Journal of Artificial Intelligence Research 77 (2023) 167-206 Submitted 05/2022; published 05/2023

Computational Modelling of Quantifier Use: Corpus, Models, and Evaluation

Guanyi Chen G.CHEN@UU.NL
Kees van Deemter C.J.VANDEEMTER@UU.NL
Department of Information and Computing Sciences
Utrecht University
Utrecht, The Netherlands

Abstract

A prominent strand of work in formal semantics investigates the ways in which human languages quantify the elements of a set, as when we say *All A are B*, *Few A are B*, and so on. Building on a growing body of empirical studies that shed light on the meaning and the use of quantifiers, we extend this line of work by computationally modelling how human speakers textually describe complex scenes in which quantitative relations play an important role. To this end, we conduct a series of elicitation experiments in which human speakers were asked to perform a linguistic task that invites the use of quantified expressions. The experiments result in a corpus, called QTUNA, made up of short texts that contain a large variety of quantified expressions. We analyse QTUNA, summarise our findings, and explain how we design computational models of human quantifier use accordingly. Finally, we evaluate these models in accordance with QTUNA.

1. Introduction

The aim of this paper is to propose and evaluate computational models of human speakers' use of quantification in the description of visual scenes.

Quantified noun phrases are studied in different research traditions. Much work has been done by formal semanticists, often building on the idea that the prime function of a noun phrase is to express quantitative relations between sets of individuals. The study of Generalised Quantifiers, as it is often called, can be understood as an attempt to understand the huge variation in quantifier patterns: not only we can say things of the form "*All A are B*" and "*All except two A are B*", but also "*Most A are B*" and "*Few A are B*", which are not expressible in First-Order Predicate logic. Quantifiers can also play other logical roles, for instance when we say "*There are (some/few/etc.) A*", where the quantifier has only one set argument (namely A). Clearly, a speaker who describes a situation by using quantified noun phrases faces a large range of options, many of which express different propositions.

Human use of quantifiers has been studied intensively since Barwise and Cooper (1981); an overview of work in this "logical" tradition can be found in Peters and Westerståhl (2006). A more empirical strand of work asks how human speakers produce and comprehend quantified Noun Phrases, focusing on properties of one particular quantifier (Lidz, Pietroski, Halberda, & Hunter, 2011; Kotek, Sudo, & Hackl, 2015), or differences between small sets (e.g., pairs) of quantifiers (Moxey & Sanford, 1993; Geurts & Nouwen, 2007; Zajenkowski & Szymanik, 2013; Solt, 2016; Lappin, 2000), focusing on quantifiers in a fixed sentence position (e.g., the position Q in the sentence "Q of the circles are round").

In recent years, many areas of human behaviour have been "simulated" using computer programs, including human memory, logical reasoning, and so on (see e.g., Sun (2008)), resulting in a methodological paradigm sometimes referred to as computational modelling. This paradigm has been extended to human language production as well (e.g., van Deemter (2016)). In the spirit of this line of work, we want to construct a computational model of human quantifier use. Unlike *process models*, which characterise the *manner* in which humans perform a given task, our models merely characterise the input-output behaviour between scenes perceived and descriptions uttered. Models of this kind are known as *product models* (Sun, 2008). Product models often focus on predicting how a human speaker would verbally describe a given visual scene without claiming that the steps that our algorithms take resemble processing steps undertaken in the human mind; in other cases, they focus on producing outputs that are optimal for hearers or readers.[1] The models presented in this paper will be evaluated both in terms of the extent to which the descriptions they produce are perceived to resemble human-produced descriptions, and, especially, in terms of their utility for human readers.

We consider our models to be a valuable addition to those computational models that focus on *interpreting* natural language because the former embodies an insight into *what utterance is most appropriate in a given situation*: thus, the model embodies an understanding of expressive *choice*. In a nutshell, *why do we say what we say?*, addressing both the strategic aspect of this question (i.e., *What information do we express?*) and the tactical aspect (i.e., *How do we express it?*). The expressive choice is the defining challenge of the research field of Natural Language Generation (e.g. Gatt and Krahmer (2018)).

Given that modelling the full range of speakers' use of quantifiers is an extremely ambitious goal, we focus on simple situations, where there is only a limited range of objects to talk about, and a limited range of things to say about them, embedded in a simple communicative setting that minimises the role of such "complicating" factors as background knowledge and expectations that the speakers or hearers may have about the domain. To build a good model of language use, one needs to know:

1. What utterances, including what quantified expressions, are likely to be uttered by a speaker in a given situation?

2. If a given quantified expression is uttered, what information does it convey?

Aspects of these questions have been addressed before. For instance, Yildirim, Degen, Tanenhaus, and Jaeger (2013) investigated speakers' use and hearers' interpretation of the quantifiers "some" or "many". Herbelot and Vecchi (2015) looked at "no", "all", "most", "some", and "few". Sorodoc, Lazaridou, Boleda, Herbelot, Pezzelle, and Bernardi (2016) focused on "no", "some", and "all".

Building on evidence that hearers interpret quantifiers probabilistically (Yildirim et al., 2013; Degen & Tanenhaus, 2011; van Tiel, 2014), works such as Franke (2014) and Qing (2014) built probabilistic speaker models for these two quantifiers, i.e., *some* and *many*, based on Bayesian pragmatics (Frank & Goodman, 2012). All these works focus on very limited sets of quantifiers, and on a given sentence pattern, where the task focused on the meaning and use of a quantifier in a given position in the sentence.

1. For further discussion of these perspectives, see van Deemter (2016), particularly Chapter 16.1.

To the best of our knowledge, there have been no attempts to model computationally how a wider range of quantifiers are used by human speakers, let alone in a setting that allows unlimited choice of sentence patterns (instead of having to choose from a list of options). [2]

To get a glimpse of the challenge, consider a table with four coffee cups, three of which are red and one is white. Each of the following expressions describes this scene truthfully:

(1) a. There are some red cups on the table.
b. At least three cups are red.
c. Fewer than four cups are red.
d. All the red objects are coffee cups.
e. Three of the four cups are red.

Each of these sentences could be uttered felicitously in some contexts. For example, (1-a) might make a fine answer to the question, *Is the table empty now?* However, as a description of the scene as a whole (e.g., answering the question, *Can you tell me what's on the table?*), (1-e) would probably be more effective. An early computational investigation of the question of *what quantifiers are called for in a given situation* proposed the principle of informativity (Creaney, 1996). This principle asserted that the speaker should always choose the logically *strongest* expression that holds true in a given situation. Although the idea of looking at the logical strength of an expression makes sense[3], Creaney's idea runs into difficulties over pairs of expressions that are logically independent of each other, such as the pair of (1-b) and (1-c), where each of the two expressions conveys some information that the other one does not. Examining the evidence, we suspect that no single "principle" can tell us what makes the best description of a visual scene and that a radically different, more empirically guided approach is called for, to inform the generation algorithm. The present paper offers such an approach.

To obtain more insight into these issues, we decided to study situations in which the sentence patterns are not given in advance, and where speakers are free to describe a visual scene in whatever way they want, using as many sentences as the speaker chooses, and using any sentence pattern that they choose. The present setup also has the advantage of leaving the decision of whether or not to use a quantifier to the speaker herself/himself. Last but not least, it permits the use of quantifiers of all possible logical types. The resulting setup has the advantage of allowing participants to use language in a more natural way than in earlier experiments: just as speakers do in daily life, they utter sequences of self-constructed sentences; this kind of set-up is thought to be more suitable for investigating real language use than when speakers are given more artificial tasks.

For this purpose, we conducted a series of *elicitation* experiments, in which each participant was asked to produce descriptions of visual scenes. For example, for the scene presented in Figure 1, a participant in our experiment might say "*Half of the objects are blue squares,*

2. Barr, van Deemter, and Fernández (2013) elicited noun phrase patterns of the form "*the square with Q dots/dashes/etc*"; though this gave the authors a range of different quantifiers, the sentence pattern was once again fixed; moreover, the paper does not attempt a computational model. More recently, Pezzelle, Steinert-Threlkeld, Bernardi, and Szymanik (2018) formalised quantifier selection task based on a cloze test, asking models to predict which quantifier is used in a given context.
3. See the Greedy Algorithm of our section 3.3, which makes use of a similar idea.

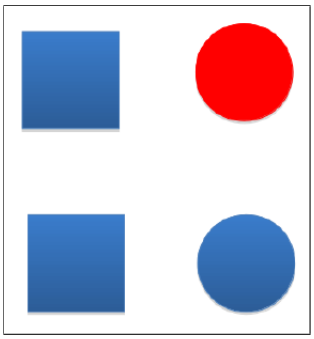


Figure 1: An example scene in QTUNA with 4 objects. Other scenes contain 9 and 20 objects.

the other half are circles in both colours." For want of a better name, we call such a stretch of text a *Quantified Description*. This elicitation experiment yielded a data-text corpus that we call QTUNA. We believe that this corpus will be a source of inspiration for researchers in various research areas, including students of Generalised Quantifiers (in the intersection of Linguistics and Logic) and psychologists interested in human language production. The present paper concentrates on a different use of this data-text corpus: Based on an analysis of this corpus we designed two rule-based "Quantified Description Generation" algorithms, which mimic the types of quantified descriptions that human speakers use in any given situation; a rule-based approach was chosen because it allows us to link with the theoretical literature on quantification, and with computational models of other linguistic phenomena. We then evaluated our algorithms and found that these work rather well, both in terms of describing scenes in the QTUNA corpus and in terms of describing scenes of different sizes (i.e., domain sizes not occurring in the corpus).

The plan of the paper is as follows. Section 2 introduces the QTUNA experiment and offers an analysis of the corpus. Section 3 motivates and describes our algorithms. Section 4 offers evaluations of its output, based on both expert judgements and scene reconstructions. Section 5 puts our results in context and discusses their merits and limitations. [4]

2. Building and Learning from a Corpus of Quantified Descriptions

Computational modelling of language production usually starts from building corpora of expressions elicited from human participants. A representative line of work focuses on corpora of referring expressions, such as GRE3D (Viethen & Dale, 2008) and TUNA (Gatt, van der Sluis, & van Deemter, 2007; van Deemter, Gatt, Sluis, & Power, 2012). These corpora were used for evaluating the "humanlikenesss" of the expressions produced by computational models (i.e., the degree to which the latter resemble human-produced expressions). In our study of quantification, we broadly follow the methodology developed in the TUNA project, which has been widely adopted, for example as the basis for a series of Shared

4. The QTUNA dataset and the corresponding materials are available at: `https://github.com/a-quei/qtuna`. The code for our quantified description generation algorithms is available at: `https://github.com/a-quei/quantified-description-generation`.

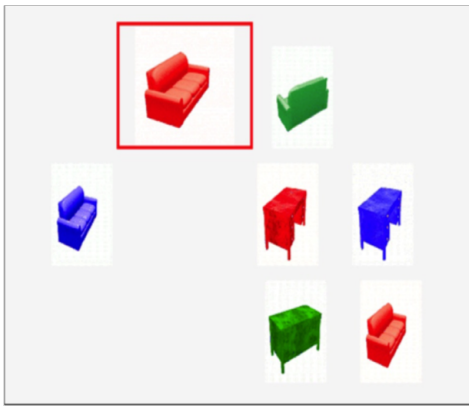


Figure 2: An example scene from the TUNA experiment, where the subjects were asked to describe the object in the red window given the other objects as context.

Task Evaluation Campaigns (Belz & Gatt, 2007). The resulting corpus was also extended to other languages, including Dutch (Koolen, Gatt, Goudbeek, & Krahmer, 2011), German (Howcroft, Vogels, & Demberg, 2017) and Mandarin (van Deemter, Sun, Sybesma, Li, Chen, & Yang, 2017). In TUNA, each subject was given a scene like the one in Figure 2, and asked to produce a description (i.e., referring expression) that singles out the object in the red window from all other objects in the scene. We decided to use a similar methodology but adapt the method to the study of quantification.

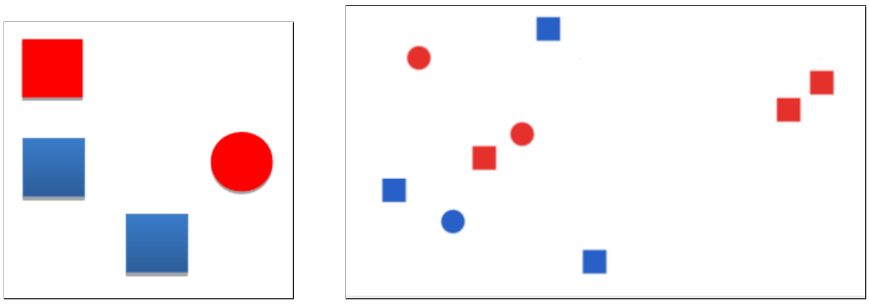
In order to understand how people use quantification, we conducted the QTUNA experiment, which led to the QTUNA[5] corpus. In this section, we explain how the experiment was set up and how the resulting corpus was analysed. An initial introduction and analysis of this experiment can be found in Chen, van Deemter, Pagliaro, Smalbil, and Lin (2019b).

2.1 Eliciting Quantified Descriptions

As discussed in the previous section, we wanted to find out how a broad range of quantified NPs is used as part of a wider communicative task. Instead of showing our subjects a scene and asking them how they would explain to a hearer how many circles are red (e.g., "*many.*"), we asked them to describe *the scene as a whole*. We made the scenes complex enough that one simple quantified expression (QE) would never suffice. Scenes came in different sizes; we use the variable N to represent the size, i.e., the number of objects in a given scene.

