Evaluation in Natural Language Generation: Lessons from Referring Expression Generation

Jette Viethen — Robert Dale

Centre for Language Technology
Division of Information and Communication Sciences
Macquarie University
Sydney NSW 2109
Australia

jviethen@ics.mq.edu.au
robert.dale@mq.edu.au

ABSTRACT. *As one of the most well-defined subtasks in Natural Language Generation (*NLG*), the generation of referring expressions looks like a strong candidate for piloting shared evaluation tasks. Different to other areas of Natural Language Processing, it is still unclear what benefit the introduction of such tasks might have for the field of* NLG. *Based on an earlier evaluation of a number of well-established algorithms for the generation of referring expressions, this paper explores several problems that arise in designing evaluation for this task, and identifies general considerations that need to be met in evaluating Natural Language Generation subtasks.*

RÉSUMÉ. *La génération d'expressions référentielles, une des sous-tâche de la génération automatique de textes les mieux définies, apparaît comme une candidate sérieuse pour la mise en place de tâches d'évaluation partagée, dans un domaine du traitement automatique des langues où la question de l'intérêt de ces tâches reste ouverte. Sur la base des résultats d'une évaluation de certains des principaux algorithmes connus de génération d'expressions référentielles, cet article explore plusieurs problèmes posés par l'évaluation et présente quelques considérations d'ordre général à prendre en compte lors de l'évaluation des sous-tâches de la génération automatique de textes.*

KEYWORDS: *Referring Expression Generation, Natural Language Generation, Evaluation*

MOTS-CLÉS : *génération d'expressions référentielles, génération automatique de textes, évaluation*

TAL. Volume 48 – n°1/2007, pages 141 à 160

1. Introduction

How do we know that something is working well, or as it is supposed to? Whether we call the relevant activity 'testing', 'review' or 'assessment', the fundamental task to be carried out is one of *evaluation*. We measure the behaviour of the artefact against some commonly accepted benchmark, or using some commonly accepted metric. Over the last 15 years, evaluation has taken on a key role in furthering research in the field of Natural Language Processing (NLP): this testing is generally carried out by embodying theoretical hypotheses in the form of implemented software, and then evaluating the behaviour of these implemented systems against collections of test data. For many of the component subtasks that make up Natural Language Understanding (NLU), and for many standard applications of NLP, this practice has become almost institutionalised in the form of shared task evaluation campaigns (STECs). In a STEC, multiple teams will produce systems that attempt to solve a particular problem (the shared task); the systems will be developed on the basis of a set of training data, and then each is evaluated against the same previously unseen collection of test data, encouraging a degree of competition. 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 a shared-task evaluation competition.

However, not *all* subfields of NLP have adopted this model. In particular, while this approach is pursued for a number of common tasks involved in Natural Language *Understanding* (where the input is text or speech, and the output is typically some aspect of the 'meaning' of that text or speech), it has not so far been adopted as a means of progressing research in Natural Language *Generation* (where the input is some representation of information, and the output is text or speech). As this difference has become more obvious, the idea that the field might benefit from the introduction of some shared evaluation task has surfaced in a number of discussions, and most intensely at the 2006 International Natural Language Generation Conference (see, for example, (Mellish and Dale, 1998; Bangalore *et al.*, 2000; Reiter and Sripada, 2002; Reiter and Belz, 2006; Belz and Reiter, 2006; Belz and Kilgarriff, 2006; Paris *et al.*, 2006; van Deemter *et al.*, 2006)). At this point in time, the NLG community seems unsure as to how the debate should proceed.

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 the problem definition: given a domain consisting of a set of entities, one of which is the intended referent, and a knowledge base that characterises those entities in terms of the attribute–value pairs that hold true of them, how do we construct a description of the intended referent that serves to distinguish it from the other entities in the domain? 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 on 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. We believe the observations to be made in this specific case raise similar questions for evaluation in Natural Language Generation more generally.

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 our chosen test domain. This leads us to identify four 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 determining input representations; Section 4 investigates how the wide variety of acceptable outputs, and the lack of a single correct answer, makes it hard to assess generation algorithms; Section 5 explores whether we can usefully provide a numeric measure of the performance of a generation algorithm; and in Section 6 we discuss difficulties arising from the fine granularity of sub-tasks and the domain specificity found in NLG systems. Finally, in Section 7 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 the output of implemented systems 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 from the literature 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.

