Natural Language Engineering **23** (6): 813–834. © Cambridge University Press 2017
doi:10.1017/S1351324917000079

Generating natural language descriptions using speaker-dependent information†

THIAGO C. FERREIRA[1] and IVANDRÉ PARABONI[2]

[1]*Tilburg center for Cognition and Communication, Tilburg University*
P.O. Box 90135, 5000 LE Tilburg, the Netherlands
e-mail: tcastrof@tilburguniversity.edu
[2]*School of Arts, Sciences and Humanities, University of São Paulo*
Av. Arlindo Bettio, 1000 - São Paulo, Brazil
e-mail: ivandre@usp.br

(*Received 24 December 2015; revised 16 January 2017; accepted 16 January 2017; first published online 27 February 2017*)

Abstract

This paper discusses the issue of human variation in natural language referring expression generation. We introduce a model of content selection that takes speaker-dependent information into account to produce descriptions that closely resemble those produced by each individual, as seen in a number of reference corpora. Results show that our speaker-dependent referring expression generation model outperforms alternatives that do not take human variation into account, or which do so less extensively, and suggest that the use of machine-learning methods may be an ideal approach to mimic complex referential behaviour.

1 Introduction

In Natural Language Generation systems, the computational task of referring expression generation (REG) concerns the production of linguistic forms that uniquely describe – or refer to – a given target object (Krahmer and van Deemter 2012). REG usually involves two subtasks: content selection (i.e., deciding 'what to say'), and surface realisation (i.e., deciding 'how to say it' in a target language) (Hervás *et al.* 2016). In this paper, we will focus on the former, discussing how to determine the semantic contents of definite descriptions as in 'the man sitting next to the door', 'the large black hound', 'the small white church on 15th st.', etc.

REG involves, at the very least, producing a description that enables identification, allowing the hearer to distinguish the intended target from the distractor objects found in the same context with no risk of ambiguity. More recently, however, a number of additional requirements have been considered, including the issue of 'humanlikeness' of the generated expressions (Gatt and Belz 2007). Clearly,

†This work has been supported by CAPES and FAPESP. The authors are also grateful to the anonymous reviewers for their valuable comments.

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press

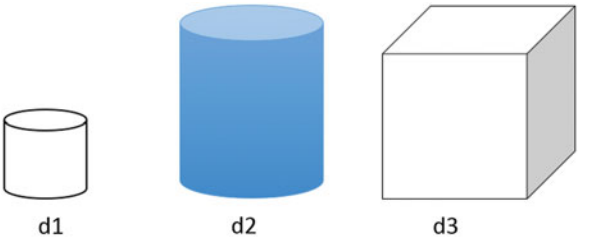


Fig. 1. (Colour online) A simple visual context. Objects d_1 and d_3 are white, and object d_2 is blue.

by generating descriptions that more closely resemble those produced by human speakers, an REG algorithm may help increase the overall realism of the underlying Natural Language Generation system as well.

Let us consider the goal of referring to d_1 in the visual context illustrated in Figure 1.

In this context, examples of unambiguous descriptions of the target d_1 include 'the small cylinder', 'the white cylinder, on the left side', 'the small cylinder next to the large one', and so on. But which one is the 'best' or more 'human-like' alternative?

Research in cognitive sciences and related fields have identified a wide range of factors that may influence content selection. These include, for instance, the need to be concise (Grice 1975), the preference for vertical relations (above/below) (Arts *et al.* 2011), the inherent preference for certain attributes (Gatt *et al.* 2013), and many others, which have been encoded as hand-crafted rules to compose a number of well-known REG algorithms (Dale 1989; Dale and Reiter 1995; Krahmer, van Erk and Verleg 2003).

Leaving the major factors that may influence content selection aside, however, content selection is known to be heavily influenced by human variation (Viethen and Dale 2010). In other words, under identical circumstances (i.e., in the same referential context), different speakers will often produce different descriptions. Human variation in this case is observed both as (a) the choice of attributes to appear in a referring expression (e.g., the choice between 'the small cylinder' and 'the white cylinder' in the previous example), and as (b) the level of referential specification (e.g., the choice between a minimally distinguishing description as 'the small cylinder' and an overspecified alternative as in 'the small cylinder, *on the left side*').

Although pervasive in language production, the issue of human variation is little explored by existing REG algorithms, most of which will generate the same, fixed output description from a given input context. In the few cases in which speaker-related information has been taken into account, this is often limited to the use of preferred (e.g., most frequent) attributes for each individual speaker (Gupta and Stent 2005; Bohnet 2008; Fabbrizio, Stent and Bangalore 2008; Viethen and Dale 2010). Strategies of this kind are of course sufficient for many practical purposes, but for applications that are required to be more psychologically-plausible (for instance,

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press

if we are to model more human-like artificial agents), it may be necessary to mimic human behaviour at a more fine-grained level.

In the present work, we discuss the issue of human variation in REG by introducing models of content selection that take speaker-dependent information into account. These models will consider the referential behaviour of individual speakers regarding both attribute choice and level of specification to produce descriptions that closely resemble those produced by each individual. Our primary goal is to show that speaker-dependent REG models outperform alternatives that do not take human variation into account, or which do so less extensively. Moreover, since many aspects of a speaker-dependent referential behaviour are difficult to encode as hand-crafted rules, our work will rely heavily on the use of machine-learning methods and, as a secondary goal, we also intend to show that the use of machine learning may be an ideal approach to mimic complex referential behaviour.

2 Background

In this section, we outline the basic REG task and review a number of algorithms that make use of speaker-dependent information. As elsewhere in this paper, our focus is on the content selection task. The related issue of surface realisation is briefly discussed at the end of the section.

2.1 Referring expression generation (REG)

The Incremental algorithm (Dale and Reiter 1995) is the basis of a wide range of approaches to REG. The input to the algorithm is a target object r to be uniquely described within a context C containing a set of distractors. All objects are modelled as sets of properties represented as (*attribute-value*) pairs, as in (*size-small*). The goal of the algorithm is to produce a set L of properties of r, such that L distinguishes r from every distractor in C.

Attributes are considered for inclusion in L incrementally by following a domain-dependent list of preferred attributes P, and provided that they help disambiguate the target. When an attribute a is selected for inclusion in L, the corresponding distractors are removed from C. For instance, selecting a property (*size-small*) rules out all distractor objects whose size is *not* small. Thus, in the context in the previous Figure 1, given a preferred order as, e.g., $P = \langle type, size, colour \rangle$, the target $r = d_1$ would be described as 'the small cylinder'. Similarly, $r = d_2$ would be described as 'the large cylinder' and $r = d_3$ as 'the box'.

The algorithm terminates when C becomes empty, or when all attributes in P have been attempted. In the former, L represents a uniquely identifying attribute set that may be realised as, e.g., a definite description as in 'the small cone'. In the latter, L may be realised as, e.g., an indefinite description, as in 'a cone'.

The order of attributes in P may greatly impact the outcome of the algorithm. For instance, had we considered the preference order $P = \langle colour, type, size \rangle$ in the

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press

above example, d_1 would have been described as 'the white cylinder', d_2 as 'the blue (cylinder)'[1] and d_3 as 'the white box'.