Each participant was presented with a series of abstract visual scenes of a certain size (measured by the number of objects contained in it). Instead of using realistic photographs, we decided to use synthetic visual scenes because this makes it easy to construct and modify the scenes where necessary (see Pezzelle and Fernández (2019) and Testoni, Pezzelle, and Bernardi (2019)). Each scene contains N objects, each of which is either a circle or a square in either blue or red. Our instructions to participants (see Figure 4) asked participants to

5. The name of QTUNA is a variant of TUNA, where Q stands for quantification.



(a) (b)

Figure 3: Examples from (a) the $N = 4$ experiment; (b) the $N = 9$ experiment.

try to produce a quantified description that would allow a reader to *reconstruct* the scene modulo location (i.e., to reconstruct the scene except for the location of each object), thus ensuring a focus on quantitative information only. Pilot experiments had taught us that without the "modulo" clause, many participants would focus on location to such an extent that it led to a large reduction in the number of different quantifiers used (e.g. producing descriptions that say "*From left to right, we see ...*").

2.1.1 Domain Size

To find out how domain size impacts the use of quantifiers, we conducted three different elicitation experiments, with domain sizes (N) of 4, 9, and 20 respectively, each containing 10 different scenes, yielding three sub-corpora. Figure 3(a) and Figure 3(b) show two example scenes from the $N = 4$ and the $N = 9$ experiment respectively.

2.1.2 Experiment Design

Designing a workable set of instructions for participants proved to be a challenging task, so we decided to start with a series of pilot experiments before conducting the real experiment. Apart from the requirement of avoiding participants from mentioning the location of each object, we also needed to discourage them from performing what we called *enumeration* as different kinds of objects in the scene are listed one by one. This had happened frequently in some of our pilot experiments, causing only a small range of (mostly existential) quantifiers to be used. For example, a scene like Figure 3(a) was described as follows:

(2) There are two blue squares, one red square, and one red circle.

Although these descriptions are perfectly legitimate, they do not contain a wide range of quantifier patterns. To ensure that descriptions fulfilled a concrete purpose, we also wanted to encourage descriptions that are logically "complete", by which we mean that participants should do their best to produce a description that allows readers to *reconstruct* the situation in all respects except the location of the objects.

In an early pilot experiment, we tried to encode the above requirements explicitly in the instructions, saying things like, *do not use numerals when describing the situation* and *do not describe the location of objects*. However, this did not work well, because many

We'd like you to describe each situation in one or more grammatically correct English sentences. (...)

1 *Based on your description, a reader will try to "reconstruct" the situation. We use the word "reconstruct" loosely here because the only thing that matters is the different types of objects that the sheet contains. Therefore, please do not say *where* in the grid a particular object is located (e.g., "top left", "in the middle", "on the diagonal").*

2 *Each object is a circle or a square, and either red or blue. Your reader knows this.*

3 *Please do not "enumerate" the different types of objects. For example, do not say "There is a red circle, two blue circles, and ...".*

4 *Every situation contains four objects. Your reader knows this in advance, and he/she will take this information into account when interpreting your description.*

Figure 4: The sketch of how an instruction looks like, taking $N = 4$ as an example.

subjects still used enumerations and locations. After a number of pilots, we decided to omit these explicit rules. Instead, we asked subjects to avoid enumeration as much as possible and added two examples in the instructions, explaining how one of them would allow a reader to reconstruct the situation whereas the other did not. [6] Figure 4 depicts what the instruction for the $N = 4$ experiment looks like. The avoidance of enumeration may have diminished the ecological validity (Schmuckler, 2001) of our findings somewhat, but we believe that this is more than outweighed by the increased richness of the resulting descriptions.

Each object has two attributes: shape and colour. Both of these two attributes have two different values, so there were 4 possible combinations of attributes, i.e., blue square, blue circle, red square and red circle. Since there were at least 4 objects (in $N = 4$ experiment) in each scene, the number of attribute combinations can vary from one (i.e., all the objects are the same) to four. In our experiment, we ensured that all these variations are presented (i.e., there were scenes with 4, 3, 2, and 1 number of attribute combinations). In addition, we took care to balance shape and colour. For example, in the $N = 4$ experiment, from the set of scenes where there are 2 combinations, we selected one in which the two combinations differ in terms of colour (2 red squares and 2 blue squares) and one in which they differ in shape (2 red circles and 2 red squares).

Furthermore, instead of placing the objects in a grid (as was done in our earliest pilots), we ended up placing objects in a more random layout as in Figure 3(a) and Figure 3(b). The changes that we made on the basis of our pilots proved to be very effective for letting speakers produce descriptions that meet the requirements spelt out above, leading to a richly varied set of quantified descriptions.

6. For instance, in the $N = 4$ experiment, the two examples are:
(i) There are equally many circles as squares. All squares are blue. Half the circles are blue.
(ii) Half of the objects are blue squares.

2.2 The QTUNA Corpus

Our experiments yielded three sub-corpora corresponding to the three scene sizes. In this section, we introduce the corpus and its annotation.

2.2.1 Basic Statistics

Participants in our elicitation experiments were students at the computing science department of Utrecht University. Descriptions from 66, 63, and 58 participants respectively were collected for the three experiments. We manually filtered out all descriptions from subjects who showed a misunderstanding of the task: 1) writing gibberish; 2) describing the scene by enumerating the objects in it; or 3) describing the scene by expressing locations (e.g., "*.. at the bottom right corner of the screen*"). The resulting corpus contains 656, 380, and 378 valid descriptions for the three domain sizes, which contain 1401, 638, and 543 quantified expressions.

2.2.2 Annotating the Corpus

Since we want to design algorithms that mimic how people use quantifiers, we needed to annotate the descriptions in the corpus with their semantic representations.

To extract useful information from the QTUNA corpus, we designed a new annotation scheme, which records quantifier patterns and the ways in which these patterns were filled. Recall that quantified expressions express relations between sets (Section 1). Following Barwise and Cooper (1981), we annotated the quantified expressions in a form in which each n-ary quantifier is a function Q that takes a number of set terms as arguments. For example, a quantified expression with a binary quantifier can be written as: $Q(A, B)$.

To keep the annotation task – and the later construction of the generation algorithm – manageable, we made a few simplifications. For example, we took the view that *all* and *every* in *all/every object(s) are/is red* express the same quantifier. Table 1 lists the top-10 most frequently used quantifiers and their frequencies in our corpus. In our annotations, $A, B, ...$ are arbitrary sets. BS, BC, RS, RC, R, B, C and S stand for blue square, blue circle, red square, red circle, red object, blue object, circle and square set, respectively. O refers to the set of all objects in a situation. [7] For example, for the quantified expression

(3) All objects are red squares,

our annotation says $\text{All}(O, RS)$. More annotation examples can be found in Table 2.

Anaphors were replaced by their corresponding antecedents. For example, the description:

(4) Most of the objects are blue. Half of them are squares.

was labelled as $\text{Most}(O, B) \wedge \text{Half}(B, S)$.

Two kinds of ambiguity appear when annotating QTUNA. One is anaphoric ambiguity. For example, the pronoun *them* in the description (4) can refer to all the objects or only the blue objects. The other is syntactic ambiguity. For instance, in

7. There are also notations for second-order sets, which will be discussed later.

Notation	Surface Form(s)	Example Quantified Expression(s)	Frequency			
			N=4	N=9	N=20	Total
`all`	all; every; each	*All A are B. / All of the A are B.*	436	147	91	674
`most`	most	*Most A are B. / Most of the A are B.*	27	63	56	146
`more`	more	*There are more A than B.*	67	23	37	127
`half`	50%; half	*Half of A are B.*	76	12	15	103
`equal`	equivalent; equal/same number	*There are/is the same number of A and B.*	72	8	23	103
`some`	some	*There are some A. / Some A are B.*	2	30	66	98
`majority`	majority	*A majority of A are B. / The majority of A are B.*	24	23	14	61
`only`	only	*There are only A. / Only A are B.*	38	13	4	55
`half-rest`	half ..., the other half ...; half ..., the rest/remaining ...	*Half of A are B, and the other half are C.*	38	0	5	43
`more-half`	more than half	*More than half of the A are B.*	28	1	3	32

Table 1: Top-10 most frequently occurring quantifiers with English examples and frequencies in the three QTUNA sub-corpora.

N	Description	Meaning
4	*There are 4 squares. All objects are blue.*	$\exists_{=4}(S) \wedge \text{All}(O, B)$
9	*Most of the items are red circles, but there are a couple of blue squares.*	$\text{Most}(O, RC) \wedge \exists_{\geq 2}(BS)$
20	*All the objects in the picture are circles and the majority of them are blue.*	$\text{All}(O, C) \wedge \text{Majority}(O, B)$

Table 2: List of example descriptions from the QTUNA corpus, with their annotations. N indicates scene size (i.e., the total number of objects in the scene).

(5) Half objects are red squares and circles.

"red" can be a modifier of either "sequares" or "squares and circles". When annotating such cases we chose a "charitable" approach: if one interpretation causes a given description to be correct and logically complete and another causes it to be correct but logically incomplete,

then annotation sides with the former. This approach was chosen for all kinds of ambiguities that we encountered in our annotation work.

2.3 Hypotheses

First of all, we wanted to see how much variation in linguistic descriptions the scenes of the QTUNA experiments would permit, and how much variation there was between speakers. We were curious what quantifiers and quantifier patterns would be used and how these would be expressed linguistically; knowing this is also essential for the computational models that we were to develop later.

Following our pilot experiments, we also wanted to know how much information speakers conveyed: How often did speakers under-specify (i.e., when they did not say enough to allow a hearer to reconstruct the scene), over-specify (i.e., offer more information than necessary), and how often did they use vague quantifiers, such as "a few ...", and "many", that lack crisp borderlines? What information would be expressed explicitly and what information would be left implicit (i.e., left to be inferred by the reader). Furthermore, we were interested in knowing how the fact that one attribute (e.g., shape) is more easily expressed as a noun than the other (e.g., colour) affects its use in quantified descriptions. Given that most scenes require several quantified expressions for their logically complete description, we were also interested in what order quantifiers tended to appear in a description. Therefore, we set out to address the following questions:

When do people use vague quantifiers? People frequently use vague quantifiers, such as *many*, *some*, and *most* (see e.g. Moxey and Sanford (1993) and Sorensen (2022)). We wanted to see how the proportion of vague quantifiers in our corpora changes with scene size. The larger a domain, the harder it is to see at a glance how many objects there are in each of its set-theoretic regions (e.g., A, B, $A \cup B$, $A \cap B$, $A - B$, $B - A$, and the domain O of objects as a whole). We therefore, hypothesised ($\mathcal{H}_1$) that, *as the domain size (N) increases, more vague quantifiers appear.*

How often do speakers describe a scene completely and correctly? We say a description is complete if the scene described is the only one (modulo location) from all possible scenes of the same size that fits the description, given the background assumptions conveyed in the instructions to participants (i.e., that there are only circles and squares, and that they can only be red or blue). Since producing a complete description requires much more work (or, sometimes, is impossible) in a larger domain, we hypothesised that *larger domains give rise to a smaller proportion of complete descriptions than smaller ones* ($\mathcal{H}_2$).

A challenge for testing this hypothesis is that speakers frequently rely on inference when describing a scene. Consider

(6) Half of the objects are blue.

We will take such inferences to be part of the meaning of the sentence. So, given our background assumptions about the domain, we will take (a) to imply that the other two objects are red.

A more difficult challenge is that even simple quantified (English) expressions can harbour a considerable amount of ambiguity and vagueness. The ambiguity of *most* and *many*

is well-attested (Lidz et al., 2011; Zajenkowski & Szymanik, 2013; Kotek et al., 2015; Solt, 2016; Lappin, 2000), but even apparently simple quantifiers such as *all* and *some* are not always clear when we realise that their correspondence with the classical quantifiers $\forall$ and $\exists$ is imperfect. For instance, if I say "*some A are B*", can I be taken to convey that there is more than one A? Do I imply that some A are *not* B? These issues are widely acknowledged (e.g., Peters and Westerståhl (2006)), but far from resolved. To show that subtle nuance can matter, consider the following example from our size-4 pilots:

(7) Everything is blue. Most things are square.

If *most* simply means "more than half", then this description is incomplete, because it does not rule out the possibility that all objects are square. But if *most* means "more than half, but not all", then the description completely describes a scene with 3 blue squares and 1 blue circle.

Given the huge complexities stemming from ambiguous and vague quantifier words, we decided to simplify matters somewhat. Most importantly, *when we tested descriptions for completeness and correctness*, during annotation, we pretended that each quantified expression is associated with exactly one meaning and that this meaning is never vague but always crisp (i.e., without borderline cases). For example, "Some circles are red" was taken to mean that *at least 2 circles* are red.

Since describing larger scenes requires more work, the task itself is harder than when describing smaller scenes, so counting and other mistakes become more likely. We, therefore, expect ($\mathcal{H}_3$) that, *in larger domains, there are more descriptions that convey incorrect information.* Information is considered to be incorrect if it is not true with respect to the scene. For example, the description *all objects are blue* is incorrect if it describes a situation in which one object is red.

Are larger scenes described more elaborately? Since there is more to describe in a large domain than in a small one, we expected ($\mathcal{H}_4$) that *participants produce longer descriptions in larger scenes.*

Left-to-right order of quantified expressions. Recall that most descriptions in the QTUNA corpus consist of multiple quantified expressions. In pilot studies, speakers tended to employ two discourse structures. The first starts by describing the whole scene, e.g., "*all objects are blue*", followed by a more detailed statement, e.g., "*half of them are squares*". A second, more frequent, discourse structure cuts the set of objects into two parts, each of which is described separately.

We decided to focus on the second discourse structure, hypothesising that *the most important information tends to be stated first* ($\mathcal{H}_5$).

