment.

3 Deciding on Input Representations

3.1 A Key Problem in NLG

It is widely accepted that the input for NLG systems is not as well-defined as it is in NLU tasks. In NLU the input will always be natural language, which is processed according to the task and transformed into *a machine-usable format of some kind*. In NLG, on the other hand, we are working in the other direction: there exists no consensus of what exact form the input into the system should take. The input is a knowledge base in *a machine-usable format of some kind*, whereas it is the desired format of the output—natural language—that is clear. As Yorick Wilks is credited with observing, Natural Language Understanding is like counting from 1 to infinity, but Natural Language Generation is like the much more perplexing task of counting from infinity to 1. The problem of determining what the generation process starts from is probably one of the major reasons for the lack of shared tasks in the field: each researcher chooses a level of representation, and a population of that level of representation, that is appropriate to exploring the kinds of distinctions that are central to the research questions they are interested in.

3.2 A Problem for Referring Expression Generation

As alluded to earlier, the generation of referring expressions seems to avoid this problem. The task is generally conceived as one where the intended referent, and its distractors in the domain, are represented by symbolic identifiers, each of which is characterised in terms of a collection of attributes (such as colour and size) with their corresponding values (red, blue, small, large...).

However, this apparent agreement is, ultimately, illusory. A conception in terms of symbolic identifiers, attributes, and values provides only a schema; to properly be able to compare different algorithms, we still need to have agreement on the specific attributes that are represented, and the values these attributes can take.

As we employed a new domain for the purpose of our evaluation experiment, we had to first decide how to represent this domain. Some of our representational primitives might seem to be non-contentious: the choice of colour, row and column seem quite straightforward. However, we also explicitly represented a more controversial attribute position, which took the value corner for the four corner drawers. Although cornerhood can be inferred from the row and column information, we added this property explicitly because it seems plausible to us that it is particularly salient in its own right.

This raises the general question of what properties should be encoded explicitly, and which should be inferred. In our experiment, we explicitly encode relational properties that could be computed from each other, such as left-of and right-of. We also chose not to implement the transitivity of spatial relations. Due to the uniformity of our domain the implementation of transitive inference would result in the generation of unnatural descriptions, such as *the orange drawer (two) right of the blue drawer* for d_{12}. Since none of the algorithms explored in our experiment uses inference over knowledge base properties, we opted here

to enable a fairer comparison between human-produced and machine-produced descriptions and decided against any inferred properties.

The decisions we took regarding the representation of cornerhood, inferrable properties in general, and transitive properties, were clearly influenced by our knowledge of how the algorithms to be tested work. If we had only assessed different types of relational algorithms, we might have implemented corners, and possibly even columns and rows, as entities that drawers are spatially related to. If the assessed algorithms had been able to handle inferred properties, cornerhood might have been implemented only implicitly as a result of the grid information about a drawer. The point here is that our representational choices were guided by the requirements of the algorithms, and our intuitions about salience as derived from our examination of the data; other researchers might have made different choices.

3.3 Consequences

From the observations above, it is evident that, in any project that focusses on the generation of referring expressions, the design of the underlying knowledge base and that of the algorithms that use that knowledge base are tightly intertwined. If we are to define a shared evaluation task or metric in this context, we can approach this from the point of view of assessing only the algorithms themselves, or assessing algorithms in combination with their specific representations. In the first case, clearly the input representation should be agreed by all ahead of time; in the second case, each participant in the evaluation is free to choose whatever representation they consider most appropriate.

The latter course is, obviously, quite unsatisfactory: it is too easy to design the knowledge base in such a way as to ensure optimal performance of the corresponding algorithm. On the other hand, the former course is awash with difficulty: even in our very simple experimental domain, there are representational choices to be made for which there is no obvious guidance. We have discussed this problem in the context of what, as we have noted already, is considered to be a generation subtask on which there is considerable agreement; the problem is much worse for other component tasks in NLG.

