Quantity, Contrast, and Convention in Cross-Situated Language Comprehension

Ian Perera and **James F. Allen**
University of Rochester, Department of Computer Science, Rochester, NY 14627 USA
Institute for Human and Machine Cognition, Pensacola, FL 32502 USA
{iperera,jallen}@ihmc.us

Abstract

Typically, visually-grounded language learning systems only accept feature data about objects in the environment that are explicitly mentioned, whether through annotation labels or direct reference through natural language. We show that when objects are described ambiguously using natural language, a system can use a combination of the pragmatic principles of Contrast and Conventionality, and multiple-instance learning to learn from ambiguous examples in an online fashion. Applying child language learning strategies to visual learning enables more effective learning in real-time environments, which can lead to enhanced teaching interactions with robots or grounded systems in multi-object environments.

1 Introduction

As opposed to the serial nature of labeled data presented to a machine learning classifier, children and robots "in the wild" must learn object names and attributes like color, size, and shape while being surrounded by a number of stimuli and possible referents. When a child hears "the red ball", they must first identify the object mentioned, then use existing knowledge to identify that "red" and "ball" are distinct concepts, and over time, learn that objects called "red" share some similarity in color while objects called "ball" share some similarity in shape. Learning for them therefore requires both identification and establishing joint attention with the speaker before assigning a label to an object, while also applying other language learning strategies to narrow down the search space of possible referents, as illustrated by Quine's "gavagai" problem (1964).

Trying to learn attributes and objects without non-linguistic cues such as pointing and gaze might seem an insurmountable challenge. Yet a child experiences many such situations and can nevertheless learn grounded concepts over time. Fortunately, adult speakers tend to understand the limitation of these cues in certain situations and adjust their speech in accordance to Grice's Maxim of Quantity when referring to objects : be only as informative as necessary (Grice, 1975). We therefore treat the language describing a particular object in a scene as an expression of an iterative process, where the speaker is attempting to guide the listener towards the referent in a way that avoids both ambiguity and unnecessary verbosity.

Language learners additionally make use of the pragmatic assumptions of Conventionality, that speakers agree upon the meaning of a word, and Contrast, that different words have different meanings (Clark, 2009). The extension of these principles to grounded language learning yields the assumptions that the referents picked out by a referring expression will have some similarity (perceptual in our domain), and will be dissimilar compared to objects not included in the reference. Children will eventually generalize learned concepts or accept synonyms in a way that violates these principles (Baldwin, 1992), but these assumptions aid in the initial acquisition of concepts. In our system, we manifest these principles using distance metrics and thereby allow significant flexibility in the implementation of object and attribute representations while allowing a classifier to aid in reference resolution.

When faced with unresolvable ambiguity in determining the correct referent, past, ambiguous experiences can be called upon to resolve ambiguity in the current situation in a strategy called Cross-Situational Learning (XSL). There is some debate over whether people use XSL, as it requires considerable memory and computational

Proceedings of the 19th Conference on Computational Language Learning, pages 226–236,
Beijing, China, July 30-31, 2015. ©2015 Association for Computational Linguistics

load (Trueswell et al., 2013). However, other experiments show evidence for XSL in adults and children in certain situations (Smith and Yu, 2008; Smith et al., 2011). We believe these instances that show evidence of XSL certainly merit an implementation both for better understanding language learning and for advancing grounded language learning in the realm of robotics where such limitations do not exist. We show that by reasoning over multiple ambiguous learning instances and constraining possibilities with pragmatic inferences, a system can quickly learn attributes and names of objects without a single unambiguous training example.

Our overarching research goal is to learn compositional models of grounded attributes towards describing an object in a scene, rather than just identifying it. That is, we do not only learn to recognize instances of objects, but also learn attributes constrained to feature spaces that will be compatible with contextual modifiers such as *dark/light* in terms of color, or *small/large* in terms of size and object classification. Therefore, we approach the static, visual aspects of the symbol grounding problem with an eye towards ensuring that our grounded representations of attributes can be composed in the same way that their semantic analogues can. We continue our previous work (Perera and Allen, 2013) with two evaluations to demonstrate the effectiveness of applying the principles of Quantity, Contrast, and Conventionality, as well as incorporating quantifier constraints, negative information, and classification in the training step. Our first evaluation is reference resolution to determine how well the system identifies the correct objects to attend to, and our second is description generation to determine how well the system uses those training examples to understand attributes and object classes.