The use of discriminatory properties favours the generation of brief descriptions in accordance to the Gricean maxim of quantity (Grice 1975). For similar reasons, the Full Brevity algorithm (Dale 1989) aims to produce minimally distinguishing descriptions by first attempting solutions conveying one single property, followed by possible alternatives conveying two properties, and so on. Examples of minimally distinguishing descriptions for the objects $d_1..d_3$ in the previous example are 'the small (object)', 'the blue (object)' and 'the box'.

In addition to the Incremental algorithm, research in the field has produced a wide range of approaches to REG. These include, for instance, the generation of boolean referring expressions (van Deemter 2002), the use of graphs to generate relational descriptions (Krahmer *et al.* 2003), the use of lexical information to generate descriptions in open domains (Siddharthan and Copestake 2004), and the generation of overspecified descriptions to facilitate search (Paraboni and van Deemter 2013), among many others. In the series of REG shared tasks (Gatt and Belz 2007; Gatt, Belz and Kow 2008, 2009) a large number of algorithms have been proposed as well, including the 2007 best-performing IS-FBN algorithm (Bohnet 2007) and the 2008 winner JU-PTBSGRE (Paladhi and Bandyopadhyay 2008). For a comprehensive review of studies of this kind, we refer to Krahmer and van Deemter (2012).

2.2 Human variation in REG

Existing REG algorithms, including the Incremental and Full Brevity approaches described in the previous section, generally do not take speaker-related information into account. In other words, given a certain input (i.e., a context C and a target r) these algorithms always produce the same output description. A number of exceptions are discussed below.

In Bohnet (2008), extensions of the Incremental (Dale and Reiter 1995) and Full Brevity (Dale 1989) algorithms are evaluated using the TUNA corpus of referring expressions (Gatt, van der Sluis and van Deemter 2007). In the Incremental extension, human variation is accounted for by computing individual preference lists (the P parameter of the algorithm, cf. the previous section) from the subset of descriptions produced by each speaker. Attributes in each individual list are ordered by frequency. Results in Bohnet (2008) show that the use of individual preference lists outperforms the use of a single preference order computed from the entire training data. In the Full Brevity extensions, human variation is accounted for by selecting the description that is closest to the previous descriptions produced by the same speaker.

The work in Fabbrizio, Stent and Bangalore (2008) also makes use of the Full Brevity (Dale 1989) and Incremental (Dale and Reiter 1995) algorithms to generate

[1] The Incremental algorithm always adds the basic *type* to the description in order to facilitate its subsequent realisation as a noun phrase.

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press

descriptions from the TUNA corpus. In the case of Full Brevity, human variation is accounted for by making use of attribute set frequencies and recency estimates for each speaker. In the case of the Incremental algorithm, human variation is implemented as in Bohnet (2008), that is, by computing individual preference lists for each speaker. In both cases, results show that the use of speaker-related information improves accuracy.

The work in Viethen and Dale (2010) does not present a REG algorithm proper, but focuses instead on the use of decision-tree induction to predict content patterns (i.e., full attribute sets representing actual referring expressions) from visual scenes in the GRE3D3/7 corpora (Dale and Viethen 2009; Viethen and Dale 2011). Human variation is accounted for by including speaker identifiers in the model. Results from Viethen and Dale (2010) show that the classifier that has access to speaker's identifiers outperforms those that do not. This approach is furthered in Viethen (2011) by modelling individual classifiers for each attribute under consideration (i.e., as opposed to using a single classifier to predict each attribute set, in Viethen and Dale (2010)). Once again, human variation is limited to the use of speaker's identifiers, and the classifiers that include speaker's IDs outperform those that do not.

In the related task of REG in dialogues (i.e., taking the previous discourse into account), the work in Gupta and Stent (2005) proposes a number of variations of the Incremental algorithm to generate descriptions from the Coconut (Eugenio *et al.* 2000) and MapTask (Anderson *et al.* 1991) corpora. Two speaker-dependent strategies are considered. The first strategy uses a preference list ordered according to attribute frequencies observed in recent expressions produced by each speaker. In addition to that, the second strategy includes the attributes mentioned in the latest reference produced by each speaker when generating a new description regardless of its frequency or discriminatory power. The study shows that both strategies outperform a standard REG approach that does not take human variation into account.

In Viethen, Mitchell and Krahmer (2013), an extension of the GRAPH algorithm (Krahmer *et al.* 2003) called *Longest First* is presented. The algorithm attempts to find the longest (up to a certain limit) possible description for a given input, a strategy that favours the generation of overspecified descriptions that would not be normally obtainable by using the original GRAPH algorithm. A speaker-dependent variation of this approach divides the training data into two clusters of speakers: one cluster containing descriptions produced by individuals that prefer longer referring expressions, often making use of spatial relations, and another containing descriptions produced by those who prefer short, non-relational expressions. This information is taken into account to compute two attribute preference orders (i.e., one for each cluster). The study shows that the use of the cluster-dependent preference order outperforms the use of a single (i.e., global) preference order in the GRE3D3 domain (Dale and Viethen 2009).

Finally, a note on human choice in surface realisation. Unlike our current work, which is solely focused on content selection, some of the existing studies (Bohnet 2008; Fabbrizio, Stent and Bangalore 2008; Hervás, Francisco and Gervás 2013)

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press

address not only content selection, but also the surface realisation of referring expressions. In particular, the work in Hervás, Francisco and Gervás (2013) addresses the issue of personal preferences in reference lexicalisation. By presenting an analysis of the descriptions in the TUNA corpus, the study shows that individuals tend to be consistent with themselves when making lexical choices and, accordingly, a lexicalisation algorithm that uses speaker-specific training data is shown to outperform the same algorithm when using training data produced by all speakers in the corpus.

2.3 Discussion

Existing studies on speaker-dependent REG are generally limited to the use of individual preferences regarding attribute selection (Gupta and Stent 2005; Bohnet 2008; Fabbrizio, Stent and Bangalore 2008) or the use of speaker's IDs for attribute classification only (Viethen and Dale 2010; Viethen 2011). These and other speaker-related features will be considered in the REG speaker model proposed in Section 4, but before that we will discuss two baseline approaches to REG that do not take speaker-related information into account.

3 Baseline REG models

In this section, we introduce two alternative approaches to REG: a standard, Incremental-like algorithm (Section 3.1) and a machine learning approach that makes use of a series of classifiers (Section 3.2). Both are intended as baseline systems to be compared against our speaker-dependent REG models in Section 4.

3.1 Extended incremental algorithm (EIA)

The original Incremental algorithm (Dale and Reiter 1995) does not explicitly handle relational properties (e.g., as in 'the cylinder *next to the box*'). An extension of this kind is in principle not difficult to envisage (Dale and Haddock 1991; Krahmer *et al.* 2003; Dale and Viethen 2009; dos Santos Silva and Paraboni 2015), but it does give rise to the issue of how to combine relations with ordinary atomic (e.g., 'the blue cylinder') properties. For instance, the work in Dale and Haddock (1991) makes use of relational properties only as a last resort, that is, when it is not possible to obtain a uniquely identifying set of atomic descriptions. Other studies as Dale and Viethen (2009), by contrast, suggest that relational properties are far more common, and may be included in the output description even when they do not help rule out distractors.