4 Dealing with Determinism

4.1 There is More than One Way to Skin a Cat

One very simple observation from the natural data collected in our experiment is that people do not always describe the same object the same way. Not only do different people use different referring expressions for the same object, but the same person may use different expressions for the same object on different occasions. Although this may seem like a rather unsurprising observation, it has never, as far as we are aware, been taken into account in the development of any algorithm for the generation of referring expressions. Existing algorithms typically assume that there is a best or most-preferred referring expression for every object.

How might we account for this variation in the referring expressions that are produced by people? Where referring expressions are produced as part of natural dialogic conversation, there are a number of factors we might hypothesize would play a role: the speaker's perspective or stance towards the referent, the speaker's assumptions about the hearer's knowledge, the appropriate register, and what has been said previously. However, it is hard to see how these factors can play an important role in the simple experimental setup we used to generate the data discussed here: the entities are very simple, leaving little scope for notions of perspective or stance; and the expressions are constructed effectively *ab initio*, with no prior discourse to set up expectations, establish the hearer's knowledge, or support alignment. The sole purpose of the utterances is to distinguish the intended referent from its distractors.

We noted earlier that one regard in which multiple different descriptions of a referent may vary is that some may be redundant where others are not. Carletta (1992) distinguishes *risky* and *cautious* behaviour in the description task: while some participants would use only the briefest references, hoping that these would do the job, others would play safe by loading their descriptions with additional information that, in absolute terms, might make the overall description redundant, but which would make it easier or less confusing to interpret. It is possible that a similar or related speaker characteristic might account for some of the variation we see here; however, it would still not provide a basis for the variation even within the redundant

and minimal subsets of the data.

Of course, it can always be argued that there is no 'null context', and a more carefully controlled and managed experiment would be required to rule out a range of possible factors that predispose speakers to particular outcomes. For example, an analysis in terms of how the speakers 'come at' the referent before deciding how to describe it might be in order: if they find the referent by scanning from the left rather than the right (which might be influenced by the ambient lighting, amongst other things), are different descriptions produced? Data from eye-tracking experiments could provide some insights here. Or perhaps the variation is due to varying personal preferences at different times and across participants.

Ultimately, however, even if we end up simply attributing the variation to some random factor, we cannot avoid the fact that there is no single best description for an intended referent. This has a direct bearing on how we can evaluate the output of a specific algorithm that generates references.

4.2 Evaluating Deterministic Algorithms

The question arising from this observation is this: why should algorithms that aim to perform the task of uniquely describing the drawers in our domain have to commit to exactly one 'best' referring expression per drawer? In the context of evaluating these algorithms against human-generated referring expressions, this means that the algorithms start out with the disadvantage of only being able to enter one submission per referent into the competition, when there are a multitude of possible 'right' answers.

This issue of the inherent non-determinism of natural language significantly increases the degree of difficulty in evaluating referring expression algorithms, and other NLG systems, against natural data. Of course, this problem is not unique to NLG: recent evaluation exercises in both statistical machine translation and document summarisation have faced the problem of multiple gold standards (see Akiba et al. (2001) and Nenkova and Passonneau (2004), respectively). However, it is not obvious that such a fine-grained task as referring expression generation can similarly be evaluated by comparison against a gold standard set of correct answers, since even a large evaluation corpus of natural referring expressions can never be guaranteed to contain all acceptable descriptions for an object. Thus an algorithm might achieve an extremely low score, simply because the perfectly acceptable expressions it generates do not happen to appear in the evaluation set. Just because we have not yet seen a particular form of reference in the evaluation corpus does not mean that it is incorrect.

We might try to address this problem by encouraging researchers to develop non-deterministic algorithms that can generate many different acceptable referring expressions for each target object to increase the chances of producing one of the correct solutions. The evaluation metric would then have to take into account the number of referring expressions submitted per object. However, this would at most alleviate, but not entirely solve, the problem.

This poses a major challenge for attempts to evaluate referring expression generation algorithms, and many other NLG tasks as well: for such tasks, evaluating against a gold standard may not be the way to go, and some other form of comparative evaluation is required.