2 Related Work

Our algorithm for reference resolution and XSL fits into our previous work on a situated language learning system for grounding linguistic symbols in perception. The integration of language in a multi-modal task is a burgeoning area of research, with the grounded data being any of a range of possible situations, from objects on a table (Matuszek et al., 2012) to wetlab experiments (Naim et al., 2014). Our end goal of using natural language to learn from visual scenes is similar to work by Krishnamurthy and Kollar (2013) and Yu and Siskind (2013), and our emphasis on attributes is related to work by Farhadi et al. (2009). However, our focus is on learning from situations that a child would be exposed to, without using annotated data, and to test implementations of child language learning strategies in a computational system.

We use a tutor-directed approach to training our system where the speaker presents objects to the system and describes them, as in work by Skocaj et al. (2011). The focus of this work is in evaluating referring expressions as in work by Mohan et al. (2013), although without any dialogue for disambiguation. Kollar et al. (2013) also incorporate quantifier and pragmatic constraints on reference resolution in a setting similar to ours. In this work, we undertake a more detailed analysis of the effects of different pragmatic constraints on system performance.

The task of training a classifier from "bags" of instances with a label applying to only some of the instances contained within is referred to as Multiple-Instance Learning (MIL) (Dietterich, 1997), and is the machine-learning analogue of cross-situational learning. There is a wide range of methods used in MIL and a number of different assumptions that can be made to fit the task at hand (Foulds and Frank, 2010). Online MIL methods so far have been used for object tracking (Li et al., 2010), and Dindo and Zambuto (2010) apply MIL to grounded language learning, but we are not aware of any research that investigates the application of online MIL to studying cognitive models of incremental grounded language learning. In addition, we find that we must relax many assumptions used in MIL to handle natural language references, such as the 1-of-N assumption used by Dindo and Zambuto. The lack of appropriate algorithms for handling this task motivates our development of a novel algorithm for language learning situations.

3 Experimental Design

3.1 Learning Environment and Data Collection

Our environment in this experiment consists of a table with blocks of nine different shapes and two toy cars, with the objects spanning four colors. A person stands behind the table, places a randomly chosen group of objects in a designated demonstration area on the table as shown in Figure 1,

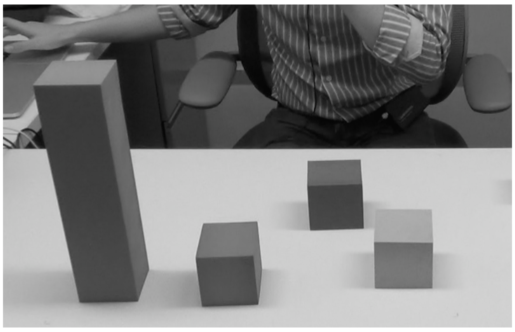

Figure 1: One of the training examples, described as “The two red blocks [left-most two blocks in this figure] are next to the other blocks.”

and describes one or more of the objects directly while possibly mentioning some relation to the surrounding objects. The goal of this setup is to facilitate object descriptions that more closely approximate child-directed speech, compared to the language in captioned images. Audio is recorded and transcribed by hand with timestamps at the utterance level, but there are no other annotations beyond timestamps. We use these intervals to match the spoken descriptions to the video data, which is recorded using the Microsoft Kinect to obtain RGB + Depth information.

3.2 Training and Test Instances

All training instances involved multiple objects, with an average of 2.8 objects per demonstration. The subject could select any set of objects to describe (often with respect to the other objects). References to objects varied in detail, from “the cube” to “a tall yellow rectangle”. Since a set of objects might have different shapes, the most common descriptor was “block”. The majority (80%) of the quantifiers were definite or numeric, and 85% of the demonstrations referred to a single object. Test instances consisted solely of single objects presented one at a time. 20% of the objects used as test instances appeared in training because of the limited set of objects available, yet the objects were placed in slightly different orientations and at different locations, deforming the shape contour due to perspective.

3.3 Prior System Knowledge

We encode some existing linguistic and perceptual knowledge into the system to aid in learning from unconstrained object descriptions. The representative feature, defined as the system’s feature space assigned to a property name (*e.g.*, color for “white”, or shape for “round”), was prechosen for the task’s vocabulary to reduce the number of factors affecting the evaluation of the system. In previous work, we showed that the accuracy of the system’s automatic choice of representative features can reach 78% after about 50 demonstrations of objects presented one at a time (Perera and Allen, 2013). In addition, we developed an extension to a semantic parser that distinguishes between attributes and object names using syntactic constructions.