In what follows, we describe a relational extension to the basic incremental approach, hereby called Extended Incremental Algorithm (EIA). EIA combines atomic and relational properties as follows. First, individual lists P_{atom} and P_{rel} are computed for atomic and relational properties. Both lists are sorted by frequency as seen in the training data (to be discussed in the next section), starting with the most frequent attribute. Next, the overall frequency of relations F_{rel} is computed as the

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press

sum of all frequencies in P_{rel}. Finally, a combined list P of atomic and relational properties is created by inserting the entire list P_{rel} in P_{atom} at the position given by F_{rel}. The list P is taken as an input to the algorithm as illustrated in Algorithm 1.

```
1  Algorithm getDescription(r, L, D, P, H)
2  L[r] ← {}                                              // output description
3  C ← D − {r}                                            // context set
4  H ← H ∪ r                                              // history
5
6  for A_i ∈ P do
7  |  V ← value(r, A_i)
8  |  if |rulesOut(⟨A_i, V⟩)| > 0 then
9  |  |  if V ∉ D then
10 |  |  |  L[r] ← L[r] ∪ ⟨A_i, V⟩                       // atomic property
11 |  |  |  C ← C − rulesOut(⟨A_i, V⟩)
12 |  |  end
13 |  |  else
14 |  |  |  if V ∉ H then
15 |  |  |  |  L[r] ← L[r] ∪ ⟨A_i, V⟩                    // relational property
16 |  |  |  |  C ← C − rulesOut(⟨A_i, V⟩)
17 |  |  |  |  L ← getDescription(V, L, D, P, H)
18 |  |  |  end
19 |  |  end
20 |  end
21 |  if |C| == 0 then
22 |  |  return L
23 |  end
24 end
25 return L
```

Algorithm 1: Extended Incremental Algorithm (EIA)

EIA works as follows. The algorithm receives as an input the target object r, an initially empty set L representing the description to be generated (recursively), the set D of all domain objects, a list P of preferred attributes, and an initially empty history list H of objects already mentioned in L. The output of the algorithm is the updated set L representing a uniquely distinguishing description of r with respect to D.

The algorithm starts by making the individual description of the current target r an empty set (line 2), making the referential context set C equal to the set of all distractor objects (3), and adding the current target to the history list (4).

As in Dale and Reiter (1995), the algorithm iterates over the list of preferred attributes P (line 6) and considers only those that have discriminatory power (7–8). To this end, the *rulesOut* function is assumed to return the set of objects that would be removed from C by the inclusion of the current property in L.

Ordinary atomic properties – e.g., (*colour-red*) – are dealt with simply by adding the property to the current description (10), and by updating the context set (11). Relational properties – those in which the attribute value V under consideration corresponds to a domain object identifier, as in (*left-*d_1) – are dealt with in lines 14–17. Provided that the intended object V is not yet in the previous history (14), the corresponding property is added to the current description (15), the context set is

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press

updated accordingly (16), and the algorithm is called recursively in order to describe V itself (17).

The procedure is repeated until a uniquely identifying description is obtained (21–22), or until all possible attributes in P have been attempted (6–24). In the former, L could be subsequently realised as a definite description (e.g., '*the* red cone'), whereas in the latter L could be realised as an indefinite description (e.g., '*a* red cone').

3.2 Machine learning REG

Computational REG has been traditionally addressed by making use of knowledge-based models, such as the Incremental algorithm (Dale and Reiter 1995) and others. In recent years, however, the increasing availability of REG corpora has enabled a number of data-driven approaches to the task, in some cases making use of machine-learning methods (Bohnet 2007; Viethen and Dale 2010; Viethen 2011; Viethen, Mitchell and Krahmer 2013). In this section, we will focus on the latter by presenting a REG algorithm based on a series of classifiers. An earlier version of this approach appeared in Ferreira and Paraboni (2014a) using a different (namely, multiclass) set of classifiers.

As in Viethen (2011), we will make use of individual classifiers to decide whether each referential attribute of a given target object r should be selected to appear in an output description of r. Unlike Viethen (2011), however, our current work is not limited to making attribute predictions: we will go one step further and combine the output of the classifiers to produce actual referring expressions.

Let D be the set of objects in a given context, and let A be the set of (atomic and relational) attributes associated to them. Given the goal of producing a description L of a target object $r \in D$, we define a set of $|A|$ binary classifiers $C = \{c^{(1)}, c^{(2)}, \ldots, c^{(|A|)}\}$ so that each classifier $c^{(i)} \in C$ predicts whether an attribute $a^{(i)} \in A$ should appear in L or not.

Individual classifiers are built from a training set of object descriptions $E = \{e^{(1)}, e^{(2)}, \ldots, e^{(|E|)}\}$. Given the subsets A_{atom} and A_{rel} of atomic and relational attributes in A, respectively, a training description $e \in E$ is represented as $e = \{X, Y_{atom}, Y_{rel}\}$. In this set, X is an n-dimensional feature vector. Y_{atom} is the atomic class set $\{y^1, y^2, ..., y^{|atom|}\}$, in which $y^k = 0$ when the atomic attribute $a^k \in A_{atom}$ does not belong to e, and $y^k = 1$ otherwise. In the same way, Y_{rel} is the relational class set, in which $y^k = 0$ when the relation $a^k \in A_{rel}$ does not belong to e, and $y^k = 1$ otherwise.

3.2.1 Classifiers

We built a series of binary classifiers to predict the use of every atomic and relational property in each domain (e.g., whether to use *colour*, *size*, *above,* etc.). As the definitions of the classifiers are strongly domain-dependent, our approach makes use of slightly different sets of classifiers and feature vectors for each of our six test domains, namely: TUNA Furniture and TUNA People (Gatt *et al.* 2007), GRE3D3

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press

(Dale and Viethen 2009), GRE3D7 (Viethen and Dale 2011), Stars (Paraboni *et al.* 2014), and Stars2 (Paraboni, Galindo and Iacovelli 2016). These are discussed below.

Descriptions in the TUNA corpus do not convey relational properties. For that reason, only atomic classifiers were defined for this domain. We produced six classifiers for the attributes in the Furniture domain (*type*, *colour*, *size*, *orientation*, *x-dimension,* and *y-dimension*), and further twelve classifiers for the People domain (*type*, *orientation*, *age*, *hair-colour*, *has-beard*, *has-hair*, *has-glasses*, *has-shirt*, *has-tie*, *has-suit*, *x-dimension,* and *y-dimension*).

In GRE3D3/7, eight atomic classifiers (*type*, *colour*, *size,* and *location*) were defined, being four classifiers corresponding to attributes of the main target object, and four classifiers corresponding to attributes of the nearest landmark object. Relational properties in GRE3D7 are accounted for by four classifiers *right-of*, *left-of*, *next-to*, *on-top-of*. In the case of GRE3D3, there is an additional *in-front-of* classifier as well.