5 Measuring Performance

Related to the above discussion is the question of how we measure the performance of these systems even when we do have a gold standard corpus that contains the referring expressions generated by our algorithms. In Section 2.3, we noted that the Incremental Algorithm achieved a recall of 95.1% against our human-produced data set, which is to say that it was able to produce 95.1% of the descriptions that happened to appear in the data set; but as noted in the previous section, we cannot simply consider this data set to be a gold standard in the conventional sense, and so it is not really clear what this number means.

The problem of counting here is also impacted by the nature of the algorithm in question: as noted in Section 2.3, this performance represents the behaviour of the algorithm in question *under at least one preference ordering*.

The Incremental Algorithm explicitly encodes a preference ordering over the available properties, in an attempt to model what appear to be semi-conventionalised strategies for description that people use. The properties are considered in the order prescribed by the preference list and a particular property is used in the referring expression if it provides some discriminatory power, oth-

erwise it is skipped.

However, even within a single domain, one can of course vary the preference ordering to achieve different effects. It was by means of manipulation of the preference ordering that we were able to achieve such a high coverage of the human-produced data. We chose to view the manipulation of the preference ordering as the tweaking of a parameter. It could be argued that each distinct preference ordering corresponds to a different instantiation of the algorithm, and so reporting the aggregate performance of the collection of instantiations might be unfair. On the other hand, no single preference ordering would score particularly highly; but this is precisely because the human data represents the results of a range of different preference orderings, assuming that there is something analogous to the use of a preference ordering in the human-produced referring expressions. So it seems to us that the aggregated results of the best performing preference orderings provide the most appropriate number here.

Of course, such an approach would also likely produce a large collection of referring expressions that are not evidenced in the data. This might tempt us to compute precision and recall statistics, and assign such an algorithm some kind of F-score to measure the balance between under-generation and over-generation. However, this evaluation approach still suffers from the problem that we are not sure how comprehensive the gold standard data set is in the first place.

Ultimately, it seems that performance metrics based on the notion of coverage of a data set are fundamentally flawed when we consider a task like referring expression generation. We have argued above that asking the question 'Does the algorithm generate the correct reference?' does not make sense when there are multiple possible correct answers. The question 'Does the algorithm generate one of the correct answers?' on the other hand, is impracticable, because we don't have access to the full set of possible correct answers. Although it is not clear if a data-driven evaluation approach can fully achieve our purpose here, a better question would be: 'Does this algorithm generate a reference that a person would use?'

6 Conclusions

It is widely agreed that the requirement of numerical evaluation has benefitted the field of NLP by focussing energy on specific, well-defined problems, and has made it possible to compare competing approaches on a level playing field. In this paper, we have attempted to contribute to the debate as to how such an approach to evaluation might be brought into the field of NLG. We did this by exploring issues that arise in the evaluation of algorithms for the generation of referring expressions, since this is the area of NLG where there already seems to be something like a shared task definition.

By examining the results of our own experiments, where we have compared the outputs of existing algorithms in the literature with a collection of human-produced data, we have identified a number of key concerns that must be addressed by the community if we are to develop metrics for shared evaluation in the generation of referring expressions, and in NLG more generally.

First, it is essential that the inputs to the systems are agreed by all, particularly in regard to the nature and content of the representations used. This is a difficult issue, since NLG researchers have typically constructed their own representations that allow exploration of the research questions in their particular foci of interest; agreement on representations will not come easily. One could look to representations that exist for separately motivated tasks, thus providing an independent arbiter: for example, one might use tabular data corresponding to stock market results or meteorological phenomena. However, such representations considerably under-represent the content of texts that might describe them, leaving considerable scope for researchers to add their own special ingredients.

Second, we observe that there are many ways in which language can say the same thing or achieve the same result. Any attempt to assess the output of a language generation system has to contend with the fact that there are generally many correct answers to the problem, and there are no easy solutions to producing a reference set that contains all the possible answers. This suggests that an alternative paradigm might need to be developed for assessing the quality of NLG system output. Task-based evaluations (for example, testing if a user is able to complete a particular task given a machine-generated set of instructions) are an option to circumvent this problem, but are too coarse-grained to give us insights into the quality of the generated

output.