3.4 Language Processing

The transcribed utterances are passed through the TRIPS parser (Allen et al., 2008) for simultaneous lexicon learning and recognition of object descriptions. The parser outputs generalized quantifiers and numeric constraints (capturing singular/-plural instances, as well as specific numbers) in referring expressions, which are used for applying quantifier constraints to the possibilities of the referent object or group of objects. The parser’s ability to distinguish between attributes and objects through syntax greatly increases learning performance, as demonstrated in our previous work (Perera and Allen, 2013). We extract the speech act (for detecting when an utterance is demonstrating a new object or adding additional information to a known object) and the referring expression from the TRIPS semantic output. Figure 2 shows the format of such a referring expression.

```
(MENTIONED :ID ONT::V11915
   :TERMS
     ((TERM ONT::V11915 :CLASS (:*
          ONT::REFERENTIAL-SEM W::BLOCK)
             :PROPERTIES ((:*
                 ONT::MODIFIER W::YELLOW))
             :QUAN ONT::THE)))
```

Figure 2: Primary referring expression extraction from the semantic parse for ”The yellow block is next to the others”.

Although there may be many objects or groups of objects mentioned, we only store the properties of the reference that is the subject of the sentence. For example, in, “Some blue cars are next to the yellow ones”, we will extract that there exists at least two blue cars. Because it is an indefinite reference, we cannot draw any further inference about whether the reference set includes all examples of blue cars.

3.5 Feature Extraction

To extract features, we first perform object segmentation using Kinect depth information, which provides a pixel-level contour around each of the objects in the scene. Then for each object, we record its dimensions and location, extract visual features corresponding to color, shape, size, color variance, and texture. No sophisticated tracking algorithm is needed as the objects are stationary on the table. Color is represented in LAB space for perceptual similarity to humans using Euclidean distance, shape is captured using scale- and rotation-invariant 25-dimensional Zernike moments (Khotanzad and Hong, 1990), and texture is captured using 13-dimensional Haralick features (Haralick et al., 1973).

3.6 Classification and Distance Measures

To determine the similarity of new properties and objects to the system's previous knowledge of such descriptors, we use a k-Nearest Neighbor classifier (k-NN) with Mahalanobis distance metric (Mahalanobis, 1936), distance weighting, and class weighting using the method described in Brown and Koplowitz (1979).

Our k-NN implementation allows negative examples so as to incorporate information that we infer about unmentioned objects. We do not train the system with any explicit negative information (*i.e.*, we have no training examples described as "This is not a red block.", but if the system is confident that an object is not red, it can mark a training example as such). A negative example contributes a weight to the voting equal and opposite to what its weight would have been if it were a positive example of that class.

The Mahalanobis distance provides a way to incorporate a k-nearest neighbor classifier into a probabilistic framework. Because the squared Mahalanobis distance is equal to the number of standard deviations from the mean of the data assuming a normal distribution (Rencher, 2003), we can convert the Mahalanobis distance to a probability measure to be used in probabilistic reasoning.

4 The Reference Lattice

To learn from underspecified training examples, we must resolve the referring expression and assign the properties and object name in the expression to the correct referents. To incorporate existing perceptual knowledge, semi-supervised methods, and pragmatic constraints in the reference resolution task, we use a probabilistic lattice structure that we call the *reference lattice*.

The reference lattice consists of nodes corresponding to possible partitions of the scene for each descriptor (either property or object name). There is one column of nodes for each descriptor, with the object name as the final column. Edges signify the set-intersection of the connected nodes along a path.

Paths through the lattice correspond to a successive application of these set-intersections, ultimately resulting in a set of objects corresponding to the hypothesized referent group. In this way, paths represent a series of steps in referring expression generation where the speaker provides salient attributes sequentially to eventually make the referent set clear.

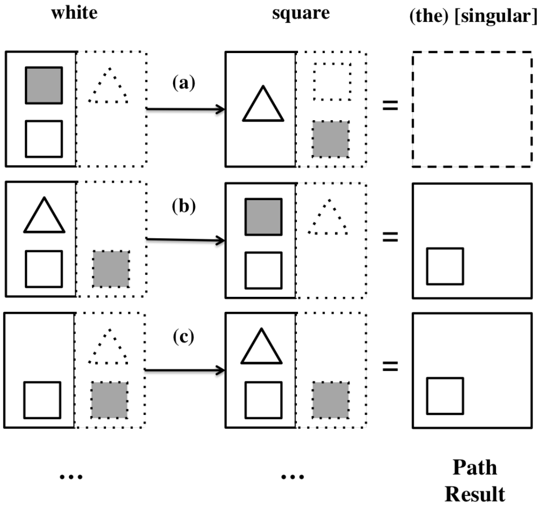

Figure 3: Three examples of paths in the reference lattice for the referring expression "the white square", when the visible objects are a grey square, white square, and a white triangle.