Most commonly, a scene is described using a succession of two quantified expressions, each of which has two set arguments; that is, each has the form $Q(A, B)$ (i.e., the most common form of quantification). Such quantifiers can be understood as being "about" the intersection of the two arguments (i.e., about $A \cap B$). Hypothesis ($\mathcal{H}_5$) says that the first of the two quantified expressions is usually "about" a larger set than the second. (For instance, "*3/4 of A are B, 1/4 are C*" is much more frequent than "*1/4 A are C, 3/4 are B*".) Sometimes, the second quantified expression is left implicit. For instance, this happens in "*3/4 of A are B*". $\mathcal{H}_5$ covers this "implicit" variant as well, predicting that "*3/4 of A are B*" is much more frequent than "*1/4 A are C*".

	$N = 4$	$N = 9$	$N = 20$
Quantified Descriptions	656	380	378
Quantified Expressions	1401	638	543
Vague Quantifiers	57	201	234
Complete Descriptions	610	175	23
Incomplete Descriptions	46	205	355
Wrong Descriptions	7	12	47
Larger Part First	123	145	99
Smaller Part First	72	54	10

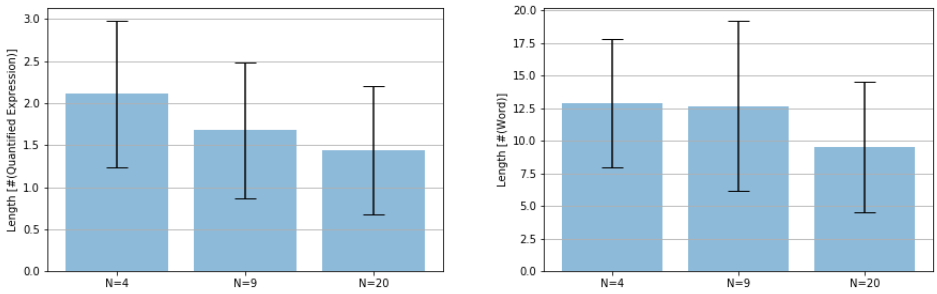
Table 3: The number of quantified descriptions, quantified expressions, incomplete descriptions, vague quantifiers, and wrong descriptions in each sub-corpus of QTUNA.

Are there any differences between the use of colour and shape? Given the well-documented primacy of colour over the shape in referring expression (e.g., Pechmann (1989) and van Deemter et al. (2012)), we expected to see that colour and shape play different roles in quantified expressions as well. Based on our pilot experiments, in which colour was often realised as an adjective, we hypothesised that ($\mathcal{H}_6$), in k-ary ($k > 1$) quantified expressions[8], *shape occurs more often in the former argument places (i.e., the A position in the quantified expression: Q* of *A* are *B) and colour in later positions.* For example, we expected to see more expressions like "*all circles are red*" than ones like "*all blue objects are circular*".

2.4 Hypothesis Testing

We tested the hypotheses introduced in Section 2.3. $\mathcal{H}_1$ asserts that vague quantifiers appear more frequently in larger scenes. In accordance with common practice (e.g., Kenney and Smith (1996)), we understand a quantifier to be vague if it permits so-called borderline cases (i.e., cases in which it is unclear whether the quantified expression is true or false)[9]. We counted the number of quantified expressions that use vague quantifiers (e.g., *many* and *few*).[10] The number of quantified expressions was compared with the total number of

8. k-ary quantified expressions are ones whose quantifier relates k sets
9. Concretely, we treat the following quantifiers we found in the corpus as vague quantifiers: *many*, *some*, *a lot of*, *lots of*, *most*, *few*, *a few*, *slightly more*, *slightly more than half*, *a small amount of*, *majority*, *minority*, *about half*, *roughly the same amount*, *amost all*, *almost half*, *many more*, *almost a quarter*, and *several*.
10. A quantifier like *m*ost was always counted as vague, despite the fact that it might acquire a precise meaning when N=4 (because when we say that *most* of a set of four 4 A are B, we can arguably only mean that *three* of the four A are B.



(a) (b)

Figure 5: The length of descriptions with respect to the domain size by means of (a) the number of quantified expressions; (b) the number of words.

quantified expressions, as listed in Table 3. The trend hypothesised in $\mathcal{H}_1$ was confirmed ($p < .001$, adjusted $p < .001$[11]) also by a binary logistic regression analysis.[12]

In order to test the second hypothesis $\mathcal{H}_2$, we annotated each description in QTUNA for being complete or not. Annotating for completeness was about whether the situation can be fully reconstructed based on a description. This was decided by the meaning of its quantified expressions and recall that, in this study, each expression is associated with a single meaning. Completeness annotation was performed by two annotators (the authors of this paper). Where disagreements occurred, the annotators discussed their initial judgement and made a final decision together. In this way, we found 46, 205 and 355 incomplete descriptions from 656, 380 and 378 descriptions of the three sub-corpus respectively table 3. As one can see, incompleteness frequencies appear to grow with scene size. Fewer than 1/10 descriptions in $N = 20$ sub-corpus are complete, most of which come from scenes with only one property combination (i.e., all the objects in a scene look alike) or two property combinations. We conducted a binary logistic regression analysis (setting completeness as the outcome variable and domain size as the predictor) on the annotated data. The result shows our hypothesis $\mathcal{H}_2$ to be confirmed ($p < .001$, adjusted $p < .001$).

For the third hypothesis $\mathcal{H}_3$, we annotated, for each description, whether it is correct or incorrect (a "wrong" description). If the property was debatable, it was considered to be correct. Such cases often occur with colour terms, for example, the colour of a red circle was sometimes described as *orange*; since only red and blue were permitted, there was no confusion possible, so we considered such descriptions to be correct. We found 7, 12 and 47 wrong descriptions for the three scene sizes. The high proportion of correctness (minimally 92.3% for $N = 20$) indicates that most of our participants understood the instructions, yet it suggests an overall association between the domain size and the error frequency, which is confirmed by a binary logistic regression analysis (setting correctness as the outcome variable and domain size as the predictor; $p < .001$, adjusted $p < .001$).

11. Adjusted p is the p-value obtained by applying Bonferroni correction, where the p-value is multiplied by 6 as there are 6 hypotheses.
12. If we had decided to count *m*ost as a precise (i.e., non-vague) quantifier when used in the N=4 domain, then this would have further strengthened the support for $\mathcal{H}_1$.

To test $\mathcal{H}_4$, we also calculated the length of each description, as defined by both the number of quantified expressions (Figure 5(a)) and the number of words (Figure 5(b)) in the description. The results show the opposite of what we expected, that is, the length of descriptions decreased. A plausible explanation may lie in the fact that speakers produced fewer complete descriptions in larger domains, as in $\mathcal{H}_2$: after all, when a task is made more complicated (in this case, because we move from smaller to larger scenes), the effect can be that participants try less hard to perform the task perfectly (i.e., they lower their standards).

Regarding our last two Hypotheses, we counted the number of descriptions that describe the larger part of a scene first (i.e., descriptions like *3/4 of A are B, 1/4 are C* or *3/4 of A are B*), and those that describe the smaller part first (i.e., descriptions like *1/4 of A are B, 3/4 are C* or *1/4 of A are B*), the numbers for each N being shown in Table 3. This confirmed the hypothesis $\mathcal{H}_5$ by a Chi-squared test ($\chi^2(2, N = 503) = 27.29, p < .001$, adjusted $p < .001$). In a similar way, we then counted the number of descriptions that place shape in the first argument while placing the colour in the latter argument (i.e., descriptions like *all circles are blue*), and the number of descriptions that order the two attributes the other way around (e.g., *all blue objects are circular*). As for shape, 489 descriptions used it in the first argument place and 121 in the second; for colour, those two numbers are 112 and 514 respectively. Consequently, a Chi-square test confirms this hypothesis $\mathcal{H}_6$ ($\chi^2(1, N = 1236) = 479.59, p < .001$, adjusted $p < .001$).

2.5 Post-hoc Observations Regarding the QTUNA Corpus

We also made a number of post-hoc observations, to be distinguished from the earlier-listed hypotheses, which were formulated before we saw the data of the experiment.

Task difficulty. The task of producing correct and complete descriptions of the scenes that make up our experiments is not always easy. In fact, we were surprised to see that speakers managed so often to perform this task successfully (see Table 3), producing a range of quantifier uses that surpassed our expectations in its variety as well.

3-ary Quantifiers. Besides binary quantifiers, we found a substantial number of 3-ary quantifiers. One class of examples is "*half ..., the other half ...*", "*one ..., the rest ...*", "*half ..., the rest ...*" and so on. Note that an expression such as (8-a) should not be confused with (8-b).

(8) a. Half of A are B, the other half are C
b. Half of A are B and half of A are C

In (8-b), the sets A and B can have a non-empty intersection, but 3(a) means that $\frac{1}{2}$ of As are B, and $(A - B) \subseteq C$.

Higher Order Quantifiers. We found a remarkable number of "higher-order" quantifiers, where quantification is not over objects but over sets of objects. For example, the word "both" in the following example quantifies over the set of colours:

(9) Half of the objects are in both colours.

Frequent examples of higher-order quantification can be found in descriptions of a situation in $N = 4$ sub-corpus where all the objects are different. Many subjects used the descriptions equivalent to (10).

(10) All possible objects are shown.

This description quantifies over elements of the Cartesian product of the colour set and the shape set (i.e., $\text{Some}(O, BS) \wedge \text{Some}(O, BC) \wedge \text{Some}(O, RS) \wedge \text{Some}(O, RC)$).

Descriptions that Rely on Implicit Information. This paper describes a set of experiments, each of which assumes a small and precisely defined domain of possibilities (e.g. scenes of N objects with only two attributes (shape and colour), each of which has only two possible values). In these cases, one can frequently infer more than is said explicitly by considering the complementary relationship of two values of one attribute. For example, if a subject says:

(11) Half of the objects are blue,

one can infer that the other half of the objects are red. Descriptions of this kind were marked as logically complete descriptions despite the appearance of incompleteness.

3. Designing Algorithms for Generating Quantified Descriptions

We aim to design a generation algorithm to construct a "product model" (see section 1) that is able to perform the same task as was given to the participants in the QTUNA experiments. Thus, we build explainable rule-based models that are inspired by our findings in Section 2.4. For two reasons, we decided not to consider deep learning approaches at this stage. The main reason is the opaqueness of such models, which makes them inherently unattractive as "product models". Secondly, as explained in section 2, the QTUNA corpus was built using a controlled elicitation experiment. The idea was to carefully select a relatively small set of inputs, for each of which a considerable amount of data is collected (e.g., in QTUNA, we collected data from more than 60 participants). Experience with other areas of computational modelling suggests that this can help us understand human behaviour, but the resulting corpus is unsuitable for training and evaluating neural models. It needs a large corpus that has a wide coverage of all kinds of input scenes.

The QTUNA scene description task involved scenes of three sizes ($N = 4$, $N = 9$, and $N = 20$). However, we do not want our generation algorithm to be limited to these scene sizes: we want them to perform well on all scenes within a certain range of sizes. We did not target scenes sized lower than 4 because we suspected that these involve quantification in very different ways, with a greater focus on exact numbers for example (see the extensive literature on "subitizable" sets, from Kaufman, Lord, Reese, and Volkmann (1949) onwards). Scenes in which there are more objects than can be counted in a few seconds were similarly beyond the scope of this study because they are likely to involve guesswork and estimation on the part of the hearer, which is not our present focus. In other words,

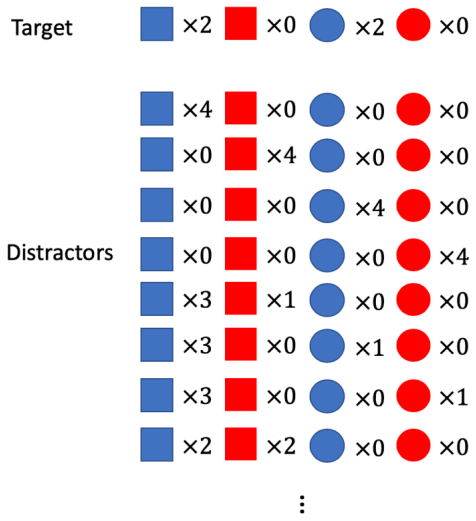


Figure 6: The target scene as one among many possible scenes ($N = 4$).

our modelling efforts focus on "mid size" domains, leaving the study of extremely small and extremely large scenes for later research.

Since our speakers were asked to produce quantified descriptions that are correct (i.e., truthful) and complete (i.e., giving as much information as can reasonably be expected), it seemed reasonable to design our algorithms with these objectives in mind. In this section, we introduce algorithms that endeavour to meet the correctness and completeness requirements as well as they can. Later on, we will evaluate our algorithms based on these two criteria and based on a comparison with the descriptions produced by human speakers as well.

To this end, akin to how we annotated the correctness and completeness of quantified descriptions in QTUNA (see Section 2.3), the algorithms need to model the meaning of each quantified expression. Recall that we pretended each quantified expression is associated with exactly one crisp meaning. The algorithms follow exactly the same simplification. For example, when a generation algorithm decides whether to use "most" as part of a description of a given scene, the algorithm will make this decision based on the meaning representation that we have associated with this word.

Below, we introduce the fundamental idea behind our algorithm, and we sketch a pipeline architecture for producing quantified descriptions, all the way from a scene to a small text. We then propose two quantified description generation algorithms, which are evaluated in Section 4. Earlier versions of these algorithms were introduced in Chen, van Deemter, and Lin (2019a).

