Size matters: Grounding quantifiers in spatial perception
Pauw, S.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Chapter 5

Clustering Quantifiers and Perceptual Deviation

This chapter studies how quantificational expressions such as few, three and all can be grounded in real-world perception. Based on findings from psycholinguistics, we propose a computational model designed for use in robot-robot interaction scenarios which involve discrimination tasks for objects in the real world. We test the performance of our model and contrast it with a type theory based model. We show that our design choices make our model more suitable for real-world applications. This chapter has previously been published as Pauw, S. and Spranger, M. (2012). Embodied quantifiers. In Lassiter, D. and Slavkovik, M., editors, New Directions in Logic, Language and Computation, pages 52–66. Springer

5.1 Introduction

The experiments reported in this chapter are part of a greater research effort that studies human language-like communication using (artificial) robotic agents (Steels, 2012b). Central to these studies is the question: How can the meaning of language be grounded in real-world perception? Answers to this problem are given for different aspects of human language such as color (Bleys et al., 2009), space (Spranger, 2012), temporal language (Gerasymova and Spranger, 2012b) and action language (Steels et al., 2012b; Steels and Spranger, 2012; Spranger and Loetzsch, 2009). All of these models operationalize basic insights from prototype theory (Rosch et al., 2004) about how people conceptualize objects and relations between them and propose a degree-based semantics. In this chapter we describe a fully operational model for natural language quantifiers such as many, all and three that builds further on these findings. The model, termed clustering quantification (Spranger and Pauw, 2012), employs a combination of prototype theory and standard clustering algorithms and has been successfully used to study the acquisition and evolution of quantificational terms (Pauw and
Inspired by existing psycholinguistic research on quantification (Hormann, 1983; Newstead and Coventry, 2000; Coventry et al., 2010), the model presented in this chapter focuses on the role of quantifiers in determining a referent of a quantified noun phrase. The quantificational information of a noun phrase imposes constraints on the cardinality of its possible referents. For example, the quantifier *three* in the utterance “the three blocks” signals that the extension of *blocks* in the context contains three elements.

We test the adequacy of the clustering quantification model for real-world perception through a series of experiments. In these experiments we contrast the performance of our model with a model that is more in line with type theoretic accounts of quantification which assume that nouns can be modeled as predicates. Accounts falling into this class are Generalized Quantifier Theory (GQ) (Barwise and Cooper, 1981) and Fuzzy Quantifier Theory (Zadeh, 1965, 1983). This chapter proceeds by introducing the embodied interaction paradigm. The section to follow introduces our model for quantification. After that, we introduce the type theory based model. Finally, we compare both models and show that the clustering quantification performs significantly better.

### 5.2 Embodied Interaction

The model presented in this chapter is designed for use in real-world situated interaction. Figure 5.1 shows an example scene with two Sony humanoid robots (Fujita et al., 2003) interacting in a shared environment. Each robot perceives the world through its own onboard sensors, e.g., the camera and proprioceptive sensors. The vision system (Spranger et al., 2012a) gathers information from the sensors into a *world model*, that reflects the current belief of a robot about the state of the environment. One of the robots is randomly chosen as the speaker and he will choose a referent, which can be any object or subsets of objects in the environment. The goal of the speaker is to draw the attention of the interlocutor to the referent and make him point to it or to each of the objects that are the referent.

We call these interactions *language games* (Steels, 2001). The type of language game the agents play depends on the particular research question. For this chapter, the two interacting agents use the following script:

1. Both agents establish a joint attentional frame (Tomasello, 2003) and a world model using their visual and proprioceptive sensors.

2. The speaker chooses an object or a set of objects as referent *R*. He conceptualizes a meaning for discriminating *R* and tries to verbalize his conceptualization into a string of words.
5.2. Embodied Interaction

Figure 5.1: (a): Example scene consisting of various objects, e.g., robots, blocks and boxes. (b) and (c): Top-down view of the world models as perceived by respectively the speaker and the hearer.

