Identifying Negative Exemplars in Grounded Language Data Sets

Nisha Pillai, Cynthia Matuszek
npillai1 | cmat @ umbc.edu
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County, Baltimore, Maryland 21250

***Abstract*—There has been substantial work in recent years on grounded language acquisition, in which paired language and sensor data are used to create a model of how linguistic constructs apply to the perceivable world. While powerful, this approach is hindered by the difficulty of obtaining unprompted negative examples of natural language annotations. In this paper, we describe an initial pilot of a system that uses natural language similarity metrics to automatically select negative examples from a paired corpus of perceptual and linguistic data.**

I. Introduction

Semantic representations of complex real-world environments are a powerful tool for supporting inference, action planning, and intuitive robot-human interaction. Obtaining such representations from interactions with non-specialists has significant advantages, allowing robots to learn appropriate language models for the wide range of real world situations, rather than requiring preemptive modeling for different settings. Natural language understanding provides a framework for both natural user interaction and for learning such semantics at the granularity of interest to users.

One promising area of research concerns using unconstrained language paired with sensor and actuator data to support learning about objects efficiently from human interaction. [11, 22] However, building semantic models from unconstrained natural language is challenging. One approach to learning grounded language is to treat it as a *joint learning problem,* in which visual classifiers are learned in conjunction with language models, treating language descriptions as labels for novel visual percepts. [10, 14] However, efficient training of joint models generally relies on both positive and negative training data.

This is a particular challenge in grounded language acquisition because it is unusual for people to provide negative examples without prompting. (For example, one would not normally describe an object as “not yellow.”) Furthermore, a lack of a positive label does not imply a negative grounding; something described as “a carrot” is not a good negative grounding for an “orange” classifier. This problem is well-known to have an effect on human grammar acquisition, [2, 8] parser learning, [5] and grounded lexical acquisition. [15]

In this paper, we describe how natural language document similarity metrics can be used to select appropriate negative examples from a corpus of training data. Our approach is to treat the set of all descriptions of an objects as a document describing that object and then determine the similarity of pairs of documents; objects with dissimilar documents are then treated as negative examples for one another (see Figure 1). Our initial results support the idea that purely linguistic tools can be used to analyze corpora of perceptual training data.

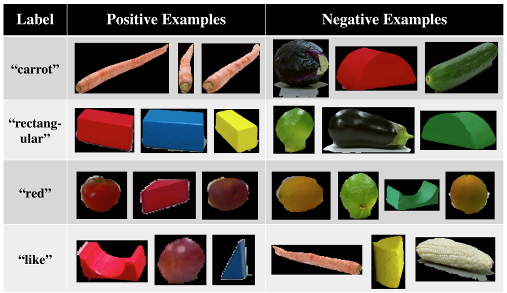

Fig. 1. Positive and negative training data for classifiers denoted by descriptive words, as automatically selected by linguistic analysis of positive object descriptions. Examples are shown of words used for object type, shape, and color; an example of a nonvisual word with poor predictive power, ‘like,’ is also shown.

II. Related Work

Almost all work on learning to understand grounded language relies in some part on learning algorithms that use negative labels as part of learning. The most straightforward approach is to explicitly collect negative labels, [23, 4] possibly through crowdsourcing [21, 6] or gameplaying. [24] However, this may not be applicable to all mechanisms for gathering language. Another possibility is to associate randomly chosen groundings with terms that are not used to describe those images. [19, 3] Because language is not exhaustive, this approach is noisy and may require manual cleanup. [20]

Another practical technique that can be incorporated is to design language collection trials that either use objects that have no shared visual characteristics, [10] or explicitly design trials that exhibit negative characteristics. [18] Our work is most similar to the fully unsupervised label identification of Roy [15], but uses document similarity metrics, rather than term clustering.

In order to choose appropriate language terms for which to train classifiers, we rely on the well-known tf-idf algorithm, which has been used to determine the descriptive power of terms, [17] their relevance to particular documents, [25] and as a document similarity metric. [16] Our selection of negative labels uses the Paragraph Vector algorithm, which learns representations of features from varying length documents. [12, 13] We employ the Distributed Memory Model of Paragraph Vectors (PV-DM) for this work. [9]

III. Background

TF-IDF: tf-idf, short for *term frequency-inverse document frequency,* is a well-studied metric that is reflects how important a word is to a document in a collection or corpus. The tf-idf value *increases* proportionally to the number of times a term appears in the document, which reflects the term's relevance to that document, and *decreases* with the number of documents containing that term, reflecting its discriminative power. Intuitively, if a term such as "cabbage" appears frequently in a document, it is important to that document, but words that appear in many documents, such as "very," have less discriminative power.