3.1 "Referring" to a Scene

The basic idea behind both our generation algorithms is to regard the production of a quantified description as an attempt to identify, within the set of all possible scenes, what specific scene we are looking at. In other words, the idea is to view the task of our partic-

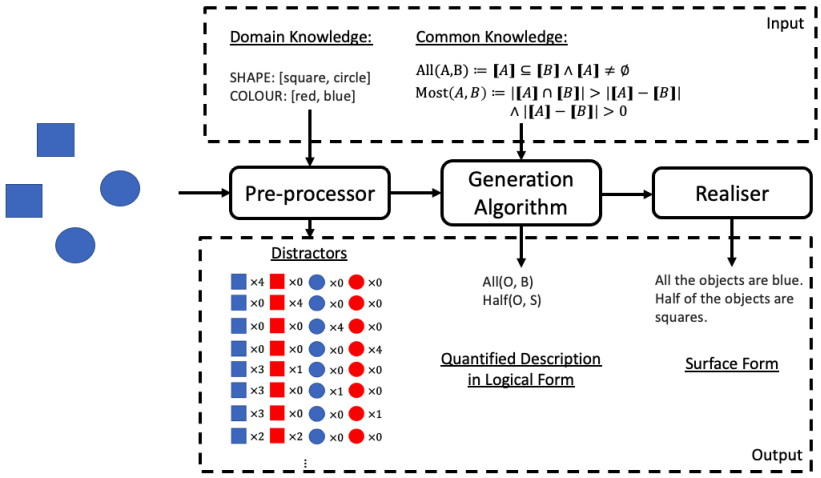


Figure 7: The pipeline of how we generate a quantified description based on a given scene. It consists of three steps: (1) pre-processing, (2) generating a quantified description in logic form, and (3) surface realization.

ipants as – very broadly – analogous to the task of referring to an object by ruling out all other objects.

Let us unpack this idea a little further, deliberately opting to use terminology familiar from work on referring expressions, in order to emphasise what the two problems have in common (despite the substantial differences, which will be discussed below). Let's call the scene that the algorithm aims to describe the *target scene*. Given a certain scene size and domain assumptions provided to our participants (i.e., what colours and shapes are permitted), the algorithm can compute how many possible scenes of this size there are. For example (as shown in Figure 6), if the target scene ($N = 4$) has two blue squares and two blue circles, then possible "distractor" scenes include a scene with 4 blue squares, a scene with 4 red squares, and so on. Generation algorithms operate by accumulating propositions that are true of the target scene but false of at least one distractor scene. For instance, if one says "*all objects are blue*", then this is true of the target scene but it will "remove" many other scenes, including the scene consisting of 4 red squares, for instance. The algorithm repeats this step until a stopping criterion is met. In simple situations, a reasonable stopping criterion is that *all* distractor scenes have been removed, though as we shall see, this idea cannot always be upheld. Let us see how these ideas can be made precise.

3.2 Generation Pipeline

Natural language generation (NLG) systems often use a pipeline architecture in which the content of the generated text is determined before its linguistic form (e.g., Reiter and Dale (2000)). We constructed our quantified description generation pipeline in line with this setup: the quantified description generation algorithms introduced in this section are responsible for determining the content of the description (i.e., essentially a logical form),

which is then turned into its linguistic form, which is a process known as *Linguistic Realisation* or *Surface Realisation* in NLG. As explained below, in order to extract the required information from the given scenes, an extra pre-processing module was inserted at the beginning of the pipeline.

Concretely, the generation pipeline consists of 3 components: a pre-processor, a quantified description generator (which runs a generation algorithm, see sections 3.3. and 3.4 below), and a surface realiser. As shown in Figure 7, given a target scene s, with its domain knowledge $\mathcal{K}_d$ (which records, among other things, how many objects and how many possible properties there are, as will be detailed later), the pre-processor calculates what kinds of distractors there are and constructs a set $\mathcal{S}$ of all possible scenes. The system then calls a generation algorithm to construct a description $\mathcal{D}$ containing a set of L quantified expressions. We write $\mathcal{D} = \{q_l(v)\}_{l=1}^{L}$, where $q(\cdot)$ is a quantified pattern with quantifier q (e.g., the *all* quantifier with two arguments can be written $\text{All}(\cdot,\cdot)$) and v is a property tuple. If v is capable of filling the slots of a quantified pattern $q(\cdot)$, we say that the pattern $q(\cdot)$ *accepts* v, and we write $q(v)$. The generation algorithm makes a selection from a set of quantified patterns $\mathcal{Q}$, based on the common knowledge $\mathcal{K}_c$ (i.e., meanings of all quantifiers) defined on $\mathcal{Q}$. Finally, with a set of logical forms $\mathcal{D}$, a simple template-based surface realiser (Section 3.5) is employed to map the logical form $\mathcal{D}$ into actual natural language text.

This generation system requires two types of knowledge:

Domain Knowledge. This is the list of all possible attributes and their possible values, with which the pre-processor could compute what distractors there are, and thus construct the set S. This knowledge is stored as a set of key-value pairs. For example, matching the current experimental setting of QTUNA, its domain knowledge is $\{\text{SHAPE} : [square, circle], \text{COLOUR} : [red, blue]\}$.

Common Knowledge. This is a body of knowledge that corresponds to the quantified patterns in $\mathcal{Q}$. For a quantified pattern $q(\cdot)$, this knowledge base includes the meaning of the quantified pattern and a set of possible property tuples that could be assigned to v. The meaning of a quantified pattern has two parts: its semantics and its pragmatics. For example, the semantics of $\text{All}(A, B)$ asserts that $[\![A]\!] \subseteq [\![B]\!]$. The pragmatics says that $[\![A]\!]$ is not empty. Determining the semantics and pragmatics of each English quantifier term is difficult in general, but the QTUNA corpus allowed us to choose definitions that match the majority usage in that corpus. The reason why we distinguish between semantics and pragmatics will become clear in the following section. Table 4 lists the quantifiers we considered in the current version of the quantified description generation algorithm. We decided to use only the most frequent quantifiers. Note that, since we assign each quantifier a precise (i.e., non-fuzzy) meaning, which causes quantifiers like *some* and *a few* to have exactly the same meaning, we chose the most frequent one among the quantifiers with the same meaning. Quantifiers like *few* and *many*, which have attracted a lot of attention from researchers, are not included in our system since they have extremely low frequency in our corpus (that is, *few* appears 2 times and *many* 13 times).

Quantifier	Semantics	Pragmatics	Quantifier	Semantics	Pragmatics
All(A, B)	$[\![A]\!] \subseteq [\![B]\!]$	$[\![A]\!] \neq \emptyset$	Fewer(A, B)	$\lvert[\![A]\!]\rvert < \lvert[\![B]\!]\rvert$	$[\![A]\!] \neq \emptyset$
Only(A)	$[\![A]\!] = [\![O]\!]$	-	Equal(A, B)	$\lvert[\![A]\!]\rvert = \lvert[\![B]\!]\rvert$	$[\![A]\!] \neq \emptyset$
Half(A, B)	$\lvert[\![A]\!] \cap [\![B]\!]\rvert = \lvert[\![A]\!] - [\![B]\!]\rvert$	$[\![A]\!] \neq \emptyset$	Most(A, B)	$\lvert[\![A]\!] \cap [\![B]\!]\rvert > \lvert[\![A]\!] - [\![B]\!]\rvert$	-
Some(A, B)	$\lvert[\![A]\!] \cap [\![B]\!]\rvert \geq 2$	$\lvert[\![A]\!]\rvert > \lvert[\![B]\!]\rvert$	Half-rest(A, B, B')	$\lvert[\![A]\!]\rvert = 2\lvert[\![B]\!]\rvert = 2\lvert[\![B']\!]\rvert$	-
Some(A)	$\lvert[\![A]\!]\rvert \geq 2$	$\lvert[\![O]\!]\rvert > \lvert[\![A]\!]\rvert$	Minority(A, B)	$\lvert[\![A]\!]\rvert > 2\lvert[\![B]\!]\rvert$	-
Only-1(A)	$\lvert[\![A]\!]\rvert = 1$	-	All-Comb(O)	All property combinations appear.	-
More(A, B)	$\lvert[\![A]\!]\rvert > \lvert[\![B]\!]\rvert$	$[\![B]\!] \neq \emptyset$			

Table 4: List of quantifiers used in our quantified description generation system and their meanings.

3.3 A Greedy Algorithm

As said, we view the quantified description generation task as a task of ruling out distractor scenes. One can view this as a search problem, namely, the problem of finding a set of quantified expressions that removes all (or as many as possible) distractors. This search can be performed by means of a *greedy* algorithm: working iteratively, this algorithm keeps selecting (and including into the quantified description) quantified expressions that jointly rule out the largest possible number of distractor scenes.

We sketch the greedy algorithm for quantified description generation (abbreviated as QDG-GREEDY) in Algorithm 1. The algorithm takes a target scene s, a set $\mathcal{S}$ of all possible scenes with the same domain as s (calculated by the pre-processor), a set of quantified patterns $\mathcal{Q}$ with their corresponding meanings (stored in $\mathcal{K}_c$) as inputs, and outputs a set $\mathcal{D}$ of quantified expressions in logical form.

The algorithm initialises the description $\mathcal{D}$ as an empty set, then inserts quantified expressions $q(v)$s iteratively into $\mathcal{D}$. During each iteration, QDG-GREEDY pluralises the $q(v)$; by this we mean adding a plural marker where necessary – namely whenever a property appears multiple times in the target scene s (e.g., Some(S, R) acquires a plural marker if there is more than one red square in the scene. For example, suppose the quantified expression is All(S, R) (meaning that all the squares are red) and the target scene contains two red squares; the expression is pluralised as All$(\langle S, pl\rangle, \langle R, pl\rangle)$ indicating that multiple squares in the target scene are red, in which, from now on, each argument is represented as a tuple and pl stands for plural while sg stands for singular. Pluralisation serves two purposes. The first is to determine the pragmatics of $q(v)$, which is then used for deciding how many distractors are left after selecting a certain quantified expression. For instance, the plurality of All$(\langle S, pl\rangle, \langle R, pl\rangle)$ could rule out distractors that contain only one red square. The second purpose is to decide the surface form of the quantified expression in English, as detailed in Section3.5. The algorithm then calls the function `FindBestQuantifiedExpression` (line 4) to choose the quantified expression that rules out the most distractors from all possible quantified expressions. Specifically, `FindBestQuantifiedExpression` checks, for each possible quantified expression $q(v)$, whether this expression fits the target scene based on the

Algorithm 1 The Greedy Algorithm for Generating Quantified Descriptions

Input: A target scene s, a set $\mathcal{S}$ of all possible scenes, a set of quantified patterns $\mathcal{Q}$, the common knowledge $\mathcal{K}_c$ defined on $\mathcal{Q}$.

Output: A quantified description $\mathcal{D}$ of s that uses conjunctions of single or multiple $q(v)$.

1: $\mathcal{D} \coloneqq \{\}$
2: **while** $\mathcal{S} \neq \{s\}$, and $|\mathcal{D}| < \delta$ **do**
3: $\quad q(v) \coloneqq$ `Pluralise`$(q(v), s)$
4: $\quad q(v) \coloneqq$ `FindBestQuantifiedExpression`$(s, \mathcal{S}, \mathcal{Q}, \mathcal{K}_c)$
5: $\quad$ **if** $q(v) = \emptyset$ **then**
6: $\quad\quad$ **break**
7: $\quad \mathcal{D} \coloneqq \mathcal{D} \cup \{q(v)\}$
8: $\quad \mathcal{S} \coloneqq \{s' \in \mathcal{S} : q(v)$ is true for $s'\}$

meaning (including both semantics and pragmatics) of $q(v)$ defined in $\mathcal{K}_c$. If yes, it calculates the number of distractors that can be ruled out by only using $q(v)$'s semantics. We call the number of distractors that a quantified expression $q(v)$ rules out in a given situation the *Discriminatory Power* of $q(v)$. The `FindBestQuantifiedExpression` function will return the quantified expression with the highest discriminatory power. If none of the candidate quantified expressions has discriminatory power, then the function returns an empty set.

To see why only the semantics, and not the pragmatics, of a quantified expression, is used for computing discriminatory power (i.e., for deciding whether to include a given quantified expression into the quantified description) consider, by way of an example, the expression All(C, B) (i.e., *All circles are blue*). Its semantics says (see Table 4) that the set of circles is a subset of the set of blue objects, and its pragmatics says, among other things, that there exists at least one circle. If the pragmatics of the expression contributed to its discriminatory power, then the algorithm would end up adding this quantified expression to a description even when the quantified expression's sole contribution is the (pragmatic) requirement that at least one object is a circle – as would happen when other quantified expressions, previously added to $\mathcal{D}$ (for example, All(O, B)), already ensure that the set of circles is a subset of the set of blue objects.[13] Additionally, as listed in Table 4, a number of quantifiers have the same pragmatics. So, if the pragmatics was taken into account when the algorithm determines the discriminatory power of a quantified expression, then some very different quantified expressions would end up having the same discriminatory power. To us at least, it was surprising to see that the distinction between semantic and pragmatic information – which although it is fairly commonplace in linguistics can feel a bit artificial in some computational settings – had such concrete practical relevance for present purposes.

Line 5 of the algorithm checks whether $q(v)$ is empty. If it is, then the algorithm concludes the *while* loop (line 6). If $q(v)$ is not empty, it is added to $\mathcal{D}$ (line 7) and the distractor scenes are removed from $\mathcal{S}$ based on both semantics and pragmatics of $q(v)$. Line 2 of the Algorithm 1 talks about the *Stop Criteria*. Generation terminates when all distractors are removed from $\mathcal{S}$ or the length of the generated description $\mathcal{D}$ reaches an

13. If plurality is also treated as a part of pragmatics, then the pragmatics of the quantified expression All$(\langle C, pl\rangle, \langle B, pl\rangle)$ says that there are at least two circles. This would exacerbate the above effect.