4.1 Lattice Generation

For each descriptor, we generate a node for every possible partition of the scene into positive and negative examples of that descriptor. For example, if the descriptor is "red", each node is a hypothesized split that attempts to put red objects in the positive set and non-red objects in the negative set. For each column there are $2^n - 1$ nodes, where n is the number of objects in the scene (the empty set is not included, as it would lead to an empty reference set). We then generate lattice edges between every pair of partitions in adjacent columns.

We can discard a large proportion of these edges, as many will correspond to the intersection of disjoint partitions and will therefore be empty. Finally, we generate all possible paths through the lattice, and, if using quantifier constraints, discard any paths with a final output referent set that does not agree with the number constraints on the mentioned referent group.

The structure of the lattice is shown in Figure 3. In this figure, partitions are represented by split boxes in the first two columns, with positive examples in solid lines and negative examples in dotted lines. Not shown are edges connecting each partition in one column with each partition in the next and the paths they create. The intersection of the partitions in path (a) lead to a null set, and the path is removed from the lattice. Path (b) is the ground truth path, as the individual partitions accurately describe the composition of the attributes. Path (c) contains an overspecified edge and achieves the correct referent set albeit using incorrect assumptions about the attributes. The result sets from both (b) and (c) agree with the quantifier constraint (definite singular).

4.2 Node Probabilities

We consider two probabilities for determining the probability of a partition: that which can be determined from distance data (considering distances between objects in the partition), and that which requires previous labeled data to hypothesize a class using the classifier (considering distances from each object to the mean of the data labeled with the descriptor).

The distance probability, our implementation of the principle of Contrast, is a prior that enforces minimum intraclass distance for the positive examples and maximum interclass distance across the partition. The motivation and implementation shares some similarities with the Diverse Density framework for multiple instance learning (Maron and Lozano-Pérez, 1998), although here it also acts as an unsupervised clustering for determining the best reference set. It is the product of the minimum probability that any two objects in the positive examples are in the same set multiplied by the complement of the maximum probability that any two objects across the partition are in the same class. Therefore, for partition N with positive examples $+$ and negative examples $-$:

$$P_{intra} = \min_{x,y \in +} P(x_c = y_c)$$

$$P_{inter} = \begin{cases} \max_{x \in +, y \in -} P(x_c = y_c) & \text{if } |-| > 0 \\ 1 & \text{if } |-| = 0 \end{cases}$$

$$P_{distance} = P_{intra} \times (1 - P_{inter})$$

The classifier probability is similar, except rather than comparing objects to other objects in the partition, the objects are compared to the mean of the column's descriptor C in the descriptor's representative feature. If the descriptor is a class name, we instead choose the Zernike shape feature, implementing the shape bias children show in word learning (Landau et al., 1998).

If there is insufficient labeled data to use, then the classifier probability is set to 1 for the entire column, meaning only the distance probabilities will affect the probabilities of the nodes. For a given descriptor C, the classifier probabilities are as follows:

$$P_{pos}(C) = \min_{x \in +} P(x_c = C)$$

$$P_{neg}(C) = \begin{cases} \max_{x \in -} P(x_c = C) & \text{if } |-| > 0 \\ 1 & \text{if } |-| = 0 \end{cases}$$

$$P_{classifier}(C) = P_{pos}(C) \times (1 - P_{neg}(C))$$

The final probability of a partition is the product of the distance probability and the classifier probability, and the node probabilities are normalized for each column.

4.3 Overspecification and Edge Probabilities

Edges have a constant transition probability equal to the overspecification probability if overspecified, or equal to the complement otherwise. We use these probabilities to incorporate the phenomenon of overspecification in our model, where, contrary to a strict interpretation of Grice's Maxim of Quantity, speakers will give more information than is needed to identify a referent (Koolen et al., 2011). An edge is considered overspecified if the hypothesis for the objects that satisfy the next descriptor does not add additional information, i.e., the set-intersection it corresponds to does not remove any possible objects from the referent set. Thus the model will prefer hypotheses for the next descriptor that narrow down the hypothesized set of referents.

4.4 Path Probabilities

The probability of each path is the product of probabilities of each of the partitions along its path and the edge (overspecification) probabilities. If there is a single path with a probability greater than all others by an amount ϵ, the labels of the partitions along that path are assigned to the positive examples while also being assigned as negative properties for the negative examples. We perform this updating step after each utterance to simulate incremental continuous language learning and to provide the most current knowledge available for resolving new ambiguous data.

If there are multiple best paths within ϵ of the highest probability path, then the learning example is considered ambiguous and saved in memory to resolve with information from future examples.

