Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

SALL-E: Situated Agent for Language Learning

Ian Perera and **James F. Allen**
Department of Computer Science,
University of Rochester, Rochester NY, USA
{iperera,james}@cs.rochester.edu

Abstract

We describe ongoing research towards building a cognitively plausible system for near one-shot learning of the meanings of attribute words and object names, by grounding them in a sensory model. The system learns incrementally from human demonstrations recorded with the Microsoft Kinect, in which the demonstrator can use unrestricted natural language descriptions. We achieve near-one shot learning of simple objects and attributes by focusing solely on examples where the learning agent is confident, ignoring the rest of the data. We evaluate the system's learning ability by having it generate descriptions of presented objects, including objects it has never seen before, and comparing the system response against collected human descriptions of the same objects. We propose that our method of retrieving object examples with a *k*-nearest neighbor classifier using Mahalanobis distance corresponds to a cognitively plausible representation of objects. Our initial results show promise for achieving rapid, near one-shot, incremental learning of word meanings.

Introduction

One of the great challenges for cognitive systems is grounded language learning. How can an agent learn the meaning of words by associating them with objects, events, and situations in the world? An additional challenge is accounting for how children learn language so quickly, often learning new words from just one training instance. This ability stands in stark contrast to previous computational models of language learning that use statistical association over large amounts of training data (e.g. work by Yu and Ballard (2004) and Roy and Pentland (2002)). We report initial results on a project that attempts to achieve one-shot learning of word meanings by incorporating some of the heuristics that children appear to use. The most critical heuristic we explore in this paper is selective learning from training instances that are just beyond the scope of the agents current understanding. By focusing solely on instances when most of a scene and much of the language is understood, we can provide a rich context that allows the agent to learn from the novel parts of the situation and language.

Copyright © 2013, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

A child's environment is crucial to language learning, as it provides referents for new words in a context and grounds them in an experience that is necessary for applying that knowledge to new situations. Learning a new word is a complex process. When the child first hears an unknown word, they can use syntactic clues to guess the word's part of speech. Next, they must determine the referent for the word - is it an object? A property? An action? There are numerous possibilities, but children use heuristics such as mutual exclusion, shape bias, and joint attention to narrow down the choices. When considering this in the context of artificial intelligence, we see this advantage in the core of the symbol grounding hypothesis (Harnad 1990), which claims that all words must eventually be grounded in an experience, whether directly or through other words.

Consider an example of learning from almost-understood situations. Say the agent hears the utterance, "This is a red truck" in an environment containing objects on a table. There is a car and truck on the table proximate to the demonstrator's hands, and that the learning agent currently doesn't know anything about trucks. Using its existing knowledge of syntax and the meaning of the words "this", "is", "a" and "red", its parser can hypothesize that the unknown word "truck" is a noun, and being in the context of a physical demonstration, learn that it probably names a physical object. Using joint attention with the proximity to the demonstrator's hands, the agent assumes that the object being referred to is either the car or the truck. There are many ways it might conclude that the truck is the object being described. If it already knows how to identify cars, then the heuristic of mutual exclusion will lead it to conclude that the new unknown word "truck" corresponds to the new unknown object, the truck. If neither the car or truck are known, but the agent knows the meaning of red, then it can use the knowledge that the new object is red in order to identify the intended object. Because the language and situation were mostly understood, the agent is able to create a very high value training instance for the meaning of the word "truck".

By leveraging the partial understanding it develops over time, our agent can identify high-quality training instances, allowing effective near-one-shot learning. In addition, the approach intuitively has great promise for accounting for the incremental nature of language learning, where the more you learn the more effectively you are able to learn. Our

preliminary results reported here indicate a significant advantage to this strategy: we significantly improve results by throwing away much of the available training data that would be used in associative learning techniques and focusing on these high-quality cases, and demonstrate near one-shot learning in simple situations.

Related Work

Early work with similar motivations towards computational language learning can be found in EBLA (Pangburn, Mathews, and Iyengar 2003), which used a similar environment with proto-English rather than unrestricted natural language. It learns object names as well as actions, an area of future work for this project, but does not learn attributes. Both their system and ours use simple features that feasibly map to natural language attributes. Although their results include action recognition, our identification results prove to be at least comparable to theirs with much less training data.