Algorithm 2 The Incremental Algorithm for Generating Quantified Descriptions

Input: A target scene s, a set $\mathcal{S}$ of all possible scenes, a set of quantified patterns $\mathcal{Q}$, the common knowledge $\mathcal{K}_c$ defined on $\mathcal{Q}$, a Quantifier Preference Order defined on $\mathcal{Q}$, a set of all possible property tuples in the domain $\mathcal{V}$, and a property preference order defined on $\mathcal{V}$.

Output: A quantified description $\mathcal{D}$ of s that uses conjunctions of single or multiple $q(v)$.

1: $\mathcal{D} \coloneqq \{\}$
2: **for** each q in $\mathcal{Q}$ (in order of the Quantifier Preference Order) **do**
3: **for** each v in $\mathcal{V}$ such that q accepts v (in order of the Property Preference Order) **do**
4: $q(v) \coloneqq$ `Pluralise`$(q(v), s)$
5: **if** $q(v)$ is true for s, and $\mathcal{D} \not\models q(v)$ **then**
6: $\mathcal{D} \coloneqq \mathcal{D} \cup \{q(v)\}$
7: $\mathcal{S} \coloneqq \{s' \in \mathcal{S} : q(v)$ is true for $s'\}$
8: Until $\mathcal{S} = \{s\}$ or $|\mathcal{D}| \geq \delta$

upper bound δ. The idea of setting an upper bound comes from the observation that, in QTUNA, descriptions were remarkably constant across domain sizes (see $\mathcal{H}_4$).

Note that in line 4 of this algorithm, the `FindBestQuantifiedExpression` is likely to find multiple quantified expressions that have the same discriminatory power (i.e., several "best" expressions). Instead of trying to choose intelligently (and in order to increase the variation in generated quantified descriptions), the `FindBestQuantifiedExpression` randomly returns one of these "best" expressions.

3.4 An Incremental Algorithm

We have seen that the Greedy algorithm iteratively selects the quantified expressions that have the highest discriminatory power. From a cognitive viewpoint, however, there could be thought to be something slightly suspect about an algorithm that needs to perform such a complicated calculation: alter all, `FindBestQuantifiedExpression` has to check, for each quantifier pattern and all its possible values, how many scenes would be ruled if these were selected. Moreover, when we examined the QTUNA dataset more closely, we found that some quantifiers patterns are far more frequent than others, and some choices of properties to fill a given pattern are far more frequent than others. For example, akin to what $\mathcal{H}_5$ indicates, we found that if *all* fits in any of the properties in a scene, subjects tend to use *all* to construct a quantified expression. Building on these observations, a natural idea would be to compose an ordered sequence of quantifiers, and an ordered sequence of fillers (i.e., property tuples), reflecting the different degrees of "popularity" of different quantifiers and different fillers. The algorithm can then make use of this ordered sequence to determine in what order to consider the different types of expressions for inclusion in the generated description. Analogous to the "preference orders" of attributes (like colour, size, etc.) that are employed in the generation of referring expressions (as proposed in Dale and Reiter (1995)), one would ultimately like to understand the reasons behind these preference orders, for instance in terms of codability (see van Deemter (2016, chapter 3) for discussion).

Lacking such a deep understanding for the moment, we considered the following two types of sequences[14]:

Quantifier Sequence. Inspired by the fact that some quantifiers occur more frequently than other quantifiers (as shown in Figure 4), quantified expressions that use frequent quantifiers like *all*, *half* or *most* should have high priority (i.e., they should occur early in the Preference Order).

Property Sequence. Analysis of QTUNA (see $\mathcal{H}_6$) suggested that, for patterns of the form All(A, B), the first argument, A, is more often a SHAPE property, whereas B is more often a `COLOUR`. For example, the algorithm should prefer the property tuple (S, R) over (R, S).

The algorithm incrementally generates the description by considering possible quantifiers and fillers one by one, starting at the top of the sequence, and working its way from the top of the preference order downwards. Given the analogy with the incremental algorithm for referring expressions generation (Dale & Reiter, 1995), we call the algorithm the incremental algorithm (abbreviated as QDG-IA). Likewise, we will speak of the *Quantifier Preference Order* (instead of quantifier sequence) and the *Property Preference Order* (instead of Property Sequence).

Note that in addition to the inputs of the QDG-GREEDY algorithm, as shown in Algorithm 2, QDG-IA require two pre-defined preference orders introduced above. Given these inputs, the QGD-IA algorithm will go through all the quantified patterns $\mathcal{Q}$ in the order of quantifier preference order. In each iteration, for the selected quantified pattern $q(\cdot)$, QDG-IA will test all possible property tuples accepted by that pattern in the order of property preference order. Recall that the information which $q(\cdot)$ accepts which property tuple can be found in $\mathcal{K}_c$. The algorithm then calls the `Pluralize` function on the quantified expression, which is the same manipulation done by QDG-GREEDY.

Line 5 of Algorithm 2 involves some important deviations from Dale and Reiter's algorithms. Here, our algorithm first tests whether $q(v)$ is correct as a quantified expression for s; the test is performed by using both its semantics and pragmatics. Subsequently, the algorithm tests whether $q(v)$ does not follow from the description $\mathcal{D}$ (i.e., $\mathcal{D} \not\models q(v)$) [15], ensuring that $q(v)$ rules out one or more further scenes (i.e., it is not logically superfluous). Crucially, the latter test uses only the semantics of $q(v)$, not the pragmatics. In the case of the present algorithm, the different roles of semantic and pragmatic information (in this case: the information provided by the plural form) is possibly even more striking than in the case of the Greedy algorithm. For example, suppose we want to generate a quantified description for a scene that consisting 2 blue squares and 2 blue circles, and the quantifier *all* has the highest priority in the quantifier preference order. In its first iteration, the algorithm produces a quantified expression like "*all objects are blue*". In the second iteration, if the pragmatics was used for validation, the algorithm could add "*all circles are blue*", whose semantics contributes no new information at all, but whose pragmatics (i.e., the claim that

14. Further details of both the Quantifier Sequence and the Property Sequence are given below.

15. Logical consequence is implemented by calculating the set of scenes that are removed by a given expression (or set of expressions). Thus, $\mathcal{D} \models q(v)$ means that the set of distractor scenes removed by $q(v)$ is a subset of the set of distractor scenes removed by $\mathcal{D}$.

there are at least two circles) rules out all those distractor scenes that contain less than two circles (which would cause it to pass the second test of line 5). The resulting description, "*All objects are blue and all circles are blue and ...*" (which can be made logically complete by adding "*... and there are squares*") would sound strange because, intuitively, the second clause is logically redundant given the first.[16]

Once the above two conditions have been validated, $q(v)$ is appended at the end of the description and the scenes for which $q(v)$ is not true are removed from $\mathcal{S}$. Both semantics and pragmatics are used for removing such distractors. The generation terminates according to the same criteria as the QDG-GREEDY algorithm.

As for the design of preference orders, we started with testing the following settings, once again based on the analysis of the corpus. The quantifier preference order is a linear preference order, namely:

$$\text{All}(\cdot,\cdot) \succ \text{Everything}(\cdot) \succ \text{Only}(\cdot) \succ \text{Half}(\cdot,\cdot) \succ \text{Half-rest}(\cdot,\cdot,\cdot) \succ \text{Equal}(\cdot,\cdot) \succ \text{Most}(\cdot,\cdot) \succ \text{More}(\cdot,\cdot) \succ \text{Minority}(\cdot,\cdot) \succ \text{Fewer}(\cdot,\cdot) \succ \text{Some}(\cdot,\cdot) \succ \text{Some}(\cdot) \succ \text{Only-1}(\cdot).$$

The second-order quantifier All-comb (see Table 4) is only applicable to a small number of scenes but is used very frequently for those scenes. Therefore, although it has a relatively low overall frequency across the whole corpus, we still assign it a high priority.[17] The property preference order was designed by following some constraints, for example, `SHAPE` properties have higher priorities in the first argument places and compounded properties (e.g., RS and BC) are more preferred than singular properties (e.g., R, C, and B).

However, when we ran the algorithm, we found that some quantified patterns that have low preference are never chosen by the algorithm, causing the generated descriptions to only use a very limited set of patterns. For example, the pattern $\text{All}(\cdot,\cdot)$ has a higher preference than the pattern $\text{Only}(\cdot)$, and consequently the latter is never chosen, because its meaning is covered by the former. (For example, the meaning of "*there are only squares*" is covered by that of "*all objects are squares*".) To increase variety, we introduced a probability θ with which the QDG-IA performs a one-off re-ordering move; for the work reported in this paper, we set θ to 0.1. Re-ordering was not performed across the entire preference order, but only within certain groups of quantifiers that have high meaning overlap with each other. To be precise, we used the following partitioning of the Preference Order of quantifiers (each $\{\cdot\}$ represents a partition):

$$\begin{aligned}&\text{All-Comb} \succ \\ &\{\text{All}(\cdot,\cdot) \succ \text{Everything}(\cdot) \succ \text{Only}(\cdot)\} \succ \\ &\{\text{Half}(\cdot,\cdot) \succ \text{Half-rest}(\cdot,\cdot,\cdot) \succ \text{Equal}(\cdot,\cdot)\} \succ \\ &\{\text{Most}(\cdot,\cdot) \succ \text{More}(\cdot,\cdot) \succ \text{Minority}(\cdot,\cdot) \succ \text{Fewer}(\cdot,\cdot)\} \succ \\ &\{\text{Some}(\cdot,\cdot) \succ \text{Some}(\cdot)\} \succ \text{Only-1}(\cdot).\end{aligned}$$

16. These observations might have applications in other areas of language use as well, for instance, Gricean conversational implicatures (Grice, 1975). Imagine the Gricean scenario in which an academic referent "praises" one of his students for having nice handwriting (implying that the student is academically inept and should not be hired). Our observations suggest that it would be odd for this academic to make the same utterance as part of a conversation in which the student's handwriting had already been favourably commented upon.
17. $A \succ B$ means that A follows B in the preference order.

Once the algorithm has decided to conduct a one-off move, the order of quantifiers within that part are re-ordered at random.

3.5 Surface Realisation

Surface Realisation is typically the last stage in an NLG pipeline, where abstract structures are turned into concrete sentences. In the present case, Surface Realisation turns the logical forms produced by the Greedy and Incremental algorithm into actual stretches of English text. Though this is not the stage of the pipeline on which our computational model focuses, it cannot be omitted because, without Surface Realisation, it would be much more difficult for human judges to evaluate the output of the algorithm: people are used to interpreting and judging text, not abstract representations.

Our system uses a simple template-based surface realiser (see e.g. (van Deemter, Theune, & Krahmer, 2005) for comparison with other types of Linguistic Realisation). For each quantified pattern, there is a specific template. For example, for All$(\cdot, \cdot)$, we have a template:

(12) All of ⟨ARGUMENT-1⟩ ⟨COPULA⟩ ⟨ARGUMENT-2⟩

where ⟨COPULA⟩ will be realised into *is* or *are* depending on the plurality of the first argument of the generated quantified expression that uses this pattern. When filling these slots with chosen properties, some simple syntactic and morphological operations are employed. For example, if a `COLOR` property takes the first place of a quantified pattern, a noun is appended to package it into a noun phrase (i.e., *red* → *red object*). If a property has a plural suffix, the surface form of the property is mapped into its plural form. A number of further constraints, specific to particular quantified patterns, were also encoded in the realiser.

The present work has focused on the way in which speakers use a variety of quantifiers, which is why Linguistic Realisation of sentences and texts was kept simple and could be improved in many ways. One significant limitation of the way in which the abstract patterns generated by the algorithms of the previous sections are put into words is that our wordings do not use *anaphora* yet. This is despite the occurrence of many different types of anaphoric expressions in our corpus, for example as when a quantified expression is followed by "*Half of them are red*" (see also section 2.2.2). Anaphoric patterns were particularly prevalent in quantified expressions with 3-ary quantifiers, for example as in "*Half of the objects are red, the other half are blue*". Using anaphora judiciously without creating unwanted ambiguities is quite doable in general, but the topic is not without its problems (e.g., Kamp and Reyle (2013, Chapter 4)). We expect that, by addressing these issues, future Linguistic Realisation modules will be able to produce even more human-like descriptions of the scenes on which we are focusing.

4. Evaluating the Generated Quantified Descriptions

Although our algorithms were informed by extensive elicitation experiments, we wanted to gain additional insights into the quality of generated descriptions though some further experiments. We were especially curious how "human-like" the generated descriptions are, and how correct and informative.

Previous studies on evaluating the human-likeness of a computational language production model tend to use corpus-based evaluation: the model generates outputs (e.g., sentences or logical forms) and these outputs are compared with a corpus using a similarity measure (e.g., van Deemter et al. (2012)), such as DICE (Dice, 1945) or BLEU (Papineni, Roukos, Ward, & Zhu, 2002). However, there are two insurmountable problems with using such a methodology in the present situation.

First, the quality of a quantified description cannot easily be measured automatically. Consider the quantified expression $\text{Few}(O, S)$ once again. Suppose the target scene is a situation in which 5 out of 20 objects are squares; then is it correct to say that $\text{Few}(O, S)$, or does this underestimate the number of squares? And if $\text{Few}(O, S)$ is all that is said about the proportion of objects being squares, is this sufficiently informative or not? We are not aware of any metric that would give us reliable answers to these questions. Therefore, we decided not to use corpus-based evaluation, but to conduct two evaluation studies: a human judgement study (i.e., asking expert human judges to rate the generated quantified descriptions) and a scene reconstruction study (i.e., asking human subjects to reconstruct the input scenes given the generated quantified descriptions).

Second, since we designed our algorithms based on the QTUNA corpus, it would be insufficient to evaluate them on the same corpus again, since this would fail to distinguish between training and test data. (Borrowing terminology from machine learning, it would risk letting the model over-fit the corpus.) To avoid this problem, we selected our experimental materials not only from our QTUNA corpus but also from scenes that do not appear in QTUNA.