4.5 Multiple-Instance Learning

In many cases, especially in the system's first learning instances, there is not enough information to unambiguously learn from the demonstration. Without any unambiguous examples, our system would struggle to learn no matter how much data was available to it. An ambiguous training example yields more than one highest probability path. Our goal is to use new information from each new training demonstration to reevaluate these paths and determine a singular best path, which allows us to update our knowledge accordingly.

To do this, we independently consider columns for each unknown descriptors from unresolved demonstrations containing that descriptor and combine them to form super-partitions which are then evaluated using our distance probability function. For example, consider two instances described with "the red box". The first has a red and a blue box, while the second has a red and a green box. Individually they are ambiguous to a system that does not know what "red" means and therefore each demonstration would have two paths with equal probability. If we combine the partitions across the two demonstrations into four super-partitions, the highest probability will be generated when the two red boxes are in the positive set. This probability is stored in each of the constituent partitions as a *meta-probability*, which is otherwise 1 when multiple-instance learning is not required to resolve ambiguity. The meta-probability allows us to find the most probable path given previous instances.

5 System Pipeline

5.1 Training

To train the system on a video, we transcribe the video with sentence-level timestamps, and extract features from the demonstration video. The system takes as input the feature data aligned with utterances from the demonstration video. It then finds the most likely path through the reference lattice and adds all hypothesized positive examples for the descriptor as class examples for the classifier. If there is more than one likely path, it saves the lattice for later resolution using multiple-instance learning.

5.2 Description Generation

During testing, the system generates a description for an object in the test set by finding examples of properties and objects similar to it in previously seen objects. For properties, the system checks each feature space separately to find previous examples of objects similar in that feature space and adds each found property label to the k-NN voting set, weighted by the distance. If the majority label does not have the matching representative feature, the system skips this feature space for adding a property to the description. The object name is chosen using a distance generated from the sum of the distances (normalized and weighted through the Mahalanobis distance metric) to the most similar previous examples. More details about the description generation process can be found in our previous paper (Perera and Allen, 2013).

6 Evaluation

To evaluate our system, we use two metrics: our evaluation method used in previous work for rating the quality of generated descriptions (Perera and Allen, 2013), and a standard precision/recall measurement to determine the accuracy of reference resolution.

The description generated by the system is compared with a number of possible ground truth descriptions which are generated using precision and recall equivalence classes from our previous work. Precision is calculated according to which words in the description could be found in a ground truth description, while recall is calculated according to which words in the closest ground truth description were captured by the system's description. As an example, a system output of "red rectangle"

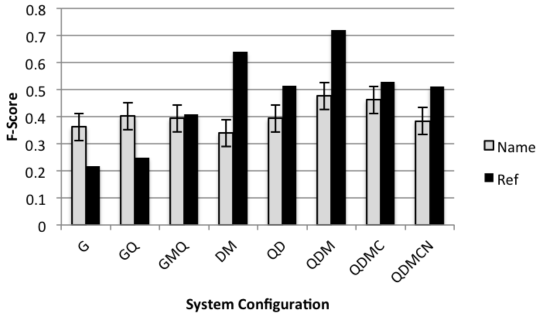

Figure 4: F-Score for Description Generation in grey and Reference Resolution in black for various configurations of the system run on 4 underspecified videos. Error bars are one standard deviation.

when the ground truth description is “red square” or “red cube” would have a precision score of 1 (because both “red” and “rectangle” are accurate descriptors of the object) but a recall of .5 (because the square-ness was not captured by the system’s description).

In the reference resolution evaluation, precision and recall are calculated on the training set according to the standard measures by comparing the referent set obtained by the system and the ground truth referent set (those objects actually referred to by the speaker). Training instances lacking feature data because of an error in recording were excluded from the F1-score for reference resolution.

Each underspecified demonstration video consisted of 15-20 demonstrations containing one or more focal objects referenced in the description and, in most cases, distractor objects that are not mentioned. We used the same test video from our previous work with objects removed that could not be described using terms used in the training set, leaving 15 objects.

We tested eight different system configurations. The baseline system simply guessed at a path through the lattice without any multiple-instance learning (G). We then added multiple instance learning (M), distance probabilities (D), classifier probabilities (C), quantifier constraints (Q), and negative information (N). We show the data for these different methods in Figure 4.

7 Results and Discussion

7.1 Learning Methods

Rather than comparing our language learning system to others on a common dataset, we choose to focus our analysis on how our implementations of pragmatic inference and child language learning strategies affected performance of reference resolution and description generation.