One of our goals is to build a system that could be trained and tested in real-time without annotations. Work by Matuszek et al. (2012) addresses a discriminative version of our task (including language learning through probabilistic categorial grammar induction), where features are assigned to descriptive and object words and then used with the language model to pick the described subset of visible objects. Their robust image features are trained with a batch process on annotated images and representative features are indicated by different feature weights for different words. While they achieve promising results, we focus on a task that does not require their annotations from Mechanical Turk, which, while improving the data collection bottleneck, does not facilitate learning in a natural environment.

We also hope to take advantage of a simpler environment and explore how knowledge can be built from the ground up in tandem with language development. The problem of attribute learning has been mainly tackled from the field of computer vision with large datasets, such as work done by Farhadi et al. (2009), which used larger annotated data sets but achieved impressive results on more complicated, natural images. Work by Fei-Fei, Fergus, and Perona (2006) showed initial results towards the goal of one-shot learning for the task of determining whether an object in a given category appeared in an image or not, using similarly annotated data. However, these results do not lead any insights into how language and attribute learning complement each other, nor do they explore learning in a natural environment.

We choose to learn words and concepts, rather than phonemes as in work by Roy and Pentland (2002). Roy and Pentland tackled symbol grounding, word learning, and segmentation simultaneously, assuming no prior lexical, syntactic, or semantic knowledge. We choose to approach language learning assuming the child already has the capability for segmenting words and identifying basic parts of speech using syntactic context. This allows us to both develop richer models of word meanings in the future as well as utilize sentence-level context to learn word meanings.

Learning Environment and Data Collection

Our training and test environment consists of a table with various colored blocks and toys. A person stands behind the table, places one or two objects at a time in a designated demonstration area on the table, and describes the objects in unconstrained natural language while pointing to them, providing an explicit indication of attention. 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 feature data extracted from each frame. Video is recorded using the Microsoft Kinect to obtain RGB + Depth information and skeleton tracking for detecting hand locations.

The motivation for this type of environment was to more closely approximate one that a child would be exposed to when learning words. We acknowledge that children may not need to be explicitly spoken to to learn words, but this method of data collection has advantages that allow us to focus on the goals of the project without dealing with unrelated factors such as belief modeling of multiple people or greater context awareness. Subjects need only a brief introduction to the task required, our attentional mechanisms can focus on a static environment, and lighting conditions remain consistent. The "blocks world" environment also allows us to focus on learning names for basic properties without having to consider more complicated object models. However, we do include a number of more complicated objects, such as toy cars and trucks.

Language Processing

The transcribed data is passed through the TRIPS parser (Allen, Swift, and de Beaumont 2008) for simultaneous lexicon learning and recognition of object descriptions. Words are part-of-speech tagged by the Stanford POS tagger, then passed to the TRIPS parser with dependency information from the Stanford dependency parser. The TRIPS parser generates a semantic parse from this input, with concepts filled in from the ontology if they are known.

Lexicon Learning

While we use the full TRIPS parser, we start with a limited initial lexicon of the 500 most commonly used words in English together with the associated subpart of the TRIPS ontology. When an unknown word is encountered, the parser constructs an underspecified lexical entry and then uses surrounding syntax to find the syntactic and semantic features for that word that allow the most semantically plausible interpretation. As a simple example, given a sentence, "This is a blue block", where *block* is the unknown word, the system determines that the most plausible parse involves *block* as a noun naming a physical object. This new word is then added to the lexicon and ontology as a subclass of PHYS-OBJECT.

Speech Act Recognition

The combination of syntactic and semantic parsing information from the parser is then passed to a rule-based system for identifying various speech acts involving objects. A *demonstration* speech act, where the subject introduces and

Figure 1: The learning environment.

describes an object, is recognized by analyzing the parse for proximal deictic expressions, such as *here* or *this*. A *mention* speech act is signified by a mention of an object that is not explicitly introduced. A *describe* speech act is one that includes a reference to a previously mentioned object and adds additional information about it. Decomposing the language input into these speech acts allows the agent to take in language that might occur in a more natural environment rather than one consisting of a person simply naming objects. Finally, during testing, the subject says some variation of "What is this?" to indicate a request for identification of the object being pointed to.

Feature Extraction

As we are prioritizing learning semantically meaningful representations of properties and objects, our focus is on generating natural language interpretable representations that can be tied to concepts. To this end, we do not use local feature-matching algorithms such as SIFT (Lowe 1999) or HOG (Dalal and Triggs 2005), and instead attempt to capture generalizable properties such as shape, color, and texture that might occur in different combinations than those seen in objects in the training set.