Concretely, we divided the evaluation experiments into experiment A and experiment B. For experiment A, we randomly selected 3 or 4 scenes from each of the 3 sub-corpora of QTUNA to construct a set of, in total, 10 scenes, each of which was paired with 3 descriptions: one by QDG-IA, one by QDG-GREEDY, and one selected at random from our corpus. A number of example scenes, paired with their descriptions, are listed in Table 5. For experiment B, we focused on three new domain sizes namely $N = 6$, $N = 10$, and $N = 16$. For each of these, we sampled 6 scenes, each of which was paired with 2 descriptions: one by QDG-GREEDY and one by QDG-IA. Finally, we have 66 scene-description pairs ready to be evaluated.

To assess the quality of each description, we used two different methods: a method based on quality judgements by human experts, and a task-based method in which readers were asked to reconstruct the scenes that are described.

Baseline. To put the performance of our algorithms in a broader context, we also tested a variant of QDG-IA-RANDOM of QDG-IA where, instead of using the preference order of QDG-IA, attributes are chosen in random order (Table 5). The resulting outputs are strikingly unnatural. For example, given the mechanism of the incremental algorithm, some quantifiers (e.g., "some", "more") are more likely to be repeatedly chosen than others (e.g., "all", "half"). For example, for the quantifier "all", it is impossible for the expression "*All squares are red*" to be true if the expression "*All squares are blue.*" has been generated, while, for the quantifier "some", the description "*Some squares are red. Some squares are blue.*" can be true. If such quantifiers rank high in the preference order used by QDG-IA-RANDOM, the algorithm may use the same quantifier again and again. Since the resulting QDs are

Scene	Model	Description
BS:2 RS:2 BC:0 RC:0	Human	All the objects are squares and half of them is blue.
	QDG-IA	Every object is square. There are equally many blue squares and red squares.
	QDG-GREEDY	Half of the objects are blue squares, the rest are red squares.
	QDG-IA-RANDOM	Some objects are red squares. Some objects are blue squares.
BS:2 RS:2 BC:5 RC:0	Human	Two objects are red squares. Two objects are blue squares and the remainder is blue.
	QDG-IA	Every circle is blue. Half of the squares are blue. More than half of the objects are blue circles.
	QDG-GREEDY	Half of the squares are red, the rest are blue. Most of the objects are blue circles.
	QDG-IA-RANDOM	A minority of the objects are red squares. A minority of the objects are blue squares. Less than half of the objects are squares. A minority of the objects are red. Less than half of the blue objects are squares.
BS:9 RS:2 BC:8 RC:1	Human	There is a mixture of squares and circles. Most of them are blue. Some of them are red.
	QDG-IA	All possible objects are shown. A minority of the objects are red squares. Less than half of the objects are blue circles. Less than half of the objects are blue squares. Less than half of the objects are circles.
	QDG-GREEDY	All possible objects are shown. A minority of the objects are red squares. Less than half of the objects are blue circles. Less than half of the objects are blue squares. Less than half of the objects are circles.
	QDG-IA-RANDOM	There are fewer blue circles than blue squares. There are fewer red squares than blue circles. There are fewer circles than squares. More than half of the circles are blue. A majority of the red objects are squares.

Table 5: Examples of quantified descriptions produced by humans, by QDG-IA, and by QDG-GREEDY. The numbers in the Scene column represent the number of objects of each type (e.g., the first scene consists of two blue squares and two red squares).

often unwieldy (see Table 5), we decided not to make QDG-IA-RANDOM part of our formal evaluation in the following two sections.

4.1 Evaluation Using Human Judgments

Settings. We recruited 4 annotators, academics from Utrecht University, none of whom had been involved in our research. Two were young lecturing staff in computational linguistics and two were senior lecturing staff in computational logic and formal argumentation. All 66 scene-description pairs (from both experiments A and B) were put together and randomly allocated to our four judges. Each of them judged 33 scene-description pairs. Thus, each scene-description pair was judged by two judges and was judged from three aspects: correctness, completeness, and naturalness.

However, correctness and completeness of a description is not an "all or nothing" affair, especially when larger domains are involved, which frequently give rise to descriptions that contain vague quantifiers. The same is true for the perceived naturalness of the description. As is often done in Natural Language Generation (Gatt & Krahmer, 2018), we used a

	Model	Naturalness	Informativity	Correctness
Experiment A	Human	3.45	4.05	4.6
	QDG-IA	2.9	3.95	4.55
	QDG-GREEDY	3.65	3.8	4.8
Experiment B	QDG-IA	3.5	3.78	4.47
	QDG-GREEDY	3.44	3.97	4.22

Table 6: Average scores for each algorithm and for human-produced descriptions, by naturalness, informativity, and correctness as annotated by our four human judges.

gradable scale. Judges were asked three Likert-scaled questions in each case: 1) Naturalness: *On a scale of 1-5, how likely do you think it might be that this description was uttered by a human?* [1=very unlikely, 5=very likely]; 2) Informativity: *On a scale of 1-5, do you believe the description is as informative as it can be expected to be?* [1=description is not even nearly informative enough, 5=description gives as much information as is possible]; 3) Correctness: *On a scale of 1-5, how correctly do you consider this description to be?* [1=the description is not ad all correct, 5=everything the description says is correct]. Note that when judges make judgments the words naturalness, informativity and correctness were invisible. In addition, our instructions said "*Please note that we are mainly interested in the logic of how people describe the scene, and less in the details of the wording, so please disregard minor syntax errors and typos*". Because in experiment A, the first question was asked about a human-produced description as well as two algorithm-generated descriptions, this setup allowed us to perform what is essentially a *Turing Test*. The other two questions offered invaluable formative evaluation.

On the basis of the nature of the task and the algorithms of quantified description production, we formulated a number of evaluation hypotheses: 1) Humans perform better at naturalness than QDG-IA and QDG-GREEDY ($\mathcal{EH}_1$); 2) Both algorithms perform better at informativity and correctness than humans because both of them were explicitly designed to optimise informativity and correctness ($\mathcal{EH}_2$); 3) QDG-IA performs better at naturalness than QDG-GREEDY ($\mathcal{EH}_3$). We reasoned that, in referring expression generation, the incremental algorithm offered greater human-likeness than the greedy algorithm (Dale & Reiter, 1995; van Deemter et al., 2012), so why should things be different this time?

Results. Table 6 shows the scores from the judges. Both algorithms scored well over 3 in all except one cell, confirming our impression that the descriptions tended to be of very respectable quality.

As for our evaluation hypotheses, our first evaluation hypothesis, $\mathcal{EH}_1$, was rejected: in terms of naturalness, QDG-GREEDY performed well above expectation, gaining a slightly better score than the human speakers, although the difference did not reach significance (using a paired t-test: $t = -0.4972, p = .6220$). QDG-IA had a lower score, but this also did not amount to a significant difference from human speakers ($t = 1.1133, p = .2726$); the difference between QDG-IA and QDG-GREEDY was significant ($t = -1.6310, p = .1111$).

Though "no difference" results always need to be approached with caution, the rejection of $\mathcal{EH}_1$ might be interpreted as our algorithm passing a limited kind of Turing test, focusing on a limited type of language use, of course, since it suggests that the perceived quality of the algorithm was indistinguishable from human speakers. In an effort to understand the low naturalness performance of QDG-IA, we had a closer look at the cases where QDG-IA had particularly low scores. We found that these almost always contained vague quantifiers (e.g., *few*, *most*), where the semantic and pragmatic definitions of which our algorithms made use were especially tentative. Moreover, vague quantifiers were used disproportionally often in the scenes of Experiment A, and far less in the scenes of Experiment B; accordingly, the QDG-IA scored much better on naturalness in Experiment B. We surmise that a possible reason for the low naturalness performance of QDG-IA is that the semantics of the vague quantifiers in $\mathcal{K}_c$ is not as accurate as it could have been. For instance, the currently-used semantics of *most* is the same as that of *more than half*, which is a precise quantifier. The effect of using more accurate, empirically based, definitions of vague quantifiers, which requires further comprehension experiments, will be investigated in further work.

Our analysis of the second evaluation hypothesis, $\mathcal{EH}_2$, shows some of the hidden difficulties of the description task that our algorithms solve. Human speakers, and both of our algorithms, all performed similarly well in terms of informativity and correctness (in terms of informativity, human/QDG-IA has $t = 0.2439, p = .8087$ and human/QDG-GREEDY has $t = 0.5903, p = .5585$; in terms of correctness, human/QGD-IA has $t = 0.1305, p = .8968$ and human/QDG-GREEDY has $t = -0.6016, p = .5510$). To understand why, we decided to separately calculate the average informativity score for those descriptions in experiment B that were *logically complete* (i.e., the algorithm stopped when $S = \{s\}$). For this reduced set of descriptions, the average scores for QDG-IA and QDG-GREEDY were a mere 3.88 and 4.1, instead of the score that one might expect, namely 5. One possible explanation is that our algorithms judged the logical correctness and completeness of these descriptions by taking both their semantics and their pragmatics into account (as discussed in Sections 3.3 and 3.4), which is something our judges may have disagreed with. Alternatively, judges may sometimes have had a lapse of concentration.

The last evaluation hypothesis, $\mathcal{EH}_3$, was also rejected, as there was no significant difference between the naturalness performance of QDG-IA and QDG-GREEDY (Experiment A: $t = -1.6310, p = .1111$; Experiment B: $t = 0.1656, p = .8690$). This may be because the preference order that we proposed for quantified patterns has much higher complexity than that of properties (or attributes) in the task of referring expression generation. In particular, the number of quantifiers is considerable, and, because of our "one-off" re-ordering move, our preference order of quantifiers was not linear. It is possible that a different preference order would have led to better results for QDG-IA, but it seems equally possible that the idea of using a preference order to determine the choice of quantifier patterns – on which the Incremental Algorithm is based – is simply not on the right track, and that a simpler "greedy" approach leads to results that are equally good.

4.2 Evaluation Using Scene Reconstruction

Settings. We recruited 20 undergraduate students from Utrecht University 13 of whom major in Artificial Intelligence; the other 7 study a variety of other subjects. The descrip-

tions in experiment A were randomly allocated to all participants. Each description was used for reconstructing the paired scene four times (i.e., by four participants). The descriptions in experiment B were allocated in the same way, except that each pair was assigned twice instead of four times. Each participant reconstructed a total of 8 or 9 scenes.

Given a description and the domain size of the paired scene, we demanded each participant to write down the number of objects (i.e., the number of BS, RS, BC, and RC) in the scene by asking "*please tell us about a scene that could be described by the description*". We chose to ask participants to write numbers instead of drawing scenes, in order to encourage them to disregard the location of each object in the scene. Participants had not seen any of the QTUNA scenes before, which makes the reconstruction task become tough. Therefore, before starting, we provided each participant with two examples to show them what the reconstructed scenes might look like. In addition, considering that some descriptions have multiple possible corresponding scenes, we told participants: "*In those cases, please choose an answer (number) that you consider to be consistent with the description*".

Given the above settings and the hypotheses of the human judgement study, we formulated two evaluation hypotheses. Firstly, we hypothesised that reconstructions based on descriptions generated by QDG-IA and QDG-GREEDY are more similar to the input scenes than those produced by humans ($\mathcal{EH}_4$). We expected this because these algorithms, especially the QDG-GREEDY are designed to be as logically complete as possible. Since the more complete the generated descriptions are, the easier for them to be reconstructed. Secondly, we hypothesised that descriptions produced by QDG-IA let readers reconstruct scenes more accurately than QDG-GREEDY ($\mathcal{EH}_5$).

Similarity between Reconstructions. A key part of our analysis is the metric that we used to measure the similarity between the reference scene and a reconstructed scene. Given a reference scene and a reconstructed scene, we care about how many "swaps" are needed to convert one into the other. Concretely, we propose the SWAP metric, which takes the absolute differences between the cardinalities of each of the four types of object in the reference scene and in the reconstructed scene, takes the sum of these, then divides that sum by 2 times the domain size.

For instance, suppose the reference scene is: $\{\mathrm{BS}:2, \mathrm{RS}:1, \mathrm{BC}:0, \mathrm{RC}:1\}$, where each number represents the cardinality of the relevant type of object. Suppose one of the reconstructed scenes is: $\{\mathrm{BS}:2, \mathrm{RS}:1, \mathrm{BC}:1, \mathrm{RC}:0\}$. Then

$$\mathrm{SWAP} = \frac{|2-2| + |1-1| + |0-1| + |1-0|}{2 \times 4} = 1/4. \tag{1}$$

The lower the SWAP score, the lower the better the reconstruction, with 0 as a minimum and 1 as a maximum.

Results. Table 7 reports the SWAP score for both experiments A and B. We analyse these results, focusing on our hypotheses first. The SWAP scores in experiment A show that hypothesis $\mathcal{EH}_4$ is only confirmed for the smallest domain size ($N = 4$) while for larger domain sizes, QDG-IA generates less reconstructable descriptions than our human speakers. When domain size is small, precise (i..e, non-vague) quantifiers tend to be used (c.f., section 2). Consequently, our algorithms always generate logically complete descriptions, so they enjoy low SWAP scores (i.e., high reconstructability). Conversely, when domain size

	Model	N=4	N=9	N=20	All
Experiment A	Human	8.33	6.25	15	9.86
	QDG-IA	0	11.81	18.33	10.05
	QDG-GREEDY	2.08	8.33	15	8.47
	Model	N=6	N=10	N=16	All
Experiment B	QDG-IA	1.39	10	10.42	7.27
	QDG-GREEDY	1.39	8.33	7.81	5.84

Table 7: Average SWAP(%) for each algorithm and for human-produced descriptions. N represents the domain size.

becomes larger, then logical completeness becomes less and less achievable (cf., section 2), and a larger number of vague quantifiers are used. Consequently, the SWAP scores go up, so the reconstructability of descriptions produced by human speakers and algorithms goes down.