The relatively strong naming performance of G can be attributed to the fact that many demonstrations had similarities among the objects presented that could be learned from choosing any of the objects. However, reference resolution performance for G averaged a .34 F1-score compared with a .70 F1-score for our best performing configuration. Adding quantifier constraints (GQ) did not help, although quantifier constraints with multiple-instance learning (GMQ) led to a significant increase in reference resolution performance.

Multiple-instance learning provided a significant gain in reference resolution performance, and with quantifier constraints also yielded the highest naming performance (QDM and QDML). The relative lower performance by inclusion of classifier probabilities with this limited training data is due to errors in classification that compound in this online-learning framework. In multiple-instance cases where there are a number of previous examples to draw from, then the information provided by classifier probability is redundant and less accurate. However, as the approach scales and retaining previous instances is intractable, the classifier probabilities provide a more concise representation of knowledge to be used in future learning.

We found that negative information hurt performance in this framework (QDMCN vs. QDMC) for two reasons. First, the risk of introducing negative information is high compared to its possible reward. While it promises to remove some errors in classification, an accurate piece of negative information only removes one class from consideration when multiple other alternatives exist, while an inaccurate piece of negative information contributes to erroneous classification.

Second, situations where negative information might be inferred are induced by a natural language description which, by Grice’s Maxims, will attempt to be as clear as possible given the listener’s information. This means that, adhering to the Contrast principle, negative examples are likely already far from the positive examples for the class.

Figure 5 shows results from the averaging of random combinations of 4 underspecified videos, using our highest-scoring configuration QDM to

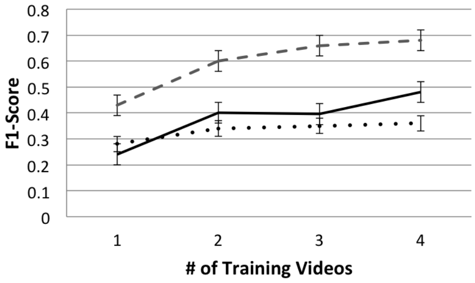

Figure 5: Description generation results from training according to the number of training videos with standard error bars. The solid line is the QDM's performance learning from underspecified videos. The dashed line is the system's performance learning from videos where objects are presented one at a time. The dotted line is the baseline (G). F1-score for reference resolution in the underspecified case was consistent across videos (mean .7, SD .01).

show the increase in performance as more training data is provided to the system. We compare our results on videos with multiple objects to the performance of the system with objects presented one at a time and with the baseline G. Because the training objects are slightly different, we present results on a subset of objects where at least a ground truth object name was present in the training data. Our results show that while the performance is lower in the ambiguous case, the general learning rate per video is comparable with the single-object case. In the 1-video case, guessing is equally as effective as our method due to the system being too tentative with assigning labels to objects without more information to minimize errors affecting learning in later demonstrations.

We did see an effect of the order in which videos were presented to the system on performance, suggesting that learning the correct concepts early on can have long-term ramifications for an online learning process. Possible ways to mitigate this effect include a memory model with forgetting or a more robust classifier. We leave such efforts to future work.

7.2 Running Time Performance

While the number of nodes and paths in the lattice is exponential in the number of objects in the scene, our system can still perform quickly enough to serve as a language learning agent suitable for real-time interaction. The pragmatic constraints on possible referent sets allow us to remove a large number of paths, which is especially important when there are many objects in the scene or when the referring expression contains a number of descriptors. In situations with more than 4-5 objects, we expect that other cues can establish joint attention with enough resolution to remove some objects from consideration.

Visual features can be extracted from video at 3 frames per second, which is acceptable for real-time interaction as only 5 frames are needed for training or testing. Not including the feature extraction (performed separately), the QUM configuration processed our 55 demonstrations in about 1 minute on a 2.3 GHz Intel Core i7.

7.3 Relation Between Evaluation Metrics

We compared our results from the description generation metric with the reference resolution metric to evaluate how the quality of reference resolution affected learning performance. The description generation F-score was more strongly positively correlated with the reference resolution precision than with the recall. We found a reference resolution F-score with $\beta = .7$ (as opposed to the standard $\beta = 1$) had the highest Pearson correlation with the F-score ($r = .63, p < .0001$), indicating that reference resolution precision is roughly 1.4 times more important than recall in predicting learning performance in this system.

This result provides evidence that the quality of the first data in a limited data learning algorithm can be critical in establishing long-term performance, especially in an online learning system, and suggests that our results could be improved by correcting hypotheses that once appeared reasonable to the system. It also suggests that F1-score may not be the most appropriate measure for performance of a component that is relied upon to give accurate data for further learning.