We first perform object segmentation using Kinect depth information, which provides a pixel-level contour around each of the objects in the scene. Then, we convert color within the contour from RGB space to Lab color space, which has the advantage that the Euclidean distance metric is roughly analogous to human perception of color. In addition, the separation of the luminance value from the chroma values allows for some invariance to changes in lighting. Shape is captured using scale- and rotation-invariant Hu moments (Hu 1962) and Zernike moments (Khotanzad and Hong 1990). Zernike moments can be used not only for classification, but also for sampling an example of the stored shape data. We also extract the color variance for a rudimentary texture feature and the size of the object in pixels. Segmentation and feature extraction can be calculated at 8-10 frames per second, enabling interactive sessions in the future.

Classification and Description Generation

Our task for the system is to generate a description of an object consisting of its salient features, such as color, shape, and size, and its name.

Matching Words to Feature Spaces

Although we can record the values of the features corresponding to mentions of a descriptive word, we want to learn which of those features is actually relevant to the meaning of the word. To do this, we calculate a ratio for each feature space: the scaled sum of square deviations for the word to that of all of the feature data. The feature space with the lowest ratio is assigned to that word as its representative feature, although it is continuously reevaluated as new data is presented to the system. Intuitively, we are comparing the variance of the data when a descriptive word is present to when it is not. If a word decreases the expected variance of a feature space, then it is likely that word conveys some meaning with respect to that feature. A similar idea was used in work by Pangburn, Mathews, and Iyengar (2003) - however, with our multi-dimensional features, our method performs better than an element-wise variance ratio.

Learning the representative feature for one word, whether from an ontology or from experience, also conveys information about other representative features. For example, if we had a training environment where all tall blocks were blue, but the agent saw some short blocks, it could learn that the representative feature of "blue" is color, and that would explain the low variance in the color dimension of the tall blue blocks, leaving only the height-width ratio to be explained by the term "tall".

For object names, we do not expect that the word can be tied to one particular feature, but rather some combination of features. However, we can still use the above method to learn how relevant different features are to the meaning of an object.

k-Nearest Neighbor Classifier

A k-Nearest Neighbor classifier (k-NN) returns the classification that has the highest score of the k closest examples according to some distance metric (usually Euclidean distance over the feature vectors). This method has the advantage that we do not require any time for batch processing and can therefore feasibly work towards an interactive system. Examples are weighted according to the number of times their classification has appeared so far, using the method described in Brown and Koplowitz (1979) to compensate for the effect the prior probabilities of classifications have on k-NN for large k.

k was chosen using a dev set of 18 objects. As our k-NN classifier is distance-weighted, small changes in k do not have a significant effect on results. However, a k set too large will tend towards the prior probability of the objects and attributes, which is undesirable since we do not expect new objects to follow the same distribution. We found that our chosen k value provided consistent results even as more training data was added, yet scaling beyond 5 videos may require a k that is dependent on the amount of training data stored.

Mahalanobis Distance We use the Mahalanobis distance metric (Mahalanobis 1936) in place of Euclidean distance for object classification. With this distance metric, distances are scaled according to the precision matrix (the inverse of the covariance matrix), providing both a way of reducing the influence of distances along dimensions irrelevant to the current descriptive word and of normalizing distances across different feature spaces to create a single distance value for object classification.

The Mahalanobis distance metric can be seen as a kind of feature weighting both within dimensions of features and between whole features. For example, for a given color word, the lightness dimension of Lab color space will vary more than the color dimensions. Therefore, distance in the lightness dimension will have a reduced effect on classification. Scaling features in this manner also allows us to combine disjoint features of varying dimensions and distributions, allowing greater flexibility for future features.

To see how the Mahalanobis distance metric is a plausible semantic representation for a cognitive system, consider an experience with an apple. After seeing a number of apples that all happen to be red, we would expect that redness is a property of apples. We would also have a shape and texture associated with an apple, and would use these properties to identify whether a new object is an apple or not. Upon seeing different color apples, however, the variance in the color space would increase, and while we would usually identify objects as apples if their color was consistent with our previous experiences, we would also be more open to the idea that an apple could be a different color, and color would be a less informative and intrinsic property of apples.

With a Gaussian mixture model, an agent would have to know the number of colors of an apple beforehand, lest a single Gaussian fitted to the color data would assign a high likelihood to orange apples. Nonparametric mixture models like Dirichlet Mixture Models would address this issue, but would likely require much more training data to form the correct representation. However, we may consider them for future work. Our results show that considering the precision of a feature as a measure of its intrinsic nature for the meaning of an object remains a reliable metric even with a different type of classifier or model.