In this work, we use the simplest definition of term frequency: $tf(t,d)$ is a raw count of the number of times a term t appears in a document d. Inverse document frequency is the inverse logarithmic fraction of the number of documents that contain the term from the set of all documents, D. This gives the tf-idf value of t for a particular descriptive document d: This gives:

$$tf\text{-}idf(t,d,D) = tf(t,d) \cdot \log \frac{N}{|\{d \in D : t \in d\}|}$$

Where $tf(t,d)$ is the number of times a term t appears in document d, N is the size of the set of documents $N = |D|$, and $|\{d \in D : t \in d\}|$ is the number of documents in which the term t appears.

Paragraph Vector: Paragraph Vector is an unsupervised learning algorithm that maps documents into a fixed-length feature vector that is robust against varying document sizes [9].

In the Paragraph Vector model, paragraphs and every word in these paragraphs are mapped to vectors P, and W respectively. We calculate the un-normalized log-probability vector of P.

$$y = b + Uh$$

Here y_i is un-normalized log-probability of a word in the vector. U and b are softmax parameters, and h is a vector formed by a concatenation of word vectors, W and paragraph vector, P.

Prediction of the 'next word' in the context or 'topic' of the paragraph is achieved using a softmax classifier. Fixed length sliding window is applied to choose contexts. Here, w_1,w_2, ..., w_T are the sequence of words that are getting trained.

$$p(w_t|w_{t-k}, ...w_{t+k}) = \frac{e^{y_{wt}}}{\sum_i e^{y_i}}$$

Model tries to maximize the average log probability,

$$\frac{1}{T}\sum_{t=k}^{T-k} \log p(w_t|w_{t-k}, ...w_{t+k})$$

We consider the output of trained model that is a fixed length dense vector, as in a bag of words model but retains the predictive power of a more semantically informed model. Training is performed using gradient descent with backpropagation. The trained paragraph vector represents the "topic" of a document, and has shown good performance for predicting other terms that may be found in that document. Paragraph Vector maps every document to a point in fixed-dimensional space irrespective of their varying description size; empirically, 100 dimensions gave sufficient representative power.

IV. Approach

We build on previous work on jointly training visual classifiers and interpretations of descriptive language to develop a semantic representation of the visual characteristics of objects. [10, 14]. The focus of this paper is the identification of suitable negative data points for training those visual classifiers, which we do using a two-step approach: first, choosing discriminative terms for which to train classifiers; second, using semantic similarity between descriptions of objects to find dissimilar examples to serve as negative training examples.

More specifically, we treat all of the descriptions of a particular object, concatenated, as a "document" associated with that object. We use tf-idf to find the most important, discriminative terms for a particular document, and attempt to train classifiers associated with those terms; positive examples for classifier learning are the images people described using that term. We choose negative examples for that classifier training by learning a paragraph vector for each document, and using cosine similarity to find the most distant paragraph vectors. Our preliminary results show that this model is sufficient to train color, shape and object classifiers successfully.

A. Data Corpus Collection

Our data set contains 72 objects, divided into 18 classes. (Classes included both food objects, such as 'banana,' 'cabbage,' and 'carrot,' and children's blocks in various shapes, such as cylinders and cuboids.) We took 3-4 RGB-D images of each object from a variety of angles. We extracted RGB features extracted from the color channel and used kernel descriptors [1, 7] to extract shape and object features from the depth channel. Kernel descriptors [1, 7] model size, 3D shape, and depth edge from the depth channel and experiments show that it significantly enhances the quality of object classification results. Figure 2 shows an example image for each class in the data set.

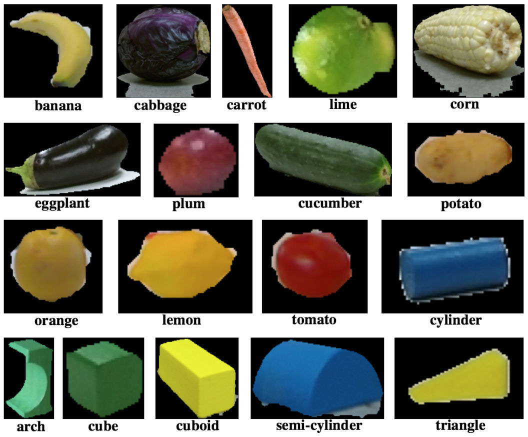

Fig. 2. Sample RGB images for each class in the dataset, as taken with a Kinect2 camera and presented to Mechanical Turk annotators.