Finally, in Stars/2 there is the additional issue of reference involving up to three objects, as in 'the box next to the sphere, near the cone'. As a result, we defined thirty-one binary classifiers (fourteen for Stars and seventeen for Stars2) corresponding to the atomic attributes of the main target, first and second landmark objects (*type*, *colour*, *size*, *vpos*, *hpos*).[2] Moreover, relational classifiers were defined for each corpus in order to predict spatial relations between target and first landmark, and also between first and second landmarks, if any. For the Stars domain, we defined four classifiers for the target-landmark relation, namely *right-of*, *left-of*, *next-to*, *on-top-of*, and the same for the landmark-second landmark relation. In the case of Stars2, two additional relations are considered, namely, *in-front-of* and *behind-of*.

3.2.2 Context features

The input to the classifiers described in the previous section consists of feature vectors extracted from the referential contexts in each domain. These features – hereby called context features – are intended to model the target and landmark objects (if any), and similarities between objects. Additional features that take the issue of human variation into account (hereby called speaker-dependent features) will be discussed in Section 4.

In the case of the TUNA-Furniture corpus, we defined context features to model the number of distractors that share the same *type*, *colour*, *size*, *orientation*, *x-dimension,* and *y-dimension* attributes with the target. For TUNA-People, we defined context features to model the number of distractors that share the same *type*, *orientation*, *age*, *hair-colour*, *has-beard*, *has-hair*, *has-glasses*, *has-shirt*, *has-tie*, *has-suit*, *x-dimension,* and *y-dimension* attributes with the target. We also defined a binary location (LOC) feature representing the experimental condition in which TUNA subjects were told that they could refer to screen coordinates if they wished to do so (Gatt *et al.* 2007).

[2] Some attributes may occur either in Stars or Stars2, but not necessarily in both.

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press

In the case of GRE3D3/7 and Stars/2, we defined context features for both target and landmark objects. As in Viethen (2011), we defined context features to model the number of distractors that share the same *type*, *colour,* and *size* with the target or landmark objects. In addition to that, we also defined context features to model *type*, *colour,* and *size* similarities between target and landmark. Finally, the size of the target and landmark objects (*big* or *small*), and the spatial relation between them (*vertical* or *horizontal*) were also modelled as context features. Objects in the Stars domain do not vary in size, and for that reason size-related features were not modelled in this case.

3.2.3 Combining predictions

The predictions made by the atomic and relational classifiers for a given context are combined to produce a referring expression according to Algorithm 2.

```
1  Algorithm getDescription(r, L, D, H)
2  L[r] ← {}                                              // output description
3  H ← H ∪ r                                                        // history
4
5  level ← |H|                                    // target, 1st or 2nd landmark
6  Pr_atom ← getPredictions(level)                              // atomic classifiers
7  Pr_rel ← getSingleRelationPrediction(level)                  // preferred relation
8
9  for A_i ∈ Pr_atom do
10 |   if Pr_atom[A_i] == 1 then
11 |   |   L[r] ← L[r] ∪ ⟨A_i, value(r, A_i)⟩                    // atomic properties
12 |   end
13 end
14 if Pr_rel ≠ null then
15 |   lm ← value(r, Pr_rel)
16 |   if lm ≠ null and lm ∉ H then
17 |   |   L[r] ← L[r] ∪ ⟨rel, lm⟩                          // relational property
18 |   |   L ← getDescription(lm, L, D, H)
19 |   end
20 end
21 return L
```

Algorithm 2: Classification-based REG

The algorithm receives as an input the object r to be referred to, an initially empty set L representing the description to be generated (recursively), the set of all domain objects D, and an initially empty history list H of objects already mentioned in L. Unlike EIA (cf. Section 3.1), the present algorithm does not take any preferred order of atomic properties into account, since all atomic properties that are predicted by the classifiers will be included in L (and a maximum of one relational property per object). The output of the algorithm is the updated set L representing a description of r with respect to D, and which may or may not be uniquely distinguishing.

The algorithm starts by making the individual description of the current object r an empty set (2), and by including it in the history list (3). Based on the number of objects already included in the history list, the current level of description

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press

is determined (5). The description level equals 1 when r corresponds to the main target, 2 when r corresponds to the first landmark object, and so on. This information is then taken into account to invoke the appropriate set of classifiers (6–7), which are implemented by the auxiliary functions *getPredictions* and *getSingleRelationPrediction*. The former is assumed to invoke the set of atomic classifiers described in Section 3.2.1 to create the Pr_{atom} vector containing the predictions for every atomic attribute of r, whereas the latter invokes the set of relational classifiers and returns the first relation found to hold between the current target r and the nearest landmark, in order of frequency as seen in the training data. If no such relation is predicted, *getSingleRelationPrediction* is assumed to return a null value.

The algorithm iterates over the list of possible atomic properties (9–13) and considers only those that are predicted by the corresponding classifier (10). All selected properties are added to the current description, one by one (11).

Relational properties are dealt with in lines 14–20. If a relation is predicted (14), and provided that the intended landmark object *lm* is not yet in the previous history (16), the corresponding property is added to the current description (17), and the algorithm is called recursively in order to describe *lm* (18).

A complete example works as follows. Let us consider once again the domain in the previous Figure 1, the target $r = d_2$ and the following predictions for the target object (i.e., for a description of level 1):

$$getPredictions(1) = \{\langle\ type,\ 1\ \rangle, \langle\ size,\ 0\ \rangle, \langle\ colour,\ 0\ \rangle\}$$

$$getSingleRelationPrediction(1) = \{\langle\ left\text{-}of,\ d_3\rangle\}$$

The above predictions would select *type* and *left-of* for inclusion in the output L, producing the following target description:

$$L = \{d_2 = \{\langle\ type,\ cylinder\ \rangle, \langle\ left\text{-}of,\ d_3\rangle\}\}$$

Next, the algorithm is called recursively to describe the landmark d_3. Let us assume that the following (level 2) predictions were made:

$$getPredictions(2) = \{\langle\ type,\ 1\ \rangle, \langle\ size,\ 0\ \rangle, \langle\ colour,\ 1\ \rangle\}$$

$$getSingleRelationPrediction(2) = \{\langle\ null\rangle\}$$

In this case, both *type* and *colour* would be selected to compose the landmark portion of the description. The final description – which could be realised as, e.g., *"The cylinder on the left side of the white box"*, is represented as follows:

$$L = \{d_2 = \{\langle\ type,\ cylinder\ \rangle, \langle\ left\text{-}of,\ d_3\rangle\}, d_3 = \{\langle\ type,\ box\ \rangle, \langle\ colour,\ white\ \rangle\}\}$$

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press

Unlike EIA, in the present approach there is no explicit test for uniqueness, and properties are added to L regardless of their discriminatory power. In practice, however, discrimination is implicitly accounted for by using context features that model the similarities between context objects, as described in Section 3.2.2.

4 Speaker model

The EIA and the machine learning models described in Section 3 provide two basic approaches to REG in which, as in most of the existing work in the field, the issue of human variation is not taken into account. In what follows these models are furthered with a range of speaker-dependent information. Both versions – with and without speaker-dependent information – will be evaluated in a REG experiment described in Section 5.

In the case of EIA, human variation was implemented by modifying the way in which the preference order (the P parameter of the algorithm, cf. previous section) is computed.[3] In its original version, the algorithm makes use of a single preference order P for each object type in the domain. The speaker-dependent version, by contrast, will make use of individual preference lists for each speaker. This approach is similar to previous work on speaker-dependent REG (Bohnet 2008; Fabbrizio, Stent and Bangalore 2008).