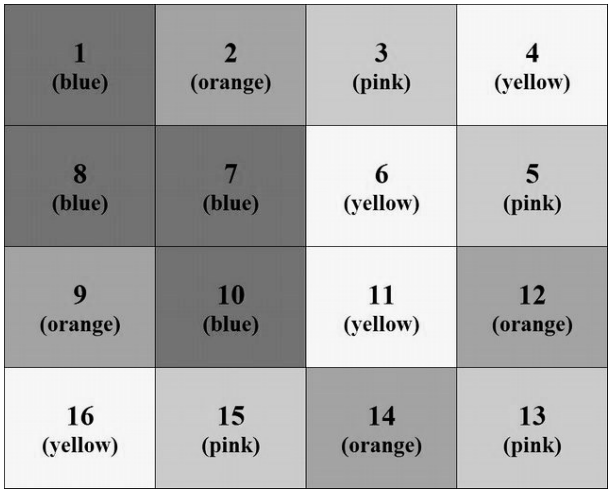

Figure 1. *An illustration of the filing cabinet domain*

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.

The human participants were given, on several 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 essentially 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 in 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 d_3 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, since we consider *in the first row, third column* and *in the top* to be semantically distinct, even if it is the case that they identify the same referent.[1] 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 exemplified 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 that expression. 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. In Table 1, the column headed Count shows the number of descriptions using each property, and the percentages indicate the ratio of the number of descriptions using each property to the number of descriptions for drawers that possess this property. For example, 27 of the descriptions referred to corner drawers, and of these, only 11 made use of the property of being in a corner to describe the drawer in question; the other 16 descriptions did not mention cornerhood. We have combined all uses of relations into one row in this table, since relational properties were used a lot less than the other properties: 103 of the 118 descriptions (87.3%) did not use relations between drawers.

1. We might think of this as being a distinction in terms of Fregean *sense*.

Property	Count	% (out of possible)
Row	95	79.66% (118)
Column	88	73.73% (118)
Colour	63	53.39% (118)
Corner	11	40.74% (27)
Relation	15	12.71% (118)

Table 1. *The properties used in descriptions*

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. Here are a few examples of descriptions that contain redundancy:

– *the yellow drawer in the third column from the left second from the top* [d_6]

– *the blue drawer in the top left corner* [d_1]

– *the orange drawer below the two yellow drawers* [d_{14}]

In the first case, either the colour property or the column property is redundant, and could be dropped without preventing the referring expression from adequately identifying its referent; in the second, colour and corner, or only the grid information, would have been sufficient; and in the third, it would have been sufficient to mention one of the two yellow drawers.

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 Greedy 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, aiming to produce a description that contains no redundant properties.

– The Relational Algorithm from (Dale and Haddock, 1991) uses constraint satisfaction to incorporate relational properties into the framework of the Greedy Algorithm. It uses a simple mechanism to avoid infinite regress (and thus prevent descriptions like *the drawer to the left of the drawer to the right of the drawer to the left of the drawer* ...).

– 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; while this allows scope for introducing redundancy into the constructed description, its script-like mode of operation is arguably more psychologically realistic. [2]

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 what 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 Greedy Algorithm is able to generate 82, providing a recall of 79.6%. The recall achieved by the Incremental Algorithm is 95.1%: it generates 98 of the 103 non-relational descriptions. 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 Greedy Algorithm and the Incremental Algorithm are able to replicate all the minimal descriptions found in the natural data. Contrary to its aim of avoiding all redundancy, the Greedy 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. On closer examination, it transpires that this failure of the Relational Algorithm to reproduce any of the descriptions from the corpus is not just incidental. In this domain, the dis-

2. The idea of using a predefined ordering over properties was already present in the earliest attempts at referring expression generation: see in particular (Winograd, 1972).

criminatory power of relational properties is generally always greater than that of any other property, so a relational property is chosen first. The poor performance of the algorithm is then compounded by its insistence on continuing to use relational properties: an absolute property will only be chosen when either the currently described drawer has no unused relational properties left, or the number of distractors has been reduced so much that the discriminatory power of all remaining relational properties is lower than that of the absolute property. Consequently, whereas a typical human description of drawer d_2 would be *the orange drawer above the blue drawer*, the Relational Algorithm will produce the description *the drawer above the drawer above the drawer above the pink drawer*. Not only are there no descriptions of this form in the human-produced data set: they also sound more like riddles someone might create to intentionally make it hard for the hearer to figure out what is meant. It is quite clear that no human would produce such a description if the sole aim was to distinguish drawer d_2 from the other drawers in the filing cabinets.