3. This utterance is passed to the hearer.

4. The hearer parses and interprets the utterance and tries to find the object or the set of objects he thinks the speaker is trying to discriminate.

5. The hearer points to the object or the set of objects.

6. The speaker checks whether the objects pointed to by the hearer were indeed the ones he had in mind.

This language game can have two different outcomes. If the hearer points to the correct set of objects the game is a success. Otherwise it is a failure.

Such interactions require a mapping of continuous perceptual data to discrete symbols (language). To this end, we use the computational semantics systems Incremental Recruitment Language (IRL) (Spranger et al., 2010a, 2012b). Since
the communicative goal is to identify some referent, the semantics of a particular phrase is modeled in IRL as a series of operations, i.e. a program, that the hearer has to go through in order to single out the objects that are the referent.

![Semantic structure representing the meaning of the utterance the left block](image)

Figure 5.2: Semantic structure representing the meaning of the utterance the left block. The network contains bind-statements that introduce semantic entities (e.g., the object class block), as well as operations that define what to do with these semantic entities. Links in the network are defined by variables (starting with a ?), e.g., the output of operation apply-class is linked to apply-spatial-category through the variable ?set-21.

Figure 5.2 shows the IRL-program underlying the utterance the left block. The program is represented as a network containing semantic entities (e.g., block and left) and cognitive operations (e.g., apply-class). The semantics entities represent concepts and categories. The cognitive operations instruct the agents what to do with the semantic entities. For example, apply-class takes the concept block and applies this to the objects in the context. The result of this application is fed to the operation apply-spatial-category, which processes the data using the spatial category left, and finally, apply-selector computes the referent using the selector unique for more information.

IRL provides a general framework for the automatic interpretation and composition of such programs, but the concrete implementation of each operation, e.g. apply-selector is outside of IRL. IRL makes no assumptions about the inner workings of these operations. Consequently, IRL is an ideal formalism for studying different models of categorization and quantification.

### 5.3 Clustering Quantification

There is been substantial research in the past 10 years on grounding basic categories and relations in real world perception. We build upon an existing system for spatial language which has been proposed for the grounding of spatial categories and quantifiers such as “the” (Spranger, 2011) and has been shown to
be very successful in real world interactions (Spranger and Pauw, 2012). This system is based on two psycholinguistic processing principles.

**acceptability** (Herskovits, 1986), also called prototypicality (Rosch and Lloyd, 1978; Lakoff, 1987), means that categories such as left apply to a certain degree. An object can be more or less to the left of a landmark.

**contrast** requires speakers which are trying to discriminate objects to choose the relation or category which maximizes acceptability of the object and minimizes acceptability of all other objects (Tenbrink, 2005). The phrase “the left block” refers to the leftmost block in a scene.

Starting from this model the main question is how notions of acceptability and contrast can be extended to quantifiers which might introduce additional constraints such as cardinality (e.g. “three”). In this section we propose to operationalize these ideas for quantifiers using mechanisms from machine learning and clustering. Before we jump to the quantifier model we briefly outline how prototype-based processing is implemented for categories.

### 5.3.1 Acceptability

The acceptability of a concept for an object depends on a similarity function that assigns a score to the combination of concept and object. For example a spatial relation such as *left* is represented by a prototypical vector in euclidean space. The degree to which an object is left depends on the angle between the object and the prototypical vector for *left*. The similarity function maps this angle difference to a score between 0 and 1. The following gives an example of a parameterized similarity function used for modeling projective categories such as *front* and *left*.

\[
s = e^{-\frac{0.5d(p,o)}{\sigma_p}}
\]

With \(s\) being the resulting similarity score, \(d(p,o)\) a distance function between prototype \(p\) and object \(o\), and \(\sigma_p\) a parameter that determines the rate at which the distance influences the similarity. In the case of projective categories such as *front* and *left*, this distance function computes the difference in angle between object and prototype. Similarly, functions for other concepts such as *block* are constructed, except that the similarity/distance space might be the set of features of an object.