The machine learning approach described in Section 3.2, by contrast, offers a greater number of opportunities for taking the issue of human variation into account. In its original version, this approach is limited to the use of context features that model domain objects and their similarities. In order to take the issue of human variation into account, this model will be presently expanded with two kinds of feature: those that model personal information about the speakers, and those that model their referential behaviour.

Speaker's personal features are intended to capture coarse-grained differences across speakers. These consist of a unique speaker identifier as in Viethen and Dale (2010) and, as an attempt to a more general model, gender, and age bracket. Personal features of this kind are available from the annotated corpora to be used as training and test data (cf. Section 5.1), except for TUNA, which does not include gender and age information.

Speaker's referential behaviour features model individual choices related to the kinds of attribute that tend to appear in his/her descriptions. We consider two features of this kind, hereby called *TG_Frequencies* and *LM_Frequencies*, representing lists of attribute frequencies of target and landmark objects. Since the TUNA domain does not include relational descriptions, *LM_Frequencies* is not applicable to this case.

The use of individual frequency lists for target and landmark objects is based on the observation that trainable methods as our current approach are highly sensitive

[3] Although other changes are in principle possible, these would greatly change the way in which the Incremental algorithm works, rendering the comparison with the previous work less meaningful.

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press

to the kinds of description available from the data, and that overall frequencies (obtainable by the sum of all target and landmark descriptions) would not account for the full range of possible referential strategies in a given domain. For instance, the Stars2 corpus includes a large number of situations in which the landmark object is particularly small. As a result, references to the landmark size (e.g., 'the ball next to the *small* cone') are common, even though overall corpus frequency estimates would suggest that *size* is less common than *colour*. Similarly, the GRE3D domain includes a large number of relational descriptions involving two objects (e.g., 'the yellow box on top of the red box'), which could be modelled by assigning an overall high preference for relational properties. However, this preference should be applicable only when describing a target object, but not when describing a landmark. Assigning an overall high preference for the spatial relation would lead to overly long descriptions that do not occur in this domain (e.g., 'the yellow box on top of the red box, *next to the ball*').

5 Evaluation

In this section, we describe an experiment to compare the two REG models discussed in Section 3 – one based on the Incremental algorithm and the other based on machine-learning methods – with and without the use of speaker-related information as described in Section 4. The goal of the experiment is to show that (a) using speaker-dependent information in REG produces descriptions that are closer to those produced by human speakers, and that (b) machine-learning methods outperform a standard incremental approach. These objectives are formalised as research hypotheses in the next sections.

5.1 Data

Six datasets are considered for training and testing purposes: the Furniture and People subsets of the TUNA corpus (Gatt *et al.* 2007), hereby called TUNA-f and TUNA-p, GRE3D3 (Dale and Viethen 2009), GRE3D7 (Viethen and Dale 2011), Stars (Paraboni *et al.* 2014), and Stars2 (Paraboni, Galindo and Iacovelli 2016). The complete dataset comprises 7,478 descriptions produced by 590 speakers.

Table 1 shows, for each of the datasets, the number of speakers, descriptions, possible attributes, and objects per scene. These are accompanied by an example of description in each domain. Although most of these datasets (with the exception of TUNA-p) make use of scenes conveying geometric objects, the computational challenges involved in the generation of the corresponding descriptions vary considerably as discussed below.

TUNA-f and TUNA-p contain descriptions of pieces of furniture (e.g., chairs, desks, etc.) and human photographs, respectively. TUNA does not contain relational descriptions, but there are references to screen positions (as in the TUNA-f example). The original TUNA corpus includes a subset of references to sets (e.g., 'the two chairs on the right') as well, but we presently consider only the singular portion of the data. The small number of possible attributes in the Furniture domain makes the

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press

Table 1. *Corpus descriptive statistics and examples*

Corpus	Speakers	Descr.	Attrib.	Obj.	Example
TUNA-f(sing.)	60	420	4	7	Red chair in the middle row.
TUNA-p(sing.)	60	360	10	7	Man with glasses and dark hair.
GRE3D3	63	630	9	3	Red ball.
GRE3D7	287	4,480	6	7	Large ball next to the red cube.
Stars	64	704	8	7–14	Star near the red box, on the left.
Stars2	56	884	9	15	Red ball near a box, near the cone.

generation of these descriptions relatively straightforward. In the People domain, by contrast, a much larger attribute set is considered and, as a result, a certain degree of data sparsity is observed.

The GRE3D3/7 corpora are the largest REG corpora publicly available to date, both in number of speakers and annotated descriptions. Descriptions produced in these domains consider only a small set of possible attributes (colour, size, and one spatial relation) and, in the case of GRE3D3, also screen position. Both corpora are largely focused on the issue of choice between atomic and relational properties. For that reason, in the situations of reference represented in GRE3D3/7 the use of a relational property is never compulsory for disambiguation. For instance, given the goal of describing a box object, it is always possible to choose between an atomic (e.g., 'the large box') and a relational description (e.g., 'the box near the ball').

Stars and Stars2 domains represent situations of reference involving up to three referents in a single description and a larger number of distractor objects in each scene. The situations of reference considered in Stars2 include properties that, although not required for disambiguation, are highly salient. These situations are intended to challenge REG algorithms that make use of a fixed preference order (e.g., always preferring colour over size etc.) Stars includes references to screen positions, but Stars2 does not. The original Stars data includes five filler scenes whose descriptions are unconventionally complex, and which are presently disregarded. In other words, only descriptions produced under the critical conditions of the Stars experiment (Paraboni *et al.* 2014) are presently considered.

5.2 Algorithms

The experiment assessed the Extended Incremental Approach (EIA, cf. Section 3.1) and the classifier models (cf. Section 3.2) in two versions each: with (+) and without (−) speaker-dependent information. Further variations are discussed below.

In the case of the classifier model, two alternative implementations were considered: a standard approach based on decision trees (Breiman *et al.* 1984) as in Viethen and Dale (2010), and a second version using support vector machines with Gaussian kernel (Cortes and Vapnik 1995) as in Ferreira and Paraboni (2014). We will refer to the former as our decision tree (DT) model, and to the latter as our support vector machine (SVM) model.

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press

All classifiers are built from the entire training dataset, that is, we intend to learn how to generate descriptions produced by every individual speaker by making use of knowledge provided by all speakers. Classifiers related to attributes of the target object are built from atomic and relational descriptions alike, and those related to attributes of landmark objects are built from relational descriptions only.

EIA and the two classifier implementations (SVM and DT), all of which with (+) and without (−) speaker-dependent information, will give rise to $3 * 2 = 6$ REG models to be evaluated.

5.3 Hypotheses

Our two main research hypotheses are as follows:

(h1) *REG algorithms that take speaker-related information into account outperform those that do not.*

(h2) *REG algorithms based on machine-learning methods outperform a standard incremental approach.*

Hypotheses h1 and h2 will be tested by comparing every description in the test data with the equivalent produced by one of our algorithms. In doing so, we will follow Gatt, van der Sluis and van Deemter (2007) and others and measure mean Dice coefficients (Dice 1945). Dice coefficients range from 0 to 1 and are intended to represent the degree of overlap between the two attribute sets. In addition to that, we will also compute mean Accuracy scores for illustration purposes. Accuracy also ranges from 0 to 1, and represents the proportion of exact matches between the two sets, that is, the proportion of comparisons in which we obtained a Dice score of 1.