Description Generation When generating a description for an object, we must choose the best object name as well as a number of descriptive words. The object name is chosen according to the k-NN using the sum of the Mahalanobis distances from each of the feature spaces. For choosing adjectives, we only consider one feature space at a time, and choose the adjective according to k-NN. However, it may be that a feature space is not particularly salient for this example or there may be conflicting descriptions. To account for this, we also consider "null" examples in the k-NN selection - those with descriptions that are not associated with that feature space - and assign them a low weight. If the highest score comes from null examples, or if there is a strong conflict between different descriptions, we do not generate a description for that feature space.

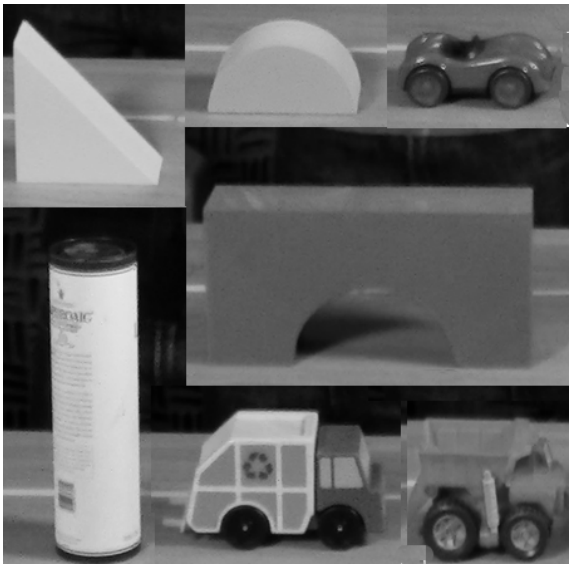

Figure 2: A selection of objects the system learns.

Evaluation

To evaluate our system, we compare the descriptions created by a number of human judges with those generated by the system and compute precision and recall over the descriptions. Nine human judges are shown the most recent video used for training, then the test video, and are asked to provide descriptions for each of the objects in natural language, focusing on physical attributes rather than the function of the object or related details. They are not told what words to use or what features the system can recognize. In our evaluation, we allow for synonymous matches ("rectangle" is equivalent to "rectangular block") and remove non-descriptive words.

$$\text{Precision} = \frac{\text{\# of words in both desc. and a human desc.}}{\text{\# words in desc.}}$$

$$\text{Recall} = \frac{\text{\# of words in desc. and closest human desc.}}{\text{\# words occurring in closest human desc.}}$$

Words that accurately describe the object but do not fully capture the words in the closest human description do not count against precision. For example, if the system generates "yellow block", and the closest human description is "tall yellow rectangle", precision would be 100% and recall would be 33.3%. An example evaluation is given in Table 1.

Human Desc.	SALL-E Desc.	Precision	Recall
Red cube	red box	1	1
Red box	red rectangle	1	.5
Red square	yellow cube	.5	.5

Table 1: Example evaluation for one test demonstration described three ways by humans with three possible descriptions generated by SALL-E. Note that, in the second description, as a square is a rectangle, it does not affect precision.

Results and Discussion

Each demonstration video consisted of 15-20 objects described in natural language. We recorded one test video of 20 objects to be described by the system. Results from random combinations of 5 training videos are averaged to show the typical increase in performance as more training data is provided to the system. "Unseen" results are the results from objects that have not been seen in the training data, although typically either the object or its attributes have been demonstrated as part of another object. We also include results with more training videos that consequently have few unseen objects to demonstrate that the system can continue to refine its models of objects. Inter-annotator agreement was measured by applying the evaluation criteria for each judge to all other responses, yielding 93% precision and 92% recall for human level performance.

In Table 2, we present results from three configurations:
Supervised Features - Features spaces are assigned to words by hand (e.g., "blue" is mapped to the color space).
Unsupervised Features - Feature spaces are assigned to words using the ratio of sum of square deviations.
POS Only - No feature space is assigned, and all cooccurring adjectives and nouns are possible descriptive words. The words are chosen using *k*-nearest neighbor with Mahalanobis distance. If instead all words are possible choices, then the maximum f-score we obtained was .34 with all training videos used - this low result shows the advantage of even rudimentary language understanding in constraining the search space over a simple word cooccurrence model.