For semantic structure such as the one in Figure 5.2 this means that operations that apply categories such as **apply-spatial-category** and object classes such as **apply-class** assign scores to the objects in the context. Scores for the spatial category and the object class are multiplied so that in the end a single acceptability rating in the form of a similarity score is computed for each object in the context. In short, this is a model for spatial language which establishes
the acceptability of a noun phrase such as “block in front of me” for every object in a given context (See Figure 5.3).

5.3.2 Contrast

But, how do agents use these scores to distill concrete referents? This is where the notion of contrast comes into play. For instance, if an agent wants to discriminate an object from the context then he is likely to try and maximize the applicability contrast (the difference in similarity scores) between the object he wants to discriminate and all other objects in the context. Hearers choose the object that best fits the description. Speakers choose the categories that maximize the contrast.

Similarly, if the referent is a set of objects (as for the utterance three blocks) we require a procedure that decides for every entity whether it is part of the referent set or not. Quantificational information constrains this process. For example, the quantifier three signals that the referent set contains (at least) three element, and the quantifier many signals that the referent set contains more elements than a certain norm. To operationalize these ideas we use standard clustering algorithms from machine learning. In particular, we apply variants of agglomerative clustering (Mitchell, 1997) and k-means (Lloyd, 1982; Manning et al., 2008).

The algorithms are used to implement the operation of apply-selector. The task of this operation is to decide for every entity in the context if it is part of the referent or not, based on the scores that were assigned by the previous operations. In essence it has to divide the input set into two sets of objects,
5.3. Clustering Quantification

Figure 5.4: This figure shows the results of applying agglomerative clustering. The algorithm finds two possible referents for the utterance “block(s) in front” (cluster-1 and cluster-2). The quantificational information of the noun phrase can be used to further constrain the possible referents of the noun. The noun phrase “the block in front of me” signals that there is one unique referent, making cluster-1 the most likely referent. For the noun phrase “all blocks in front of me”, the most likely referent is cluster-2.

The objects that are part of the referent (REFSET) and the objects that are not (COMPSET). Finding such a partitioning is precisely what clustering algorithms are designed for. However, there are many ways to partition an input set. The particular partition that is chosen depends on a combination of factors. First of all, the clustering algorithms use heuristics to find good partitioning. Good partitionings are those that maximize inter-cluster variance and minimize intra-cluster variance. The first is a measure of how far clusters are apart (contrast). The second is a measure how much cohesion there is in each cluster. Both k-means and agglomerative clustering are algorithms that optimize for these two indicators and we apply them here to similarity scores computed for the spatial relation and the object classes. Figure 5.4 shows an example result of the clustering algorithm. For the present experiment, the precise details of this clustering are not relevant. With different parameter settings different clusters could have been computed. The only constraint is that all agents use the same clustering algorithm.

Another factor that is taken into account in determining a good partitioning is the quantificational information. The quantificational information provides information on the cardinality of the REFSET. Consider for example Figure 5.4. For the utterance “all blocks in front of me”, the plural marker and the quantifier all enforce that the REFSET should at least contain two (and preferably more) elements. The REFSET for the utterance “the block in front of me” should contain precisely one element. Thus depending on the quantificational information,
Chapter 5. Clustering Quantifiers

Figure 5.5: Plot of the similarity functions of many and few. The prototypical value for few and many are 1 and 6 respectively. They intersect at 3.5, meaning that for any cardinality above 3.5 the quantifier many is more acceptable, and for any cardinality under 3.5 the quantifier few is more acceptable. In practice, the similarity function is modulated by a context parameter which shifts the exact point where few becomes more acceptable than many. For the purpose of this chapter, the parameter is left fixed.