For h1, we will compare algorithms that do not take (−) speaker-related information with those that do (+). More specifically, we will compare EIA− with EIA+, SVM− with SVM+, and DT− with DT+. In all cases, we expect the version that uses (+) speaker-related information to outperform the version that does not (−).

For h2, we will compare the EIA with both machine learning models (SVM and DT). More specifically, we will compare EIA− with SVM−, EIA+ with SVM+, EIA− with DT− and EIA+ with DT+. In all cases, we expect SVM/DT to outperform EIA.

Regarding the possible difference between DT and SVM, no predictions will be made. DT and SVM are intended simply as alternative implementations of our machine learning model, and we assume that their results will vary across domains.

5.4 Procedure

All algorithms used six-fold cross validation to generate Stars and TUNA descriptions, and used ten-fold cross-validation to generate GRE3D3, GRE3D7, and Stars2 descriptions. The number of folds for each domain was defined so as to balance the number of speakers within each fold.

In the case of EIA, one fold was left out as test data, and the others were used to estimate attribute frequencies, and also to build the preference lists used in each iteration of the cross-validation.

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press

Table 2. *Results (top Dice scores for each domain are presented in boldface)*

	TUNA-f		TUNA-p		GRE3D3		GRE3D7		Stars		Stars2		Overall	
Model	Dice	Acc.	Dice	Acc.	Dice	Acc.	Dice	Acc.	Dice	Acc.	Dice	Acc.	Dice	Acc.
EIA−	0.72	0.23	0.61	0.09	0.69	0.24	0.88	0.61	0.53	0.00	0.63	0.26	0.79	0.45
EIA+	0.79	0.36	0.72	0.24	0.68	0.23	0.88	0.61	0.65	0.11	0.66	0.26	0.81	0.47
SVM−	0.75	0.28	0.61	0.19	0.78	0.46	0.88	0.61	0.59	0.13	0.58	0.29	0.80	0.49
SVM+	**0.84**	0.41	0.78	0.36	**0.93**	0.74	**0.94**	0.77	0.71	0.31	0.73	0.42	**0.89**	0.65
DT−	0.75	0.28	0.61	0.20	0.77	0.40	0.89	0.63	0.70	0.21	0.58	0.29	0.81	0.51
DT+	0.83	0.40	**0.82**	0.37	0.88	0.61	0.93	0.72	**0.84**	0.53	**0.75**	0.44	**0.89**	0.63

Table 3. *REG without (−) and with (+) speaker-related information (hypothesis h1)*

(−) Speaker info			(+) Speaker info			h1	
Model	Dice	Acc.	Model	Dice	Acc.	W	p
EIA−	0.79	0.45	EIA+	0.81	0.47	130944.5	< 0.0001
SVM−	0.80	0.49	SVM+	0.89	0.65	780908.0	< 0.0001
DT−	0.81	0.51	DT+	0.89	0.63	1292288.0	< 0.0001

For the SVM classifiers, optimal values for the cost (C) and the Gaussian Kernel's gamma (γ) parameters were determined by performing Grid Search as follows. For each iteration of the cross-validation task, one-fold f was reserved for testing, and the remaining folds were used in a secondary cross-validation task in which the classifiers were trained using different combinations of C (1, 10, 100, and 1000) and γ (1, 0.1, 0.001, and 0.0001) values. The parameters that produced highest accuracy were thus selected, and then applied to the primary cross-validation task to test fold f. This procedure was repeated for each iteration of the primary cross-validation.

6 Results

Table 2 presents mean Dice and Accuracy scores for each of the algorithms under evaluation, and according to each test corpus. In all comparisons, we use the Wilcoxon's signed rank test over mean Dice scores.

Table 3 summarises the overall comparisons between the models that do not (−) take speaker information into account and those that do (+), as required for hypothesis h1.

Results in Table 3 show that algorithms that take (+) speaker-related information into account generally outperform those that do not (−). The differences are significant as noted. This supports hypothesis h1.

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press

Table 4. *Incremental and Machine Learning REG (hypothesis h2)*

Incremental			Machine Learning			h2	
Model	Dice	Acc.	Model	Dice	Acc.	*W*	*p*
EIA−	0.79	0.45	SVM−	0.80	0.49	5186199.5	< 0.05
EIA+	0.81	0.47	SVM+	0.89	0.65	2644193.5	< 0.0001
EIA−	0.79	0.45	DT−	0.81	0.51	2211870.0	< 0.0001
EIA+	0.81	0.47	DT+	0.89	0.63	2169395.5	< 0.0001

Table 4 summarises the overall comparisons between the models that are based on the incremental approach (i.e., EIA−/+) and those that are based on machine-learning methods (i.e., SVM−/+ and DT−/+), as required for hypothesis h2.

Results in Table 4 show that algorithms based on machine-learning methods (SVM and DT) generally outperform the EIA in both versions (i.e., regardless of using speaker related information or not). The differences are significant as noted. This supports h2.

6.1 Discussion

Results of our REG models based on six test datasets generally support both our research hypotheses: the use of speaker-related information does produce descriptions that are closer to those produced by human speakers (as measured by Dice coefficients) and, perhaps to a lesser extent, so does the use of machine learning. From these general findings, the following observations are warranted.

First, we notice that the overall best-performing strategy (taking both Dice and Accuracy measures into account) was SVM+, that is, the model based on support vector machines, and which makes use of speaker-related information. This generally confirms previous studies on human variation in REG (Viethen and Dale 2010) and on the use of machine-learning methods in this task (Viethen 2011). SVM+ is closely followed by DT+, which turned out to be even better for the generation of Stars and Stars2 descriptions.

Regarding differences across domains, mean Dice scores in Table 2 suggest that the REG task for GRE3D3 and GRE3D7 was considerably easier than for the other corpora. This may be explained by the relative simplicity of these domains, which generally involve a smaller number of possible referential attributes. This simplicity seems to be fully accounted for by the machine learning models, but less so by the incremental approach.

Regarding the use of speaker-related information, Table 3 shows that all models benefit from this knowledge source. Once again, however, the improvement is much greater in the case of the machine learning models. This difference was to be expected since the underlying speaker models for SVM and DT are richer than the speaker model in EIA. In fact, SVM and DT combine the contributions of the

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press

Table 5. *Comparison with alternative speaker models*

	SVM		DT	
Speaker information	Dice	Acc.	Dice	Acc.
None (SVM− and DT−)	0.80	0.49	0.81	0.51
Age+gender only	0.82	0.53	0.82	0.53
Identifier only	0.83	0.56	0.83	0.52
Preferred list only	0.88	0.64	0.88	0.60
All (SVM+ and DT+)	0.89	0.65	0.89	0.63

machine-learning methods and the use of speaker-related information, whereas EIA – which was solely intended as a baseline system – was kept as close as possible to the original Incremental algorithm. For similar reasons, we notice in Table 4 that the impact of a machine learning model is greater when taking speaker-related information into account.