# of Videos		1	2	3	5
Supervised Features	Precision	.48	.65	.70	.78
	Recall	.40	.56	.62	.64
	F-score	**.43**	**.60**	**.66**	**.70**
Unsupervised Features	Precision	.41	.69	.66	.79
	Recall	.32	.59	.58	.70
	F-score	**.35**	**.63**	**.61**	**.74**
POS Only	Precision	.39	.55	.56	.65
	Recall	.31	.42	.48	.52
	F-score	**.34**	**.48**	**.52**	**.58**

Table 2: Results for testing on entire test set.

# Videos		1	2	3
Supervised Features	Precision	.46	.51	.64
	Recall	.42	.42	.56
	F-score	**.43**	**.46**	**.59**
Unsupervised Features	Precision	.38	.51	.52
	Recall	.29	.45	.47
	F-score	**.32**	**.48**	**.50**
POS Only	Precision	.31	.45	.39
	Recall	.23	.31	.30
	F-score	**.26**	**.36**	**.34**
# of unseen objects		14	9	7

Table 3: Results for testing on only unseen objects in test set.

We can test whether the system is learning properties as opposed to learning to identify objects by looking at the results on previously unseen objects. One would expect testing on the training objects to provide significantly higher scores than testing on unseen objects. However, Table 3 shows that using previously unseen objects yields only a small decrease in performance. This suggests our system is learning the attributes of objects rather than matching words to previous occurrences.

# of Videos	1	2	3	5
% of properties learned	40	70	80	78

Table 4: The percentage of learned properties correctly mapped to a semantically matched representative feature.

After only three videos, or about 50 demonstrations, our system shows promising results, as the expected precision and recall for randomly choosing a description for an object in the test set is .11. Furthermore, this task is more difficult than most situations an agent would face in normal conversation - in many cases, the agent would only need to pick out the object in a given environment once it is mentioned, not provide an accurate description for each object it sees. Although color is typically salient and easier to learn, the system's performance does not seem to be the result of any one feature being particularly accurate - accuracy for properties and objects were roughly equal.

The typical increase in performance from choosing representative features by hand shows the effect a developed ontology could have on object and attribute learning. We can see how *a priori* knowledge of attributes allows for faster learning of attributes by restricting the search space for a matching description. In the other direction, learning representative features for attributes provides useful information that could be added to a developing ontology. The number of properties correctly learned and thus suitable for an ontology is shown in Table 4. In some cases, learning the features automatically provides better results on unseen objects, as our world knowledge can tell us that a word corresponds to a shape, but not which shape feature represents it most accurately. In the case of using all five training videos, we also see that the unchosen representative feature test outperforms the prechosen features. However, this may be a result of the system determining which representative features would be best for classifying the training set, and therefore we cannot always assume the increased performance will generalize to the same extent as the semantically "true" representative features.

The decrease in performance when considering only part-of-speech tags for descriptive words can be explained by two effects: one, there is no distinction between attributes, which are assigned to a single feature, and objects, which are a combination of features, and two, more complicated discourse will lead the simple cooccurrence assumption astray. The first effect is significant in this training data because much of the training descriptions happened to be in rather simple language. In situations with multiple objects and less direct descriptions, the advantage of using discourse processing will be even more apparent.

While the results may not seem impressive compared to other computer vision work, this is primarily due to our feature set consisting only of features with natural language analogues. The object and attribute recognition process is not entirely decomposable into semantic elements, but we believe semantic attributes are more relevant to learning the meanings of words.

Future Work

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

Ontology Mapping

As our goal is to build knowledge about objects, properties, and the relations between them, we need to infer logical relations from our data. We have started developing an interface between the perceptual system and an OWL ontology that consists both of meta-knowledge that we store about the feature spaces (the perceptual distance metric, dimensions, and numerical constraints) and of relationships between concepts. We can infer conceptual relations from perceptual data, and can also use conceptual relations to constrain our perceptual mappings. For example, if *turquoise* data points are contained entirely within *blue* data points, we can infer that *turquoise* implies *blue*. The main challenge in implementing this is determining the *contains* relation in a model that does not have explicit decision boundaries.

Canonical Viewpoints

When working in a three dimensional space, we must account for the fact that objects have canonical viewpoints that are associated with their shape. For example, a triangular block may look like a rectangle from the side, but is classified as a triangle because the view from other sides shows a triangular shape. While placing objects on the table eliminates this problem to an extent, it is unrealistic to assume that a child would only see objects in their canonical viewpoint when learning object names. To learn canonical viewpoints, we would downweight frames with conflicting classifications (according to relations in the ontology) rather than averaging equally across frames.

Learning Strategies

Research in child language acquisition has identified a number of strategies and heuristics children use to learn language. We plan to implement the following strategies to improve performance on the existing task and to allow for greater variety in future tasks.