The result is a flexible algorithm which allows agents to choose a partitioning of data based on whatever quantifying criteria they want to convey. Conversely, it allows them to find the best interpretation upon hearing a quantified noun phrase. The precise nature of the constraints depends on the type of quantifier. The examples above show how this works for crisp quantifiers such as all or three. It is also possible to define constraints for gradable quantifiers such as many and few. In this case, the quantifiers are not binarily constraining the cardinality of the referent, but rather, they assign a score to every possible partitioning and work as a heuristic value in the same way as the inter- and intra-cluster variance. For example in Figure 5.4 the REFSET of the utterance “many blocks in front” refers to cluster-2, not because the many excludes cluster-1 entirely, but because the constraint imposed by many assigns a higher score to a set of cardinality 6 than a set of cardinality 1.

For the sake of simplicity we implemented quantifiers in the same way as the spatial prototypes, using a prototypical value and a similarity function. The distance function in this case is defined as the difference between the cardinality of the REFSET and some prototypical cardinality \(d(c_p, c_{REF}) = |c_p - c_{REF}|\). For the current experiment the average cardinality of the REFSET is around 3.5. For the purpose of the present experiment, we have chosen the prototypical values for few \(c_p = 1\) and many \(c_p = 6\) such that any cardinality above 3.5 is will be more similar to many and anything under 3.5 will be more similar to few.\(^1\)

\(^1\)The prototypical value of 1 may seem somewhat unnatural, we have choses this value only for the sake of this experiment. With a higher prototypical value few would be used much more to describe groups of objects than many. This asymmetry, would make it harder to interpret the results.
5.4. Generalized Quantifiers

Figure 5.5 shows a plot of the similarity functions of few and many.

In sum, this approach regards quantifiers as constraints. They help with identifying the referent of a noun phrase. We use existing clustering methods, that model the reification of the referent as a partitioning process that is partly steered by pragmatic heuristics (e.g., inter- and intra-cluster variance) and partly by semantic heuristics (quantificational expressions).

5.4 Generalized Quantifiers

To measure the performance of our approach we compare it with an implementation of a common type theoretic way of dealing with quantification (Montague, 1974). These approaches commonly consider the noun (e.g., ball) as a predicate that together with a quantifying expression forms a (quantified) noun phrase (e.g. all balls). Such a noun phrase is modeled as a generalized quantifier (GQ) (Barwise and Cooper, 1981).

For those not familiar with generalized quantifiers, we provide a brief explanation: A noun or verb phrase denotes a property that can be represented as a function from entities to truth values, in other words, as the set of entities for which the property holds. Consequently, the interpretation of ball is the set of all balls B in a context, and are red is the set of all things that are red R. Quantifying expressions then are understood as set relations. For instance, the sentence all balls are red can be modeled as $B \subseteq R$. The determined noun phrase is therefore modeled as a function from a set to truth values, in other words, a generalized quantifier. For example, the determined noun phrase all balls is interpreted as a function $f(Q)$ that is true iff $B \subseteq Q$. The functional role of the quantifier under this analysis is to transform the noun predicate into such a generalized quantifier. For example, the meaning of the quantifier all is a function $g(P, Q)$ that is true iff $P \subseteq Q$. Where $P$ is the predicate of the noun and $Q$ the predicate of the verb-phrase.

The essential restriction imposed by this approach is the fact that the noun is considered to be a predicate. In light of the previous model, this means that before applying the quantifier, we require some procedure that turns the set of scored items into a predicate (i.e., a procedure that decides for every element if it is part of the noun or not).

In accordance with this observation, we implement a model we will henceforth refer to as the Generalized Quantifier approach. The main difference between the Generalized Quantifier approach and our model as described earlier in this chapter lies in the operation apply-selector (as seen in Figure 5.2). Before applying the quantificational information, the operation establishes the set of objects that forms the extension of the noun. Just as in the previous model, the interpretation of the noun “block(s) in front of me” establishes a similarity score for every object in the context. The operation apply-selector determines for every object if its
score is high enough to be part of the noun. In order to make a fair comparison between the two models, we employ the exact same clustering methods for the reification of the noun as above. The main (and essential) difference with the previous model is that the reification is done before considering the quantificational information. This difference might seem insignificant, but it is needed to stay in line with the type theoretic approaches and, as we are about to show, has a very important impact on the performance of the model.