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 tasks 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*. The principle decisions to be taken are whether to work on written or spoken language and whether to restrict the input to text or speech from a certain domain. In NLG, on the other hand, we are working in the other direction: there exists no consensus regarding the exact form the input provided to the system should take. The input is generally 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 current 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 of lack of agreement. The task is generally conceived as one where the intended referent, and its distractors in the domain, are represented by symbolic iden-

tifiers, 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.

This is amply demonstrated by the experiments we have just described. As we employed a new domain for the purpose of our evaluation experiment, we had to first decide how to represent this domain. It turns out that this raises some interesting questions closely related to the functioning of the referring expression generation algorithms to be applied in the domain. Some of our representational primitives might seem to be uncontentious: the choice of colour, row and column in particular seem quite straightforward. However, we also explicitly represented a more controversial attribute position, which took the value corner for the four corner drawers (the attribute was not specified for the other drawers). Although this property, which we might refer to as 'cornerhood', can be inferred from the row and column information, we added it as an explicit property because it seems plausible to us that it is particularly salient in its own right. Of course, others might not agree with this decision.

This raises the general question of what properties should be encoded explicitly, and which should be derived by means of some process of inference. In our experiment, we explicitly encoded 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. For example, if d_1 is above d_9 and d_9 is above d_{16} , then it can be inferred that d_1 is transitively above d_{16}. Due to the uniformity of our domain the implementation of a mechanism for transitive inference could 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 is able to carry out inferences like these over knowledge-base properties, we opted here to enable a fairer comparison between human-produced and machine-produced descriptions and relied only on explicitly-encoded properties.

As some of the comments above make clear, the decisions we took regarding the representation of cornerhood, inferrable properties in general, and transitive properties, were influenced considerably by our knowledge of how the algorithms to be tested actually work. If we had only assessed different types of relational algorithms, for example, 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 row and column properties of the drawers. The point here is that our representational choices were guided by, on the one hand, the requirements of the algorithms; and on the other, by our intuitions about salience as derived from our examination of the data. Importantly, other researchers might have made different choices based on other intuitions or observations.

3.3. *Consequences for Evaluation*

From the observations above, it is evident that, in any project which focusses on the generation of referring expressions, the design of the underlying knowledge base and that of the algorithms which use this knowledge base are tightly intertwined. If we are to define a shared evaluation task or metric in this context, we seem to have two alternatives: either we can approach this from the point of view of assessing only the algorithms themselves; or we can assess algorithms in combination with their specific representations.

In the first case, clearly the input representation should be agreed on 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. If there is no agreement on what constitutes an appropriate input representation, then different algorithms and techniques cannot be compared.

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 in 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 hypothesise 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 ex-

pectations, 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) in her analysis of descriptions in the Map Task (Anderson *et al.*, 1991), 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 our data. In many cases the same participant would on different occasions produce different minimal descriptions for the same object, and the same applies for varying redundant descriptions delivered by the same participant.

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 sum-

marisation 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 could 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 we noted in Section 2.3, the performance cited above represents the behaviour of the algorithm in question *under at least one preference ordering*. To see exactly what this means requires some understanding of how the Incremental Algorithm works. The Incremental Algorithm explicitly encodes a preference ordering over the properties available to be used in descriptions, in an attempt to model what appear to be semi-conventionalised strategies for description that people use: so, for example, in describing an object in a physical scene, it is very common to first use the colour of the object, even if this property ultimately does not add anything to the discrimination provided by the other parts of the referring expression. In our implementations of the other algorithms, we also included a preference-ordering mechanism in order to force a choice in those cases where two properties rule out the same number of distractors.

In the case of the Incremental Algorithm, 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, otherwise it is skipped.[3] The use of an explicit preference ordering over properties is a way to facilitate porting the algorithm to new domains, since all one needs to do is define an appropriate ordering over the properties available in the domain.

However, even within a single domain, one can of course vary the preference ordering to achieve different effects. Dale and Reiter (1995) envisaged the use of different preference orderings for different domains; but the orderings can just as well be interpreted as personal preferences of different speakers or reflect any number of other environmental factors, such as different degrees of salience accorded to different properties by different individuals at different times. 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 for the Incremental Algorithm. 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, as discussed in Section 4. On the one hand, we are unable to penalise systems for under-generating; and on the other hand we cannot be sure whether an algorithm that exactly reproduces the data in the corpus is not just lucky. 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.