Mutual Exclusion When learning words for objects and properties, children use the learning strategy of *mutual exclusion* - unknown words map to unknown concepts. Jaswal and Hansen (2006) show this bias can be even stronger than pointing and gaze. While our system exhibits this bias in the limit of large amounts to data, we plan to enforce it when dealing with multiple objects (rather than choosing the one that is pointed to) and when updating implicit decision boundaries for perceptual regions in feature spaces.

Shape Bias While we define an object in terms of multiple weighted features using Mahalanobis distance, Landau, Smith, and Jones (1998) show that children tend to assign distinct names to objects based on shape, rather than other features. Although we were able to use certain other features to classify objects, we plan to more heavily weight shape features once we address the issue of canonical viewpoints, as this method should generalize to broader domains more effectively.

Pragmatic Inference and Informative Questions We plan to support a broader range of questions, such as those involving multiple objects or asking about specific properties. We also have begun work on extracting pragmatic information hidden in these questions to improve performance on future tasks. For example, "Where is the red block?" both implies the existence of a red block and at least one other block that is not red. Work towards this goal also opens up many different possibilities for interacting with the system, as joint attention can be inferred through language rather than explicit pointing.

Conclusion

We have shown an initial capability for near one-shot learning of the meaning of words used to describe simple objects in demonstrations. We find that a modified k-nearest neighbor classifier is an effective means of achieving our goal of quickly learning grounded language, and permits further analysis to determine which attributes map to natural language. We also show that language understanding both improves classification results and allows for more natural environments for data collection, opening the way for agents that learn simply by being situated in the environment of our everyday lives.

Unannotated training data requires the system to determine quality examples to learn from. Moving such processing forward in the learning process can greatly reduce the need to saturate a system with training examples. While we show an improvement through parsing to determine intent even with relatively simple descriptions and a controlled environment, we believe this principle can bring a wide range of AI systems closer to near one-shot learning abilities.

Acknowledgements

This work was supported in part by the National Science Foundation (grant IIS-1012205), and the Office of Naval Research (grant N000141110417).

References

Allen, J.; Swift, M.; and de Beaumont, W. 2008. Deep Semantic Analysis of Text. In *Symposium on Semantics in Systems for Text Processing (STEP)*, volume 2008, 343–354. Morristown, NJ, USA: Association for Computational Linguistics.

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

Dalal, N., and Triggs, B. 2005. Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition*.

Farhadi, A.; Endres, I.; Hoiem, D.; and Forsyth, D. 2009. Describing Objects by Their Attributes. *2009 IEEE Conference on Computer Vision and Pattern Recognition* 1778–1785.

Fei-Fei, L.; Fergus, R.; and Perona, P. 2006. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence* 28(4):594–611.

Harnad, S. 1990. The Symbol Grounding Problem. *Physica D: Nonlinear Phenomena*.

Hu, M.-K. 1962. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on* 8(2):179 –187.

Jaswal, V. K., and Hansen, M. B. 2006. Learning Words: Children Disregard Some Pragmatic Information That Conflicts with Mutual Exclusivity. *Developmental science* 9(2):158–65.

Khotanzad, A., and Hong, Y. 1990. Invariant Image Recognition by Zernike Moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12(5):489–497.

Landau, B.; Smith, L.; and Jones, S. 1998. Object Shape, Object Function, and Object Name. *Journal of Memory and Language* 38(1):1–27.

Lowe, D. G. 1999. Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision* 2([8):1150–1157 vol.2.

Mahalanobis, P. 1936. On The Generalized Distance in Statistics. *Proceedings of the National Institute of Sciences of India* 49–55.

Matuszek, C.; FitzGerald, N.; Zettlemoyer, L.; Bo, L.; and Fox, D. 2012. A Joint Model of Language and Perception for Grounded Attribute Learning. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Pangburn, B.; Mathews, R.; and Iyengar, S. 2003. EBLA: A Perceptually Grounded Model of Language Acquisition. In *Proceedings of the HLT-NAACL 2003 Workshop on Learning Word Meaning from Non-Linguistic Data*, 46–53.

Roy, D., and Pentland, A. 2002. Learning words from sights and sounds: A computational model. *Cognitive science* 26(1):113–146.

Yu, C., and Ballard, D. 2004. On the Integration of Grounding Language and Learning Objects. In *Proceedings of the National Conference on Artificial Intelligence*, number Quine, 488–494. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.