Generalized Quantifiers is fundamentally a theoretical proposal which does not propose any specific form of implementation. Therefore, one might be tempted to question how general our modeling is. For the concrete reification operation, other mechanisms are possible, but the main point from this section is that no matter what the particular implementation is, type theoretic approaches rely on the assumption that we can unambiguously establish the cardinality of the interpretation of the noun without regarding the quantificational information. The fact that there is no easy fix for this problem can be seen in the model of Fuzzy Quantifiers. This model is an fuzzy extension of GQ. Here, all set relations and predicates can be degree-based, but nonetheless, the model requires a mechanism to establish the cardinality of the fuzzy set representing the noun. This crisp intermezzo in the analysis of quantified noun phrases is needed to save the GQ representation of nouns, but makes Fuzzy Quantifiers prone to the same problem as GQ.

In sum, although type theoretic approaches do not propose a concrete operationalization, they do impose particular constraints on the way the referent is determined. The reliance on a defined cardinality for the extension of the noun is incompatible with our model as proposed in the previous section. And, as we will see in the next section, it is precisely this reliance on a defined cardinality that makes the Generalized Quantifier approach a much less suitable model for real-world application.

### 5.5 Experimental Setup and Results

Since we have operational models of the two approaches we can compare their performance in real world interactions. A population of agents play thousands of language games. Each agent is equipped with English spatial categories such as front, back, left, right, near and far and with the quantifiers many, few and the cardinals one to twelve. We consider two different populations of agents: in

---

1) Syntactic processing is implemented in Fluid Construction Grammar (FCG) (Steels and De Beule, 2006). FCG maps IRL-programs to natural phrases and back given a particular lexicon and grammar. Here we implemented lexical and grammatical. Here, we equipped agents with lexical items for spatial categories (e.g., left, back, front, right), object classes (e.g., block, box, robot, thing) (Spranger, 2012) and quantifiers (e.g., many, few, one, nine). Moreover, rules for quantified adjective noun phrases, quantified noun phrases, and quantified noun phrases like “three blocks left of the box” are provided (Spranger and Steels, 2012).

2) The scenes contain...
5.5. Experimental Setup and Results

Figure 5.6: Average communicative success for 5000 interactions on different sets of spatial scenes (grouped according to number of objects in each scene).

Figure 5.7: Average communicative success for populations with vague quantifiers such as many and few.
Chapter 5. Clustering Quantifiers

Figure 5.8: Average communicative success for cardinal only populations.

Figure 5.9: Average communicative success for populations with all quantifiers. Here both agents perceive the scene through the same camera. I.e., both agents have identical world models.

one population all agents use our model; the other population uses the generalized quantifier model. Each interaction is either successful (the hearer points up to only ten objects. So the cardinal numbers one to twelve are enough any group of objects, even if the agent’s perception is off by two objects.
to the correct set of objects) or unsuccessful (any of the steps in the language game script fails). The performance of the respective models is reflected by the average communicative success (the percentage of successful interactions) over all language games.

Scene Complexity

Figure 5.6 shows how the two approaches perform. We test performance in different scenes. Some contain only two objects, others up to ten. The results show two important points: 1) the clustering approach performs much better than the generalized quantifier approach in all experimental conditions; and 2) increasing the complexity of the scene the difference in performance grows.

Already in the first condition, where there are only two objects in each scene, our approach reaches ca. 95% success whereas the GQ-based model reaches only ca. 85%. More strikingly though when the number of objects increases this difference grows even more. In the condition where ten objects success of GQ drops to below 50% – only every second interaction is a success. Our approach reaches 80% success even in difficult conditions.