7.4 Overspecification

Accounting for overspecification in the model more closely approximates human speech at the expense of a strict interpretation of the Maxim of Quantity. It allows us to use graded pragmatic constraints that admit helpful heuristics for learning without treating them as a rule. In our training data, the speaker was typically over-descriptive, leading to a high optimal overspecification. Figure 6 shows the effect of different values for the overspecification probability on the performance of the

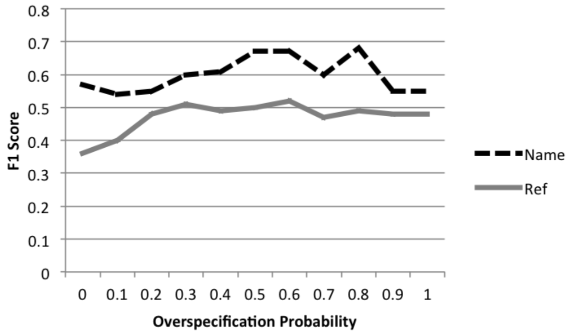

Figure 6: Effect of varying overspecification probability on the F1 score for both Description Generation (black dashed) and Reference Resolution (grey solid), calculated on a dataset with hand location information.

system. The strong dip in reference resolution performance at an overspecification probability of 0 shows the significant negative effect a strict interpretation of the Maxim of Quantity would have in this situation. The correct value for overspecification probability for a given situation depends on a number of factors such as scene complexity and descriptor type (Koolen et al., 2011), but we have not yet incorporated these factors into our overspecification probability in this work.

7.5 Comparison to Other Multi-Instance Learning Methods

Our multi-instance learning procedure can be classified as instance-level with witnesses, which means that we identify the positive examples that lead to the label of the "bag", or demonstration in this case. In addition, we relax the assumption that there is only a single positive instance corresponding to the label of the demonstration. This relaxation increases the complexity of cross-instance relationships, but allows for references to multiple objects simultaneously and therefore faster training than a sequential presentation would allow. In accounting for overspecification, we also must establish a dependence on the labels of the image via the edges of the lattice. This adds additional complexity, but our results show that accounting for overspecification can lead to increased performance.

8 Future Work

Work on this system is ongoing, with extensions planned for improving performance, generating more complete symbol grounding, and allowing more flexibility in both environment and language.

While the parser in our system can interpret phrases such as "the tall block", we do not have a way of resolving the non-intersective predicate "tall" in our current framework. Non-intersective predicates add complexity to the system because their reference point is not necessarily the other objects in the scene - it may be a reference to other objects in the same class (i.e., blocks).

Also, our set of features is rather rudimentary and could be improved, as we chose low-dimensional, continuous features in an attempt to facilitate a close connection between language and vision. The use of continuous features ensures that primitive concepts are grounded solely in perception and not higher-order conceptual models (Perera and Allen, 2014). Initial results using 3D shape features show a considerable performance increase on a kitchen dataset we are developing.

9 Conclusion

We have proposed a probabilistic framework for using pragmatic inference to learn from underspecified visual descriptions. We show that this system can use pragmatic assumptions attenuated by overspecification probability to learn attributes and object names from videos that include a number of distractors. We also analyzed various learning methods in an attempt to gain a deeper understanding of the theoretical and practical considerations of situated language learning, finding that Conventionality and Contrast learning strategies with quantifiers and overspecification probabilities yielded the best performing system. These results support the idea that an understanding of how humans learn and communicate can lead to better visually grounded language learning systems. We believe this work is an important step towards systems in which natural language not only stands in for manual annotation, but also enables new methods of training robots and other situated systems.

10 Acknowledgements

This work was funded by The Office of Naval Research (N000141210547), the Nuance Foundation, and DARPA Big Mechanism program under ARO contract W911NF-14-1-0391.

References

J. Allen, Mary Swift, and Will de Beaumont. 2008. Deep Semantic Analysis of Text. In *Symp. Semant. Syst. Text Process.*, volume 2008, pages 343–354, Morristown, NJ, USA. Association for Computational Linguistics.

D A Baldwin. 1992. Clarifying the role of shape in children's taxonomic assumption. *J. Exp. Child Psychol.*, 54(3):392–416.

T Brown and J Koplowitz. 1979. The Weighted Nearest Neighbor Rule for Class Dependent Sample Sizes. *IEEE Trans. Inf. Theory*, I(5):617–619.

Eve V. Clark. 2009. On the pragmatics of contrast. *J. Child Lang.*, 17(02):417, February.

T Dietterich. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.*, 89:31–71.