Finally, in addition to the baseline models without speaker-related information (SVM− and DT−) and those that use the full set of speaker-related features (SVM+ and DT+), we well briefly consider three intermediate solutions: a model that considers only the use of speaker's identifiers, a model that uses age and gender information, and a model that uses the individual's preferred attribute lists. Results for our SVM and DT methods and these alternatives are summarised in Table 5.

We compared the Dice scores obtained by the more complete models SVM+ and DT+ with the second best alternatives (in both cases, the use of the preferred attribute list for each speaker). A Wilcoxon's signed rank test shows that the differences are significant both in case of SVM+ (W=496,289, p<0.0001) and DT+(W=657,255, p<0.0001). Thus, the use of the entire set of speaker-dependent features outperforms the alternatives under consideration.

Regarding the individual contributions of each alternative, we notice that, as in previous work in the field (cf. Section 2.2), the use of individual preference lists has the greatest overall impact, although the use of other speaker-related attributes also improves results to a lesser extent.

One possible explanation for the relatively small role of personal attributes is their distribution in the four test datasets that include gender and number information. In particular, we notice that both Stars and Stars2 show a strong bias towards male participants (Eighty-four per cent in Stars and ninty-one per cent in Stars2) with very little variation in age (a standard deviation of only 3.2 years in Stars, and 2.9 years in Stars2). Gender and age distributions are less problematic in the GRE3D domain, although we still observe that in the GRE3D3 corpus fifty-one per cent of the speakers are in the 20–25 age bracket, and further twenty-nine per cent are aged 26–30. Based on these observations, we assume that the use of personal features (as in the more complete models SVM+ and DT+) is indeed helpful, and their impact may become more evident when these features are sufficiently well-balanced in the training data.

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press

Table 6. *Best-performing previous work versus SVM+ results for each test corpus*

		Previous		SVM+	
Test data	Algorithm	Dice	Acc.	Dice	Acc.
TUNA-f (2007)	IS-FBN	0.80	na	0.84	0.44
TUNA-p (2007)	IS-FBN	0.74	na	0.75	0.31
GRE3D3	Longest First	0.79	0.45	0.93	0.74
GRE3D7	Viethen and Dale (2010)	na	0.67	0.94	0.77
Stars	Paraboni *et al.* (2014)	0.61	0.11	0.71	0.31
Stars2	Paraboni *et al.* (2016)	0.73	0.36	0.73	0.42

6.2 Comparison with previous work

In addition to providing support to our research hypotheses, results of our speaker-dependent models are often superior to those obtained by previous work over TUNA, GRE3D3/7, and Stars/2 data. To illustrate this, Table 6 summarises the best results obtained by previous work and our overall best-performing SVM+ model.[4] Further details are discussed below.

The comparison involving TUNA data is complicated by the fact that different studies make use of different portions of the corpus as test data, which may be a consequence of its gradual release along the first two TUNA shared tasks (Gatt and Belz 2007; Gatt *et al.* 2008) before changes in the evaluation methodology were introduced (Gatt *et al.* 2009), and Dice scores were no longer reported. We are unaware of any REG algorithm that has been evaluated over the entire set of TUNA singular descriptions as presented in the previous section, and we notice that the reported results tend to vary widely depending on the portion of test data under consideration.

Bearing these difficulties in mind, our analysis is based on the test dataset provided for the 2007 Challenge. Results in Table 6 show that SVM+ outperforms the 2007 winner IS-FBN in both Furniture and People domains. As for the TUNA 2008 test data (Gatt *et al.* 2008), we notice that the test data provided for the shared task contained referring expressions produced by previously unseen speakers, which in principle defeats speaker-dependent methods such as the present work.

In the GRE3D domain, SVM+ obtains its best results, outperforming both the *Longest First* strategy in Viethen, Mitchell and Krahmer (2013) using the GRE3D3 data, and the decision trees induction method in Viethen and Dale (2010) using the GRE3D7 data.

Finally, regarding the Stars corpus, SVM+ outperform the use of Naive Bayes classifiers to decide whether to produce under or overspecified descriptions in Paraboni *et al.* (2014). In the case of Stars2, SVM+ also outperforms the use of an overspecification strategy based on the preference for highly discriminatory properties in Paraboni, Galindo and Iacovelli (2016).

[4] Although in some cases results for DT+ may be even higher, as discussed in the previous section.

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press

7 Final remarks

This article has discussed the use of speaker-dependent information in a machine learning approach to the generation of natural language referring expressions. The proposed model refines previous accounts of human variation in REG, and results are generally superior, or at least comparable to, existing work in the field. Moreover, our study has suggested that the use of machine learning may not only improve overall results, but it may be actually necessary for a comprehensive account of human variation in Natural Language Generation tasks of this kind.

The present study leaves many open research questions and opportunities for future work. First, we notice that we may of course consider expanding the current model with even further speaker-related features. In particular, we may envisage the use of features to represent a more detailed account of how different speakers decide to under or overspecify individual portions of a complex description. For instance, it may be useful to model features that describe the specification level of the main target and a related landmark individually, an issue that has a number of known consequences for identification (Paraboni and van Deemter 2013). However, we are aware that modelling referential behaviour at this level of detail may require more data than what is currently available from existing corpora of referring expressions.

Second, we notice that descriptions produced by the present work were tailor-made for each individual speaker. Whereas, this may be in principle adequate for certain purposes, at a more general level we may consider the use of speaker profiles representing the referential behaviour of a group of individuals. Profiles of this kind may in principle be built, for instance, with the aid of non-supervised machine-learning methods.

Finally, we notice that the present model depends entirely on training data provided by existing REG corpora. Considering however that resources of this kind are rare and generally costly, a legitimate question to ask is how to model human variation when large training datasets are not available. Although findings in Koolen, Krahmer and Theune (2012) suggest that training on a small number of descriptions may already be sufficient for good performance in certain domains, an investigation of this kind is also left as future work.

References

Anderson, A., Bader, M., Bard, E., Boyle, E., Doherty, G. M., Garrod, S., Isard, S., Kowtko, J., McAllister, J., Miller, J., Sotillo, C., Thompson, H. S., and Weinert, R. 1991. The HCRC map task corpus. *Language and Speech* **34**: 351–66.

Arts, A., Maes, A., Noordman, L. G. M., and Jansen, C. 2011. Overspecification facilitates object identification. *Journal of Pragmatics* **43**: 361–74.

Bohnet, B. 2007. IS-FBN, IS-FBS, IS-IAC: The adaptation of two classic algorithms for the generation of referring expressions in order to produce expressions like humans do. In *Proceedings of the Language Generation and Machine Translation UCNLG+MT Workshop at MT Summit XI*. Copenhagen, pp. 84–6.

Bohnet, B. 2008. The fingerprint of human referring expressions and their surface realization with graph transducers. In *Proceedings of the 5th International Natural Language Generation Conference (INLG-2008)*. Association for Computational Linguistics, Salt Fork, USA, pp. 207–10.

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. 1984. *Classification and Regression Trees*. Statistics/Probability Series. Belmont, USA: Wadsworth Publishing Company.

Cortes, C. and Vapnik, V. 1995. Support-vector networks. *Machine Learning* **20**(3): 273–97.