Cardinal vs Vague

To discern the exact performance of cardinal and vague quantifiers we tested each of them separately. Figure 5.7 shows the result for the quantifiers few and many. The results show a worse performance overall for our and the GQ model when agents can not use cardinals. Also, the difference between the two models is smaller. Only for four objects or more, the clustering approach is performing better than GQ. And, only for 9 or more objects the difference starts to be more than 10%. This contrasts with populations in which agents can only use cardinal quantifiers (see Figure 5.8). The overall average communicative success is much higher than vague-only and slightly lower than with all quantifiers. but essentially the same result is obtained. This means that cardinals are responsible for most of the communicative success when agents are given also vague quantifiers. The reason cardinals perform better than just vague quantifiers is that they communicate hard constraints. If the speaker signals he is talking about three objects this is a very clear constraint on the referent set much more so than signalling few or many.

Perceptual Deviation

To understand why our approach performs consistently better than GQ we have to consider another condition. Figure 5.9 shows a case where the two agents interacting in a language game are perceiving the scene through the same robot body. (This manipulation is possible because software agents can access the same hardware.) In this case both approaches perform perfectly.
In embodied interactions, each agent perceives the scene through his own body. Often two agents estimate properties of the world differently. For instance, the agents in Figure 5.1 each estimate the distance and angle of objects in the scene differently. For instance, for the speaker object-1 lies at a distance of 31.6cm and an angle of −107 degrees from the box. For the hearer the same object is 22.4cm away from the box at an angle of 81 degrees. We call this the problem of perceptual deviation (see the previous chapter or Spranger and Pauw, 2012, for more details). This problem is one of the defining characteristics of interactions in the real world. Importantly, these differences in perception affect the performance of different semantic modeling approaches.

5.6 Conclusion

In this chapter, we have proposed a model for the processing of quantifiers that is intended for use in real world situations. We have extended contrast and acceptability principles known from the psycholinguistics of spatial language and showed how they can be incorporated into a semantics of quantifiers that further adds cardinality constraints. We contrasted our model with a type theory based model and showed that our model 1) is more robust against the effects of perceptual deviation, and 2) scales better with respect to the complexity of scenes.

Cardinality is not the only constraint important for understanding quantifiers. Model theoretic accounts, for instance, strongly focus on the role of quantifiers for inference – a tradition that dates back as far as Aristotle’s syllogisms – by considering quantifiers as a functional relation between noun and verb phrase. Our model does not deal with these aspects of quantification. It does however provide an important first step in grounding quantified noun phrases.

An important next step could be to investigate how our model of quantification holds up when used in full sentences. This would undoubtedly raise issues of reasoning and scope resolution (Kurtzman and MacDonald, 1993) and eventually its role in more complex discourse situations (Kamp, 1981b; van Eijck, 1990). In principle it is fairly straightforward to extend the current model to be used with entire sentences. IRL-networks can easily be extended to verify if a specific property (such as “is red” or “rolls”) holds for (a subset of) the referent set. This way IRL-networks can be used to assign truth-values to sentences.

When it comes to inference and scope resolution, a grounded semantics approach like ours can provide important advantages. While on the one hand, grounding introduces complications, such as the problem of perceptual deviation (Spranger and Pauw, 2012), grounding does allow to resolve ambiguities by verifying different possible interpretations in the context (Spranger and Loetzsch, 2011). This reduces the need for syntactic resolution. When it comes to dealing with these concerns, the most obvious vantage point is to look at results in
Discourse Representation Theory (DRT) (Kamp, 1981b; van Eijck, 1990). All semantic entities that IRL introduces, become available as free variables throughout the entire discourse. So any referent that is being introduced can freely be used for reference later in the discourse — a treatment of referents that is quite similar to DRT.

Of course, an in depth analysis of quantification in all its complexity is well beyond the scope of this chapter. In spite of these reservations, the model does what it was designed for: It models the semantics of natural language quantifiers as expressions of quantity, grounded in real-world perception.

Acknowledgements

We thank Masahiro Fujita, Hideki Shimomura and their team at Sony Corporation Japan for their help with the robotic setups used in the experiments. Funding for the research reported in this chapter was provided by Sony CSL Paris and the EU projects ECAgents (FP6) and ALEAR (FP7).