As a consequence of this discussion, it is important to note that the failure of the Relational Algorithm to reproduce the human data, as discussed earlier, does not consist in the 0% coverage it achieved in our experiment. We cannot condemn an

3. Note that the order in which properties are selected by the algorithm is deliberately independent of the order in which they appear in the surface order of the referring expression; that is, we do not assume that properties are selected in the left to right order in which they appear in the surface form, although clearly there are some questions to be explored here regarding incremental construction of descriptions.

algorithm purely on the basis of recall and precision scores. However, the Relational Algorithm's extremely low score in the metric we chose did us a limited service by pointing to a more systematic problem with the functioning of this particular algorithm. By conducting a subsequent error analysis we realised where the real failure of the algorithm lies: due to the mechanism it applies to choose properties and relations to be included in a description, as discussed in Section 2.3, it would never produce any referring expression even similar to one that a human would use—at least in our test domain, but probably also in many other natural settings.

We have argued above that asking the question 'Does the algorithm generate the correct referring expression?' 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 referring expression that a person would use?'

6. A Small Research Community Tackling a Large Field

6.1. *Fine Grained Sub-tasks*

In the experiment described in Section 2, we only assessed three different algorithms which are all aimed at producing relatively simple referring expressions. Despite the seeming agreement on the task, and the small number of algorithms tested, we encountered the problem that the algorithms are ultimately targeted at producing different *types* of referring expressions. The Greedy Algorithm and the Incremental Algorithm were never meant to produce relational descriptions; the Relational Algorithm, on the other hand, should be able to produce the same type of basic referring expressions as the Greedy Algorithm, but in addition to these should also generate relational descriptions when appropriate. Even the types of the referring expressions the Greedy and the Incremental Algorithms aim to produce can be distinguished: only the Greedy Algorithm explicitly aims to avoid the generation of redundant descriptions containing more properties than absolutely necessary to distinguish the intended referent from its distractors.

Other algorithms in the literature target other types of referring expressions. For example, some approaches are specialised on generating referring expressions for sets of objects (Gatt, 2006a; Gatt, 2006b; Funakoshi *et al.*, 2006); others concentrate on generating vague or under-specified expressions (van Deemter, 2000; van Deemter, 2006); and some even target multimodal algorithms that incorporate gestures (van der Sluis, 2005). To date, there are very few cases where a new algorithm attempts to tackle exactly the same task specification, at a detailed level, as an existing algorithm. This lack of competition to get a certain problem right by attacking it in a novel way is largely a consequence of the small size of the research community. Just as we noted in the case of the underlying representations used, on the surface there seems to be

broad agreement on the task; but as we look closer, it becomes clear that different approaches are rarely addressing the same problem.

The point here is that, just as the field of NLG can be divided into numerous subfields, the task of referring expression generation can be further subdivided into many different sub-tasks. Compared to Natural Language Understanding, the Natural Language Generation community is relatively small; this means that, for any given subtask, there is simply not the critical mass of researchers interested in a common problem that would provide a basis for a shared task competition. In fact, there is the danger that a competitive evaluation process could potentially restrict, rather than encourage, development in the field of Natural Language Generation. Any attempt to focus the attention of an already small community on optimising performance in one particular subtask would simultaneously discourage people from tackling new problems and broadening the field.

Another option for controlled evaluation, if competitive evaluation is indeed problematic, would be a purely corpus-based evaluation similar to the procedure used in the experiment described in Section 2 for referring expression generation. The performance of systems tackling different subtasks would not be evaluated on exactly the same basis and then compared to each other; rather, each system would be evaluated only against the subset of the evaluation corpus that contains instances of the kind of output the system was designed to produce. However, this approach would require a considerable amount of effort in developing the evaluation corpus to ensure that it meets the requirements of different subtasks.

6.2. *Domain Specificity*

Early algorithms for the generation of referring expressions, such as those evaluated in the experiment described above, are very rarely formally tested or even developed on the basis of a solid data set of human descriptions of objects. In the reported literature, the closest this work comes to an evaluation is to sketch a few worked examples, typically from a simple toy domain. These mini-domains usually consist of not more than a few objects: a couple of bowls, cups and tables, or a few animals of different types, sizes and colours.

Some more recent approaches use production experiments involving human participants for the development or evaluation of their algorithms. The algorithm presented by Funakoshi *et al.* (2004) is based on the analysis of human data obtained from experiments in a handcrafted domain. van der Sluis and Krahmer (2004a) and van der Sluis and Krahmer (2004b) draw on production experiments to verify assumptions made by the algorithm they describe in (Krahmer and van der Sluis, 2003). Gatt (2006b) reports on the only research we know of in the area of referring expression generation where algorithm performance is directly compared to human performance. However, it is not the referring expressions themselves, but the underlying clustering of objects that is at the centre of interest in this work.