As for our hypothesis $\mathcal{EH}_5$, by looking at results from both experiments A and B, its null hypothesis is retained; in fact, the data go in the opposite direction, since descriptions generated by the QDG-GREEDY have better reconstructability than QDG-IA. One possible explanation is that QDG-IA may have generated less complete quantified descriptions than QDG-GREEDY (since QDG-GREEDY always looks at quantified expressions that have the highest discriminatory power). In other words, when readers were simply reading these descriptions together with paired scenes (i.e., in the human judgement study), this difference may not be have been noticed (note that QDG-IA had a similar level of informativity as QDG-GREEDY), the difference may have been "enlarged" in the reconstruction experiments.

Besides, from results in Table 7, we also found that when focusing on hard cases (i.e., Experiment A), reconstructability decreases with the increase of domain size. In contrast, when we use randomly selected scenes (i.e., Experiment B), although differences between small and large domains exist, it appears that if the domain size is large enough, then no significant difference exists (i.e., there is no significant difference between the SWAP score when $N = 9$ and when $N = 20$).

4.3 Discussion

We have reported two evaluation studies. Although our algorithms were designed to produce quantified descriptions that are logically complete, in the human judgement study, the algorithm-produced quantified descriptions did not receive a higher informativity score than human-produced ones. A similar phenomenon occurs in experiment A of the reconstruction study. The only exception is the smallest domain ($N = 4$), where algorithm-generated descriptions had better reconstructability than human-generated ones. A possible explanation is that, given a task that is as demanding as the one in our experiments, in all situations except the simplest ones, so many obstacles can get in the way of a proper understanding of the descriptions that the logical completeness that human-produced descriptions might suffer from is overwhelmed by these obstacles. For example, if a logically perfect description

is expressed by means of a syntactically ambiguous structure, then both reconstructability and correctness and informativity are bound to suffer.

Comparing the two algorithms, the greedy algorithm has slightly higher reconstructability scores than the incremental algorithm. In line with this, in Experiment B of the human judgement study, QDG-GREEDY had slightly higher informativity than QDG-IA. Yet QDG-GREEDY was not significantly judged to be more informative than QDG-IA ($t = -0.6026, p = .5487$).

5. General Discussion

In this paper, we have reported on an elicitation experiment in which human speakers describe geometrical scenes. The resulting corpus, called QTUNA, was annotated and analysed. [18] We then proposed two generation algorithms that aim to mimic the language production behaviour recorded in the corpus, understood as what is known in the computational modelling community as a *product model*, that is, a model that focuses on the relation between inputs and outputs without any claims about the manner in which this is done (see section 1 for explanation). We then evaluated these algorithms, looking at scenes of a variety of sizes, including scenes that contained a number of objects not seen in the corpus. Our evaluations suggest that our algorithms produce descriptions that are both natural (i.e., human-like) and useful.

Computational models of language use can offer a wealth of insight into the choices that human speakers and writers make when they use language. Let us take stock to see what lessons may be drawn from our computational modelling exercise.

5.1 Quantified Descriptions and Referring Expressions

Computational models of the production of *referring expressions* have been studied widely (Dale & Reiter, 1995; Dale & Viethen, 2009; van Deemter et al., 2012; Krahmer & van Deemter, 2012; van Gompel, van Deemter, Gatt, Snoeren, & Krahmer, 2019; Chen & van Deemter, 2020; Same, Chen, & Van Deemter, 2022). They aim to mimic how human speakers use referring expressions to single out a referent for a hearer. For example, given a scene such as Figure 2, a participant could say "*the large sofa*" or "*the large right-facing sofa*". Each of these expressions lets readers identify the target reference from the context.

Quantification is not reference, of course. Nonetheless, it is illuminating to compare the two phenomena and, in fact, the algorithmic approach we have chosen to model quantification resembles some algorithms originally discussed in Dale and Reiter (1995), where a referring expression is constructed by accumulating properties (e.g., `COLOUR`, `SIZE`) one by one, each of which is thought to "remove" from consideration a set of "distractor objects", that is, potential referents that differ from the target referent in one or more respects. We have emphasized this similarity by using terms familiar from referring expression generation (e.g., "target", "removing distractors", "preference order" and so on). In a nutshell,

- In the generation of both referring expressions and quantified descriptions, the task can be viewed as a step-wise addition of descriptive information that narrows down

18. On a different note, following on QTUNA, we recently conducted a similar study in Mandarin Chinese. Please see Chen (2022) and Chen and van Deemter (2022) for more details.

an initial set of possibilities (a set of possible referents in one case, and a set of scenes in the other case) to a small set – typically a singleton set.

- In both situations, the "narrowing down" metaphor gives rise to a range of possible algorithms. In each case, for instance, a "greedy" algorithm might proceed by always adding the information that most effectively reduces the size of the current set of possibilities. In other words, the notion of *discriminatory power*, which is crucial for models of reference, looms large in the modelling of quantification as well.

- In both cases, the effect of adding information must be understood in the context of the Common Ground of the speaker and hearer. When the speaker is unsure as to what the hearer knows (e.g., what the initial set of possibilities is), for example, the question can arise of whether it is practically feasible – in a reasonable time, and using a description that is not too lengthy or complicated – to reduce the initial set of possibilities to a singleton set. In the realm of reference, for example, Kutlak, van Deemter, and Mellish (2016) model a situation in which the aim of a referring expression is not to uniquely pick out one single referent. Below we will discuss similar situations in the realm of quantification.

These similarities should not close our eyes to the important differences that exist between the two tasks. Firstly, in the most often studied versions of the reference task (see e.g. the start of section 2), distractors are concrete objects, which are observed by the speaker and the hearer; in our quantification task, the distractors are a set of *possible* scenes, only one of which is observable, namely the target scene. This makes our quantification task much more abstract than most versions of the reference task. In our generation system, this is reflected by the stage in which the pre-processor computes the set $\mathcal{S}$ of all possible scenes from the properties that are given.

Secondly, in the reference task, properties (such as *red*) take the place that quantified expressions have in the quantification task. Quantified expressions are much more complicated than properties, hence the distinction between choosing a pattern $p(\cdot)$ (line 2 of Algorithm 2) and choosing a value v to fill the pattern (line 3).

Thirdly, the algorithms proposed in the present paper have had to find a way to take both the semantics and pragmatics of quantifier patterns into account. In a nutshell, semantics is about literal meaning whereas pragmatics is about other ways in which language use can convey information. That said, the distinction between semantics and pragmatics is much debated within Theoretical Linguistics, and the precise boundary between the two is notoriously difficult to draw (Levinson, 1983). The way in which this distinction works in relation to reference is relatively well understood, but the distinction has proved to be much harder to deal with in connection with quantification because if semantic information is lumped together with pragmatic information, our algorithms tend to generate descriptions that are unnecessarily unwieldy (see our explanation in section 3.3). Whether the solution embodied in our algorithms generalises to other types of pragmatic information is a question for further research.

5.2 Representing the meanings of quantifiers

Our generation algorithms embody specific assumptions concerning the meaning of each quantifier. For example, when an algorithm adds the quantified expressions *All circles are blue* to a description, we assume that "All A are B" means $[\![A]\!] \subseteq [\![B]\!] \wedge [\![A]\!] \neq \emptyset$; consequently, our algorithms remove from the set S all those scenes for which this logical conjunction does not hold true. Although we have done our best to choose representations of quantifier meaning that is consistent with both the Linguistics literature and the way in which quantifiers are used in our corpus, we cannot claim yet to have found the optimal representation in each case. For example, various authors (Moxey & Sanford, 1993; Nouwen, 2010) have pointed out that human quantifier use is guided not only by raw numbers of objects alone but by (speakers' and) hearers' expectations about the number as well; for example, a child in The Netherlands who has seen 10 animals on a given day may say she has seen *many elephants* (if that's what they were) but *a few cows* (if that's what they were). Although the geometrical scenes on which we have focused in this paper have sought to minimise these issues, there is surely a lot of progress to be made; in fact, it is perhaps remarkable that our algorithms work as well as our evaluation suggests they do.

A class of quantifiers where this disclaimer is particularly opportune are "vague" quantifiers, that is, quantifiers where there can exist borderline cases in which it is debatable whether or not the quantifier applies; cases in point are quantifiers like "many", "few", "all except a few", and so on. In all these cases, the set-up of our generation algorithms forces us to use a crisp cut-off point – deciding, for example, that *Many A are B* is true if less than 20% of A are B, and false otherwise. Although this contradicts received wisdom about the meaning of these quantifiers, our evaluation in section 4 suggests that, for the type of generation task at hand, our algorithms "get away" with this simplification. While this outcome gives rise to interesting questions – Could an NLG algorithm that models vague expressions as if they were crisp pass the Turing test? – we believe that it would be interesting to experiment with alternative assumptions that do more justice to what is known about these quantifiers.

For instance, one could represent the meaning of quantifiers probabilistically (Moxey & Sanford, 1993), or using a version of Fuzzy Logic. In both cases, the representations in question would tell us to what extent a given quantified expression is applicable in a given situation: let's call this its *degree of applicability*. Such a move could even benefit quantifiers that linguists generally consider to be crisp. For example, Degen and Tanenhaus (2011) and van Tiel (2014) pointed out that, when reading quantified expressions like *Some of A are B*, readers' acceptability is lower than 1 (though often higher than 0) if the target set is either too small or too large. A similar approach is taken in the Bayesian quantifier models of (Frank & Goodman, 2012), Franke (2014) and Qing (2014, Chapter 4), which are learned from experimental data. The resulting non-crisp meaning representations could be fed into our generation algorithms in a number of ways. For example, in the Incremental Algorithm of section 3.4, the choice of the next quantified description to be included in the description (which was done in lines 2 and 3 of Algorithm 2) could be made on the basis of the degree of applicability of the expression in combination with its preference degree. It would be interesting to see whether, as a result of this move, the quality of the resulting quantified descriptions (as measured by evaluation studies such as the one reported in section 3) will

improve. Since the present paper focuses on the production of a wide range of quantifiers rather than on sophisticated models for specific quantifiers, this exploration was left for future research.

5.3 Ecological Validity

As discussed in Section 1, previous studies on the production of quantified expressions, in both cognitive science (Yildirim et al., 2013; Herbelot & Vecchi, 2015) and computational linguistics (Barr et al., 2013)), asked speakers to pick a quantifier from a listed set of quantifiers; speakers were asked to base their choice of quantifier on a given sentence pattern, where the quantifier itself was the only thing that was left open. Our own experiments, by contrast, gave much more freedom to speakers, who could use any sentence pattern they liked (and hence any arguments for the quantifier as well), as well as any quantifier they wished to utter. Moreover, speakers were not restricted to uttering only one sentence; instead, they could produce a small discourse consisting of any number of sentences, containing any number of quantifiers. We believe that this makes our study the most ecologically valid study of its kind to date.

Our study inherits one limitation from previous studies. Like previous studies, we used artificial scenes, which showed arbitrary arrangements of abstract shapes (instead of, say, holiday snaps). We believe that this limitation can be overcome in future research that could use a similar setup based on realistic scenes (e.g. photographs of people gathering around a table); such scenes may also contain many more objects than the ones in the present study, thereby raising interesting new research questions (see more discussion in Section 5.4.3).

Our study may also possess a shortcoming that was not inherited from its predecessors but that was necessitated by our wish to study a wide range of quantifiers. A sequence of pilot experiments suggested to us that the best way to ensure this was to discourage the use of enumerations because otherwise the corpus would have been dominated by a family of highly uniform expressions, all of which are of the form n A are B (e.g., as in "There are 3 blue circles"; see Section 2 for explanation). Although this may have compromised ecological validity to some extent, we believe that this is a price worth paying for the richness of the resulting QTUNA corpus.

5.4 Open Questions and Future Work

The work on which we have reported here gives rise to a number of questions that are waiting to be explored.

5.4.1 How Efficiently Do Speakers Use Quantification?

Speakers in our corpus were frequently less than optimally "efficient" in their use of quantification, saying more than was strictly necessary for describing the scene. An extreme example is the quantified description for the scene in Figure 8 where some speakers use as many as three quantifiers (i.e., "*Half the objects are squares. Half the squares are red. Half the circles are red.*"), whereas others use only one (i.e., "*All possible combinations are shown.*") Another type of example arises when a scene of size $N = 4$ can be described by saying "*There are red circles and blue squares.*" (using two plural noun phrases), in which

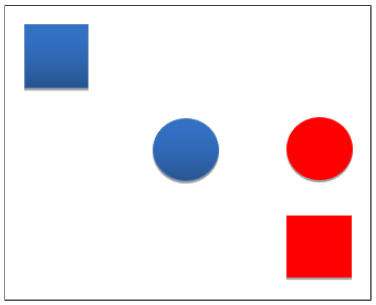


Figure 8: A scene of domain size $N = 4$ from QTUNA.

case the description "*There are two red circles and two blue squares.*" could be regarded as inefficient. Investigating the mechanisms that allow speakers to be maximally efficient – and the conditions under which these mechanisms are actually deployed – is a rich area for further research. Once again, there is an analogy here with research on the production of referring expression, where researchers have studied under what circumstances speakers tend to "over-specify" a referent. For example, when referring to the target object in Figure 2, many speakers say "*the big red sofa*" although either *red* or *sofa* is a superfluous property. This includes both experimental works (e.g., Pechmann (1989), Engelhardt, Bailey, and Ferreira (2006), Engelhardt, Demiral, and Ferreira (2011) and Koolen et al. (2011)), and computational work (e.g., Dale and Reiter (1995), van Gompel et al. (2019) and Degen, Hawkins, Graf, Kreiss, and Goodman (2020)). Perhaps the main question raised by these phenomena is whether speakers are "inefficient" because they cannot help themselves, or to help the reader understand the description (i.e., Bell (1984) and Coupland and Jaworski (2008)). Analogous questions regarding quantification have yet to be answered.