Finally, and related to the point above, it is not at all obvious that numeric measures like precision and recall make any sense in assessing generation system output. A generation system that replicates most or all of the outputs produced by humans, while overgenerating as little as possible, would clearly be highly adequate. However, we cannot automatically penalise systems for generating outputs that have not, so far, been seen in human-produced data.

Our analysis makes it seem likely that the impracticability of constructing a gold standard data set will prove itself as the core problem in designing tasks and metrics for the evaluation of systems for the generation of referring expressions and of NLG systems in general. There are various ways in which we might deal with this difficulty, which will need to be examined in turn. One possible way forward would be to take a more detailed look at the solutions that other tasks with output in the form of natural language, such as machine translation and text summarisation, have found for their evaluation approaches. We might also come to the conclusion that we can make do with a theoretically 'imperfect' evaluation task that works well enough to be able to assess any systems conceivably to be developed in the near or medium term.

Although we concede that a lot of groundwork still needs to be done, we are convinced that a more standardised evaluation approach is important for the advancement of the field of NLG.

References

Akiba, Y., Imamura, K., and Sumita, E. 2001. Using multiple edit distances to automatically rank machine translation output. In *Proceedings of the MT Summit VIII*, 15–20, Santiago de Compostela, Spain.

Arts, A. 2004. *Overspecification in Instructive Texts*. Ph.D. thesis, Tilburg University.

Bangalore, S., Rambow, O., and Whittaker, S. 2000. Evaluation metrics for generation. In *Proceedings of the First International Natural Language Generation Conference*, 1–8, Mitzpe Ramon, Israel.

Belz, A. and Kilgarriff, A. 2006. Shared-task evaluations in HLT: Lessons for NLG. In *Proceedings of the Fourth International Natural Language Generation Conference*, 133–135, Sydney, Australia.

Belz, A. and Reiter, E. 2006. Comparing automatic and human evaluation of NLG systems. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics*, 313–320, Trento, Italy.

Carletta, J. 1992. *Risk-taking and Recovery in Task-Oriented Dialogue*. Ph.D. thesis, University of Edinburgh.

Dale, R. and Haddock, N. 1991. Generating referring expressions involving relations. In *Proceedings of the Fifth 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. 1989. Cooking up referring expressions. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, 68–75, Vancouver, British Columbia.

Nenkova, A. and Passonneau, R. 2004. Evaluating content selection in summarization: The pyramid method. In *Main Proceedings of HLT-NAACL 2004*, 145–152, Boston, Massachusetts, USA.

Paris, C., Colineau, N., and Wilkinson, R. 2006. Evaluations of nlg systems: Common corpus and tasks or common dimensions and metrics? In *Proceedings of the Fourth International Natural Language Generation Conference*, 127–129, Sydney, Australia. Association for Computational Linguistics.

Reiter, E. and Belz, A. 2006. Geneval: A proposal for shared-task evaluation in NLG. In *Proceedings of the Fourth International Natural Language Generation Conference*, 136–138, Sydney, Australia.

Reiter, E. and Dale, R. 1992. A fast algorithm for the generation of referring expressions. In *Proceedings of the 14th International Conference on Computational Linguistics*, 232–238, Nantes, France.

Reiter, E. and Sripada, S. 2002. Should corpora texts be gold standards for NLG? In *Proceedings of the Second International Natural Language Generation Conference*, 97–104, New York, USA.

van Deemter, K. and Halldórsson, M. M. 2001. Logical form equivalence: The case of referring expressions generation. In *Proceedings of the Eighth European Workshop on Natural Language Generation*, Toulouse, France.

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 the Fourth International Natural Language Generation Conference*, 130–132, Sydney, Australia.

Viethen, J. and Dale, R. 2006. Algorithms for generating referring expressions: Do they do what people do? In *Proceedings of the Fourth International Natural Language Generation Conference*, 63–70, Sydney, Australia.