In all cases, the domains on which the assessment is based are handcrafted and highly artificial, often only involving geometric shapes. Although cautious claims are made regarding the portability of the algorithms to other domains, these are never tested. Ultimately, most algorithms for the generation of referring expressions are designed with a certain domain in mind; if they are systematically tested at all, then it is on this one domain and against data from experiments in the same domain.

The surprisingly bad results of the Relational Algorithm in our evaluation experiment, as discussed above, show that this domain specificity of algorithms for the generation of referring expressions makes it extremely hard to compare existing approaches. While the Relational Algorithm might perform well in the toy domain used for the worked examples in (Dale and Haddock, 1991), it never had a chance in our still relatively simple real-world domain. With hindsight, it becomes obvious that the toy domain used in that work is not well-suited for testing the ability of the algorithm to choose between relational and non-relational properties in the way people do. The only non-relational property in the domain used in (Dale and Haddock, 1991) is the type of the objects, which is added in all cases to provide a head for the nominal expressions produced. Consequently, the only way to make a distinction between objects of the same type, for a human speaker or for the algorithm, is to use spatial relations.

The problem of an implicit domain specificity in approaches to referring expression generation is one of the reasons to argue for a shared test domain. Researchers developing a new algorithm, or hoping to improve an existing algorithm, are only able to verify their advances if they can compare old and new systems in a controlled test environment.

However, this issue also points to the implausibility of 'blind development' for an evaluation competition in GRE where the test domain is only revealed after development is concluded. This is common practice in other shared evaluation task communities; but the fundamental differences between Natural Language Understanding and Natural Language Generation mean that we are still far from being able to develop any kind of NLG system that is portable to a new domain without considerable effort.

7. Conclusions

Many would agree that the requirement of comparative 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 that might circumvent this problem; but evaluations of this kind are mostly too coarse-grained to allow us to assess specific selections of semantic content in referring expressions.

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.

Finally, when discussing shared evaluation schemes for Natural Language Generation, it is important to keep in mind that NLG is a vast research field and can be subdivided into numerous subtasks which are worked on by a comparatively small research community. As a result, new approaches are highly domain specific and rarely aimed at exactly the same task. A shared evaluation campaign will need to circumvent the danger of concentrating too many resources on one subtask or one domain.

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. One way forward would be to examine more closely the solutions that other tasks with output in the form of natural language, such as machine translation and text summarisation, have adopted in their evaluation exercises. 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. Careful construction of an extensive corpus of, for example, referring expressions might allow us to use this corpus as a gold standard after all, as long as we keep in mind that no corpus of natural language expressions can ever be complete. Another possibility might be to introduce semi-supervised evaluation metrics which are not entirely data-driven; whether this is practicable remains to be seen.

We are convinced that a more standardised approach to evaluation is required in order to advance research in Natural Language Generation. As we have argued in this paper, the inherently different nature of NLG tasks compared to other Natural Language Processing tasks makes a straightforward transfer of established schemes for shared evaluation impossible; there are difficult issues that must first be addressed before this goal can be achieved. However, as we noted earlier, discussion of these issues is now firmly on the stage. We hope that this article has served to move that discussion forward.

8. References

Akiba Y., Imamura K., Sumita E., "Using Multiple Edit Distances to Automatically Rank Machine Translation Output", *Proceedings of the MT Summit VIII*, Santiago de Compostela, Spain, p. 15-20, 2001.

Anderson A., Bader M., Bard E., Boyle E., Doherty G., Garrod S., Isard S., Kowtko J., McAllister J., Miller J., Sotillo C., Thompson H., Weinert R., "The HCRC Map Task Corpus", *Language and Speech*, vol. 34, p. 351-366, 1991.

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

Bangalore S., Rambow O., Whittaker S., "Evaluation Metrics for Generation", *Proceedings of the First International Natural Language Generation Conference*, Mitzpe Ramon, Israel, p. 1-8, June, 2000.

Belz A., Kilgarriff A., "Shared-Task Evaluations in HLT: Lessons for NLG", *Proceedings of the Fourth International Natural Language Generation Conference*, Sydney, Australia, p. 133-135, July, 2006.

Belz A., Reiter E., "Comparing Automatic and Human Evaluation of NLG Systems", *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, p. 313-320, 3-7 April, 2006.