5.4.2 How to Capture Variation in the Corpus?

Substantial differences between speakers are known to exist in many other areas of language production (e.g., Horton and Keysar (1996), Holden, Van Orden, and Turvey (2009), Gibbs and Van Orden (2012), van Deemter (2016)). Such differences are likely to affect all the issues discussed in this paper. One approach would be to investigate how key properties of the descriptions vary between different types of speakers, looking at differences in level or type of education for example. A different approach would be to design a probabilistic generator, which generates all the different types of descriptions that are seen in the corpus but takes into account their frequencies. The degree of fit between such a probabilistic model and the corpus could be measured using the *generalisation criterion* methodology of Busemeyer and Wang (2000), analogous to the probabilistic modelling of reference in van Gompel et al. (2019).

5.4.3 How to Quantify Over More Challenging Types of Scenes?

The scenes on which this paper has focused are relatively simple. How does quantification work if the domain size is further increased? For example, one might expect to find that, similar to the finding of this paper, the participants would produce even more vague quantifiers, more incompleteness, and so on. Scenes could also be populated by more naturalistic

objects, standing in more naturalistic situations (e.g. a person walking a dog). Evidently, naturalistic scenes permit many more than 2 attributes, each of which will tend to have more than 2 values, and so on. Naturalistic scenes threaten to undermine one of our ideas on which our algorithm rests, namely to start computing the set of all possible scenes (i.e., constructing $\mathcal{S}$), and to work by chipping away from that set. Suppose one wants to describe the people in a football stadium, saying something like:

(13) Nearly everyone in the stadium was wearing the Liverpool colours.

It is unclear what were all the possibilities that this description is trying to rule out since it is difficult to determine all the things people might be wearing. Furthermore, it seems likely that the aim of the utterance is to state that the situation in the stadium runs counter to normal expectations – an aspect of quantification that has been noted widely in the literature (Moxey & Sanford, 1993) but was not covered by our models so far.

One possible solution is to abandon the idea of starting from the complete set of all possibilities, starting instead from a suitably sized *sample* of possible scenes, possibly gleaned from other football matches in the same stadium, proceeding as before in other ways (e.g., terminating when all distractor scenes from the sample have been ruled out). Note that this approach would be sensitive to constraints and statistical regularities that the speaker and hearer are attuned to. For instance, the sample would tend to bear out the regularity that if one's left shoe is brown then so is one's right shoe. More interestingly, a large-enough sample of scenes could go a long way towards embodying our "normal expectations" regarding the outfits that people in stadiums normally wear, including the expectation that the Liverpool colours do not normally dominate to such an extent.

Acknowledgments

The authors wish to thank Ruby Pel, Silvia Pagliaro, Louk Smalbil, and Chenghua Lin for helping set up some of the experiments. We thank Larry Moss, Jakub Szymanik, Camilo Thorne, Denis Paperno, Rick Nouwen, and Gordon Briggs for the suggestions that helped shape our work. We also thank the anonymous reviewers for their helpful comments.

References

Barr, D., van Deemter, K., & Fernández, R. (2013). Generation of quantified referring expressions: Evidence from experimental data. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pp. 157–161, Sofia, Bulgaria. Association for Computational Linguistics.

Barwise, J., & Cooper, R. (1981). Generalized quantifiers and natural language. In *Philosophy, language, and artificial intelligence*, pp. 241–301. Springer.

Bell, A. (1984). Language style as audience design. *Language in society*, *13*(2), 145–204.

Belz, A., & Gatt, A. (2007). The attribute selection for gre challenge: Overview and evaluation results. In *Proceedings of UCNLG+MT: Language Generation and Machine*

Translation, pp. 75–83, Copenhagen, Denmark. Association for Computational Linguistics.

Busemeyer, J. R., & Wang, Y.-M. (2000). Model comparisons and model selections based on generalization criterion methodology. *Journal of Mathematical Psychology*, *44*(1), 171–189.

Chen, G. (2022). *Computational Generation of Chinese Noun Phrases*. Ph.D. thesis, Utrecht University.

Chen, G., & van Deemter, K. (2020). Lessons from computational modelling of reference production in Mandarin and English. In *Proceedings of the 13th International Conference on Natural Language Generation*, pp. 263–272, Dublin, Ireland. Association for Computational Linguistics.

Chen, G., & van Deemter, K. (2022). Understanding the use of quantifiers in Mandarin. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pp. 73–80, Online only. Association for Computational Linguistics.

Chen, G., van Deemter, K., & Lin, C. (2019a). Generating quantified descriptions of abstract visual scenes. In *Proceedings of the 12th International Conference on Natural Language Generation*, pp. 529–539, Tokyo, Japan. Association for Computational Linguistics.

Chen, G., van Deemter, K., Pagliaro, S., Smalbil, L., & Lin, C. (2019b). QTUNA: A corpus for understanding how speakers use quantification. In *Proceedings of the 12th International Conference on Natural Language Generation*, pp. 124–129, Tokyo, Japan. Association for Computational Linguistics.

Coupland, N., & Jaworski, A. (2008). *Sociolinguistics: a reader and coursebook*. Palgrave.

Creaney, N. (1996). An algorithm for generating quantifiers. In *Eighth International Natural Language Generation Workshop*.

Dale, R., & Reiter, E. (1995). Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive science*, *19*(2), 233–263.

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

Degen, J., Hawkins, R. D., Graf, C., Kreiss, E., & Goodman, N. D. (2020). When redundancy is useful: A bayesian approach to "overinformative" referring expressions.. *Psychological Review*, *127*(4), 591.

Degen, J., & Tanenhaus, M. K. (2011). Making inferences: the case of scalar implicature processing. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 33.

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

Engelhardt, P. E., Bailey, K. G., & Ferreira, F. (2006). Do speakers and listeners observe the gricean maxim of quantity?. *Journal of Memory and Language*, *54*(4), 554–573.

Engelhardt, P. E., Demiral, Ş. B., & Ferreira, F. (2011). Over-specified referring expressions impair comprehension: An erp study. *Brain and cognition*, *77*(2), 304–314.

Frank, M. C., & Goodman, N. D. (2012). Predicting pragmatic reasoning in language games. *Science*, *336*(6084), 998–998.

Franke, M. (2014). Typical use of quantifiers: A probabilistic speaker model. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 36.

Gatt, A., & Krahmer, E. (2018). Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, *61*, 65–170.

Gatt, A., van der Sluis, I., & van Deemter, K. (2007). Evaluating algorithms for the generation of referring expressions using a balanced corpus. In *Proceedings of the Eleventh European Workshop on Natural Language Generation (ENLG 07)*, pp. 49–56, Saarbrücken, Germany. DFKI GmbH.

Geurts, B., & Nouwen, R. (2007). At least et al.: the semantics of scalar modifiers. *Language*, *83*, 533–559.

Gibbs, R. W., & Van Orden, G. (2012). Pragmatic choice in conversation. *Topics in Cognitive Science*, *4*(1), 7–20.

Grice, H. P. (1975). Logic and conversation. In *Speech acts*, pp. 41–58. Brill.

Herbelot, A., & Vecchi, E. M. (2015). Building a shared world: mapping distributional to model-theoretic semantic spaces. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 22–32, Lisbon, Portugal. Association for Computational Linguistics.

Holden, J. G., Van Orden, G. C., & Turvey, M. T. (2009). Dispersion of response times reveals cognitive dynamics.. *Psychological review*, *116*(2), 318.

Horton, W. S., & Keysar, B. (1996). When do speakers take into account common ground?. *Cognition*, *59*(1), 91–117.

Howcroft, D. M., Vogels, J., & Demberg, V. (2017). G-tuna: a corpus of referring expressions in german, including duration information. In *Proceedings of the 10th International Conference on Natural Language Generation*, pp. 149–153.

Kamp, H., & Reyle, U. (2013). *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*, Vol. 42. Springer Science & Business Media.

Kaufman, E. L., Lord, M. W., Reese, T. W., & Volkmann, J. (1949). The discrimination of visual number. *The American journal of psychology*, *62*(4), 498–525.

Kenney, R., & Smith, P. (1996). *Vagueness: A reader*. MIT press.

Koolen, R., Gatt, A., Goudbeek, M., & Krahmer, E. (2011). Factors causing overspecification in definite descriptions. *Journal of Pragmatics*, *43*(13), 3231–3250.

Kotek, H., Sudo, Y., & Hackl, M. (2015). Experimental investigations of ambiguity: the case of most. *Natural Language Semantics*, *23*(2), 119–156.

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

Kutlak, R., van Deemter, K., & Mellish, C. (2016). Production of referring expressions for an unknown audience: A computational model of communal common ground. *Frontiers in psychology*, *7*, 1275.

Lappin, S. (2000). An intensional parametric semantics for vague quantifiers. *Linguistics and philosophy*, *23*(6), 599–620.

Levinson, S. C. (1983). *Pragmatics*. Cambridge Textbooks in Linguistics. Cambridge University Press.

Lidz, J., Pietroski, P., Halberda, J., & Hunter, T. (2011). Interface transparency and the psychosemantics of most. *Natural Language Semantics*, *19*(3), 227–256.

Moxey, L. M., & Sanford, A. J. (1993). *Communicating quantities: A psychological perspective*. Lawrence Erlbaum Associates, Inc.

Nouwen, R. (2010). What's in a quantifier?. *The Linguistics Enterprise: From knowledge of language to knowledge in linguistics*, *150*, 235.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318. Association for Computational Linguistics.

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

Peters, S., & Westerståhl, D. (2006). *Quantifiers in language and logic*. Oxford University Press.

Pezzelle, S., & Fernández, R. (2019). Is the red square big? MALeViC: Modeling adjectives leveraging visual contexts. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2865–2876, Hong Kong, China. Association for Computational Linguistics.

Pezzelle, S., Steinert-Threlkeld, S., Bernardi, R., & Szymanik, J. (2018). Some of them can be guessed! exploring the effect of linguistic context in predicting quantifiers. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 114–119, Melbourne, Australia. Association for Computational Linguistics.

Qing, C. (2014). *Quantiative social-cognitive experimental pragmatics*. Ph.D. thesis, Universiteit van Amsterdam.

Reiter, E., & Dale, R. (2000). *Building Natural Language Generation Systems*. Studies in Natural Language Processing. Cambridge University Press.

Same, F., Chen, G., & Van Deemter, K. (2022). Non-neural models matter: a re-evaluation of neural referring expression generation systems. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5554–5567, Dublin, Ireland. Association for Computational Linguistics.

Schmuckler, M. A. (2001). What is ecological validity? a dimensional analysis. *Infancy*, *2*(4), 419–436.

Solt, S. (2016). On measurement and quantification: The case of most and more than half. *Language*, *92*(1), 65–100.

Sorensen, R. (2022). Vagueness. In Zalta, E. N., & Nodelman, U. (Eds.), *The Stanford Encyclopedia of Philosophy* (Winter 2022 edition). Metaphysics Research Lab, Stanford University.

Sorodoc, I., Lazaridou, A., Boleda, G., Herbelot, A., Pezzelle, S., & Bernardi, R. (2016). "look, some green circles!": Learning to quantify from images. In *Proceedings of the 5th Workshop on Vision and Language*, pp. 75–79, Berlin, Germany. Association for Computational Linguistics.

Sun, R. (2008). *The Cambridge handbook of computational psychology*. Cambridge University Press.

Testoni, A., Pezzelle, S., & Bernardi, R. (2019). Quantifiers in a multimodal world: Hallucinating vision with language and sound. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pp. 105–116, Minneapolis, Minnesota. Association for Computational Linguistics.

van Deemter, K. (2016). *Computational models of referring: a study in cognitive science*. MIT Press.

van Deemter, K., Gatt, A., Sluis, I. v. d., & Power, R. (2012). Generation of referring expressions: Assessing the incremental algorithm. *Cognitive science*, *36*(5), 799–836.

van Deemter, K., Sun, L., Sybesma, R., Li, X., Chen, B., & Yang, M. (2017). Investigating the content and form of referring expressions in mandarin: introducing the mtuna corpus. In *Proceedings of the 10th International Conference on Natural Language Generation*, pp. 213–217.

van Deemter, K., Theune, M., & Krahmer, E. (2005). Real versus template-based natural language generation: A false opposition?. *Computational linguistics*, *31*(1), 15–24.

van Gompel, R., van Deemter, K., Gatt, A., Snoeren, R., & Krahmer, E. (2019). Conceptualization in reference production: Probabilistic modeling and experimental testing.. *Psychological review*, *126*(3), 345.

van Tiel, B. (2014). *Quantity matters: Implicatures, typicality, and truth*. Ph.D. thesis, [Sl: sn].

Viethen, J., & Dale, R. (2008). The use of spatial relations in referring expression generation. In *Proceedings of the Fifth International Natural Language Generation Conference*, pp. 59–67. Association for Computational Linguistics.

Yildirim, I., Degen, J., Tanenhaus, M. K., & Jaeger, T. F. (2013). Linguistic variability and adaptation in quantifier meanings. In *Proceedings of the 35th Annual Meeting of the Cognitive Science Society*.

Zajenkowski, M., & Szymanik, J. (2013). Most intelligent people are accurate and some fast people are intelligent.: Intelligence, working memory, and semantic processing of quantifiers from a computational perspective. *Intelligence*, *41*(5), 456–466.