Dale, R. 1989. Cooking up referring expressions. In *Proceedings of the 27th Annual Meeting on Association for Computational Linguistics* (ACL '89). Association for Computational Linguistics, Vancouver, Canada, pp. 68–75.

Dale, R. and Haddock, N. J. 1991. Content determination in the generation of referring expressions. *Computational Intelligence* **7**(4): 252–65.

Dale, R. and Reiter, E. 1995. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science* **19**(2): 233–63.

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

Dice, L. R. 1945. Measures of the amount of ecologic association between species. *Ecology* **26**(3): 297–302.

dos Santos Silva, D., and Paraboni, I. 2015. Generating spatial referring expressions in interactive 3D worlds. *Spatial Cognition & Computation* **15**(03): 186–225.

Eugenio, B. D., Jordan, P. W., Thomason, R. H., and Moore, J. D. 2000. The agreement process: an empirical investigation of human-human computer-mediated collaborative dialogues. *International Journal of Human-Computer Studies* **53**(6): 1017–76.

Fabbrizio, G. D., Stent, A. J., and Bangalore, S. 2008. Trainable speaker-based referring expression generation. In *Proceedings of the 12th Conference on Computational Natural Language Learning* (CoNLL '08). Association for Computational Linguistics, Manchester, UK, pp. 151–8.

Ferreira, T. C., and Paraboni, I. 2014a. Classification-based referring expression generation. In A. Gelbukh (ed.), *Computational Linguistics and Intelligent Text Processing: 15th International Conference, CICLing 2014*, vol. 8403, pp. 481–91. Lecture Notes in Artificial Intelligence. Berlin: Springer Verlag.

Ferreira, T. C., and Paraboni, I. 2014b. Referring expression generation: taking speakers' preferences into account. In P. Sojka, A. Horák, I. Kopeček, and K. Pala (eds.), *Text, Speech and Dialogue: 17th International Conference, TSD 2014*, vol. 8655, pp. 539–46. Lecture Notes in Artificial Intelligence. Berlin: Springer Verlag.

Gatt, A. and Belz, A. 2007. The attribute selection for GRE challenge: overview and evaluation results. In *Proceedings of the Language Generation and Machine Translation UCNLG+MT Workshop at MT Summit XI*. Copenhagen, pp. 75–83.

Gatt, A., Belz, A., and Kow, E. 2008. The TUNA challenge 2008: overview and evaluation results. In *Proceedings of the 5th International Natural Language Generation Conference (INLG-2008)*. Association for Computational Linguistics, Salt Fork, USA , pp. 198–206.

Gatt, A., Belz, A., and Kow, E. 2009. The TUNA challenge 2009: overview and evaluation results. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG-2009)*. Association for Computational Linguistics, Athens, Greece , pp. 174–82.

Gatt, A., Krahmer, E., van Gompel, R., and van Deemter, K. 2013. Production of referring expressions: preference trumps discrimination. In *The Annual Meeting of the Cognitive Science Society* (CogSci-2013). Berlin, Germany, pp. 483–8.

Gatt, A., van der Sluis, I., and van Deemter, K. 2007. Evaluating algorithms for the generation of referring expressions using a balanced corpus. In *Proceedings of the 11th European Workshop on Natural Language Generation* (ENLG-2007). Association for Computational Linguistics, Schloss Dagstuhl, Germany, pp. 49–56.

Grice, H. P. 1975. Logic and conversation. In P. Cole and J. L. Morgan (eds.), *Syntax and Semantics: Vol. 3: Speech Acts*, pp. 41–58. San Diego, CA: Academic Press.

Gupta, S. and Stent, A. J. 2005. Automatic evaluation of referring expression generation using corpora. In *Proceedings of the Corpus Linguistics 1st Workshop on Using Corpora in Natural Language Generation* (UCNLG). Birmingham, UK, pp. 1–6.

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press

Hervás, R., Arroyo, J., Francisco, V., Peinado, F., and Gervás, P. 2016. Influence of personal choices on lexical variability in referring expressions. *Natural Language Engineering* **22**(2): 257–90, DOI 10.1017/S1351324915000182.

Hervás, R., Francisco, V., and Gervás, P. 2013. Assessing the influence of personal preferences on the choice of vocabulary for natural language generation. *Information Processing & Management* **49**(4): 817–32.

Koolen, R., Krahmer, E., and Theune, M. 2012. Learning preferences for referring expression generation: effects of domain, language and algorithm. In *Proceedings of the 7th International Natural Language Generation Conference* (INLG-2012). Association for Computational Linguistics, Utica, USA, pp. 3–11.

Krahmer, E. and van Deemter, K. 2012. Computational generation of referring expressions: a survey. *Computational Linguistics* **38**(1): 173–218.

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

Paladhi, S. and Bandyopadhyay, S. 2008. JU-PTBSGRE: GRE using prefix tree based structure. In *Proceedings of the 5th International Natural Language Generation Conference (INLG-2008)*. Association for Computational Linguistics, Salt Fork, USA, pp. 230–1.

Paraboni, I., Galindo, M. R., and Iacovelli, D. 2016. Stars2: a corpus of object descriptions in a visual domain. In *Language Resources and Evaluation*, DOI 10.1007/s10579-016-9350-y.

Paraboni, I. and van Deemter, K. 2013. Reference and the facilitation of search in spatial domains. *Language, Cognition and Neuroscience*, DOI 10.1080/01690965.2013.805796, pp. 1002–17.

Paraboni, I., Yamasaki, A. K., da Silva, A. S. R., and Teixeira, C. V. M. 2014. Generating underspecified descriptions of landmark objects. In P. Sojka, A. Horák, I. Kopeček, and K. Pala (eds.), *Text, Speech and Dialogue: 17th International Conference, TSD 2014*, vol. 8655, pp. 76–83. Lecture Notes in Artificial Intelligence. Berlin: Springer Verlag.

Siddharthan, A. and Copestake, A. 2004. Generating referring expressions in open domains. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics* (ACL). Association for Computational Linguistics, Barcelona, Spain, pp. 407–14.

van Deemter, K. 2002. Generating referring expressions: boolean extensions of the incremental algorithm. *Computational Linguistics* **28**(1): 37–52.

Viethen, H. A. E. 2011. *The Generation of Natural Descriptions: Corpus-based Investigations of Referring Expressions in Visual Domains*. PhD Thesis, Macquarie University, Sydney, Australia.

Viethen, J. and Dale, R. 2010. Speaker-dependent variation in content selection for referring expression generation. In *Proceedings of the Australasian Language Technology Association Workshop 2010*. Melbourne, Australia, pp. 81–9.

Viethen, J. and Dale, R. 2011. GRE3D7: a corpus of distinguishing descriptions for objects in visual scenes. In *Proceedings of the UCNLG+Eval: Language Generation and Evaluation Workshop*. Association for Computational Linguistics, Edinburgh, Scotland , pp. 12–22.

Viethen, J., Mitchell, M., and Krahmer, E. 2013. Graphs and spatial relations in the generation of referring expressions. In *Proceedings of the 14th European Workshop on Natural Language Generation*. Association for Computational Linguistics, Sofia, Bulgaria, pp. 72–81.

https://doi.org/10.1017/S1351324917000079 Published online by Cambridge University Press