Carletta J., Risk-taking and Recovery in Task-oriented Dialogue, PhD thesis, University of Edinburgh, 1992.

Dale R., "Cooking up Referring Expressions", *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, Vancouver, British Columbia, p. 68-75, 1989.

Dale R., Haddock N., "Generating Referring Expressions Involving Relations", *Proceedings of the Fifth Conference of the European Chapter of the ACL*, Berlin, Germany, p. 161-166, 1991.

Dale R., Reiter E., "Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions", *Cognitive Science*, vol. 19, n° 2, p. 233-263, 1995.

Funakoshi K., Watanabe S., Kuriyama N., Tokunaga T., "Generating referring expressions using perceptual groups", *Proceedings of the Third International Conference on Natural Language Generation*, Brighton, UK, p. 51-60, 2004.

Funakoshi K., Watanabe S., Tokunaga T., "Group-Based Generation of Referring Expressions", *Proceedings of the Fourth International Conference on Natural Language Generation*, Sydney, Australia, p. 73-80, July, 2006.

Gatt A., "Generating collective spatial references", Vancouver, Canada, 2006a.

Gatt A., "Structuring Knowledge for Reference Generation: A Clustering Algorithm", *Proceedings of the 11th Meeting of the European Chapter of the ACL*, Trento, Italy, p. 321-328, 2006b.

Krahmer E., van der Sluis I., "A New Model for Generating Multimodal Referring Expressions", Budapest, Hungary, p. 47-57, 2003.

Mellish C., Dale R., "Evaluation in the Context of Natural Language Generation", *Computer Speech and Language*, vol. 12, p. 349-373, 1998.

Nenkova A., Passonneau R., "Evaluating Content Selection in Summarization: The Pyramid Method", *Main Proceedings of HLT-NAACL 2004*, Boston, Massachusetts, USA, p. 145-152, 2004.

Paris C., Colineau N., Wilkinson R., "Evaluations of NLG Systems: Common Corpus and Tasks or Common Dimensions and Metrics?", *Proceedings of the Fourth International Natural Language Generation Conference*, Association for Computational Linguistics, Sydney, Australia, p. 127-129, July, 2006.

Reiter E., Belz A., "GENEVAL: A Proposal for Shared-task Evaluation in NLG", *Proceedings of the Fourth International Natural Language Generation Conference*, Sydney, Australia, p. 136-138, July, 2006.

Reiter E., Dale R., "A Fast Algorithm for the Generation of Referring Expressions", *Proceedings of the 14th International Conference on Computational Linguistics*, Nantes, France, p. 232-238, 1992.

Reiter E., Sripada S., "Should Corpora Texts be Gold Standards for NLG?", *Proceedings of the Second International Natural Language Generation Conference*, New York, USA, p. 97-104, July, 2002.

van Deemter K., "Generating Vague Descriptions", *Proceedings of the First International Conference on Natural Language Generation*, p. 179-185, 2000.

van Deemter K., "Generating Referring Expressions that Involve Gradable Properties", *Computational Linguistics*, vol. 32, n° 2, p. 195-222, 2006.

van Deemter K., Halldórsson M. M., "Logical Form Equivalence: The Case of Referring Expressions Generation", *Proceedings of the Eighth European Workshop on Natural Language Generation*, Toulouse, France, p. 1-8, 2001.

van Deemter K., van der Sluis I., Gatt A., "Building a Semantically Transparent Corpus for the Generation of Referring Expressions.", *Proceedings of the Fourth International Natural Language Generation Conference*, Sydney, Australia, p. 130-132, July, 2006.

van der Sluis I., Multimodal Reference, Studies in Automatic Generation of Multimodal Referring Expressions, PhD thesis, Tilburg University, 2005.

van der Sluis I., Krahmer E., "Evaluating Multimodal NLG using Production Experiments", *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, Lisbon, Portugal, p. 209-212, 26-28 May, 2004a.

van der Sluis I., Krahmer E., "The influence of target size and distance on the production of speech and gesture in multimodal referring expressions", *Proceedings of the Eighth International Conference on Spoken Language Processing*, Jeju, Korea, p. 1005-1008, 4-8 October, 2004b.

Viethen J., Dale R., "Algorithms for Generating Referring Expressions: Do They Do What People Do?", *Proceedings of the Fourth International Natural Language Generation Conference*, Sydney, Australia, p. 63-70, July, 2006.

Winograd T., *Understanding Natural Language*, University of Edinburgh Press, 1972.