Haris Dindo and Daniele Zambuto. 2010. A probabilistic approach to learning a visually grounded language model through human-robot interaction. *IEEE/RSJ 2010 Int. Conf. Intell. Robot. Syst. IROS 2010 - Conf. Proc.*, pages 790–796.

A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. 2009. Describing objects by their attributes. *2009 IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1778–1785, June.

James Foulds and Eibe Frank. 2010. A review of multi-instance learning assumptions. *Knowl. Eng. Rev.*, 25:1.

HP Grice. 1975. Logic and conversation. In Peter Cole and Jerry L. Morgan, editors, *Syntax Semant.*, pages 41–58. Academic Press.

Robert M. Haralick, K. Shanmugam, and Its'hak Dinstein. 1973. Textural features for image classification. *IEEE Trans. Syst. Man, Cybern. SMC-3*.

A Khotanzad and Y H Hong. 1990. Invariant Image Recognition by Zernike Moments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(5):489–497, May.

Thomas Kollar, Jayant Krishnamurthy, and Grant Strimel. 2013. Toward interactive grounded language acquisition. *Proc. Robot. Sci. Syst.*

Ruud Koolen, Albert Gatt, Martijn Goudbeek, and Emiel Krahmer. 2011. Factors causing overspecification in definite descriptions. *J. Pragmat.*, 43(13):3231–3250, October.

Jayant Krishnamurthy and Thomas Kollar. 2013. Jointly Learning to Parse and Perceive: Connecting Natural Language to the Physical World. *Trans. Assoc. Comput. Linguist.*, 1:193–206.

Barbara Landau, Linda Smith, and Susan Jones. 1998. Object Shape, Object Function, and Object Name. *J. Mem. Lang.*, 38(1):1–27, January.

Mu Li, James T. Kwok, and Bao Liang Lu. 2010. Online multiple instance learning with no regret. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 1395–1401.

PC C Mahalanobis. 1936. On The Generalized Distance in Statistics. *Proc. Natl. Inst. Sci. India*, pages 49–55.

Oded Maron and Tomás Lozano-Pérez. 1998. A framework for multiple-instance learning. *Adv. Neural Inf. Process. Syst.*, 10:570 – 576.

Cynthia Matuszek, N FitzGerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. 2012. A Joint Model of Language and Perception for Grounded Attribute Learning. In *Proc. Int. Conf. Mach. Learn.*

Shiwali Mohan, John E Laird, and Laird Umich Edu. 2013. Towards an Indexical Model of Situated Language Comprehension for Real-World Cognitive Agents. 2013:153–170.

Iftekhar Naim, Young Chol Song, Qiguang Liu, Henry Kautz, Jiebo Luo, and Daniel Gildea. 2014. Unsupervised Alignment of Natural Language Instructions with Video Segments. In *AAAI*.

Ian Perera and JF Allen. 2013. SALL-E: Situated Agent for Language Learning. In *Twenty-Seventh AAAI Conf. Artif. Intell.*

Ian Perera and James F Allen. 2014. What is the Ground ? Continuous Maps for Grounding Perceptual Primitives. In P Bello, M. Guarini, M McShane, and Brian Scassellati, editors, *Proc. 36th Annu. Conf. Cogn. Sci. Soc.*, Austin, TX. Cognitive Science Society.

A C Rencher. 2003. *Methods of Multivariate Analysis*. Wiley Series in Probability and Statistics. Wiley.

Danijel Skocaj, Matej Kristan, Alen Vrecko, Marko Mahnic, Miroslav Janicek, Geert-Jan M. Kruijff, Marc Hanheide, Nick Hawes, Thomas Keller, Michael Zillich, and Kai Zhou. 2011. A system for interactive learning in dialogue with a tutor. In *2011 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pages 3387–3394. IEEE, September.

Linda Smith and Chen Yu. 2008. Infants rapidly learn word-referent mappings via cross-situational statistics. *Cognition*, 106:1558–1568.

Kenny Smith, Andrew D M Smith, and Richard a. Blythe. 2011. Cross-situational learning: An experimental study of word-learning mechanisms. *Cogn. Sci.*, 35:480–498.

John C Trueswell, Tamara Nicol Medina, Alon Hafri, and Lila R Gleitman. 2013. Propose but verify: fast mapping meets cross-situational word learning. *Cogn. Psychol.*, 66(1):126–56, February.

W Van Orman Quine. 1964. *Word and Object*. MIT Press paperback series. MIT Press.

Haonan Yu and Jeffrey Mark Siskind. 2013. Grounded Language Learning from Video Described with Sentences. In *Proc. 51st Annu. Meet. Assoc. Comput. Linguist.*, pages 53–63.