To obtain descriptive language, the RGB images were posted on Amazon Mechanical Turk, and users provided short descriptions. A total of 3055 descriptions were collected, an average of 42 descriptions per object. All descriptions of a single object are concatenated into an unordered "document" describing that object. Documents range from 200–450 words, and our corpus contains a total of 19,947 unique words. A short list of stop words is stripped from the documents, and the remaining word are lemmatized as "terms".

B. Positive Token Formation

In order to select tokens for which to learn visual classifiers, we first employ tf-idf to find discriminative terms from the set of descriptive documents. We calculate tf-idf for all terms and pass it through an activation function to learn how important the term is to that document. Empirically, we found that selecting terms with a tf-idf value above a threshold of 10 produced the best values. This gives the most significant labels that are effective representations of object instances for which to create visual classifiers.

golden
cylinder
square
apple
potato
cuboid
yellow
carrot
tomatoe
rom
isgreen
lime
blue
lemon
peach

purple
cherry
wedgeshaped
circle
long
cob
wedge
orange
plantain
white
banana
red
semicircle
head
archshaped

triangle
corn
triangular
mango
tomato
isyellow
cylindrical
cucumber
half
cabbage
ear
arch
brick
building

cababage
potato
cube
rectangular
eggplanet
item
husk
green
rectangle
eggplant
cylindershape
cheese
block

Fig. 3. The set of the top-scoring terms for each object in our data set. 57 terms are shown because there was some overlap across objects. The results reflect errors made by human annotators; for example, 'tomato' and 'tomatoe' are both present.

For each term, all images that have been described using that term become positive examples for training a classifier. From the original 19,947 words used to describe 72 objects, 230 tokens were selected for classifier training (see Figure 3 for some examples of positive tokens selected.) Some examples of words that scored very low in the process include: 'picture', 'colored', 'look', 'color', 'image', 'object', and 'laying'.

C. Negative Instance Selection

Once a set of images has been selected as positive examples, the second step is to find the most semantically distant objects in the data set to serve as negative examples by comparing the objects' descriptive documents. First, the set of objects that share no mutual positive terms is selected. For this we calculate the paragraph vector (described above) for each document. These vectors are then compared using cosine similarity.The greater the angle between vectors, the more semantically dissimilar the documents are. From this similarity matrix, we choose the most dissimilar instances as negative instances (see Figure 5 for an example).

D. Classifier Learning

In order to test the effectiveness of our approach, we trained three different types of classifiers: color, shape, and object type. The first two are suitable for the current problem and have been used in previous work on this topic. [14] In addition to that, type-of-object classifiers demonstrate the possibility of learning more complex concepts.

Because an unsupervised learner has no way of knowing which of these categories a word actually refers to, multiple classifiers must be trained over the positive and negative groundings for each term. We use RGB and RGB-D images based on the objects to extract color, shape and object features, and then apply tf-idf to find positive labels and PV-DM combined with cosine similarity to find dissimilar objects in the language corpus.

We use the most dissimilar objects as negative instances while learning. When the learning system encounters a previously unseen ``label'', it creates visual classifiers named ``label''-color, ``label''-shape, and ``label''-object, which are trained using the color, shape, and object feature type of the positive and negative instance. Objects with the same label are added as additional positive samples to this classifiers. Training is performed using logistic regression.

V. Results

In this section, we discuss the performance of negative sample selection and the quality of the resulting classifiers. In this initial work we present only representative results, rather than a complete analysis of results. For example, Figure 4 shows the selected and rejected terms for one of the objects in our dataset. Nonetheless, we believe that our approach of choosing negative samples exhibits promising performance.

A. Examples of Negative Label Selection

One of the primary contributions of using the instance paragraph vector model is that it addresses a major failing in the common bag-of-words model: it considers the ordering and semantics of words, but still allows vector-space-based comparisons. Figure 5 illustrates the cosine similarity of the same "banana" object from our data set with other objects. Similarity between pairs of documents is

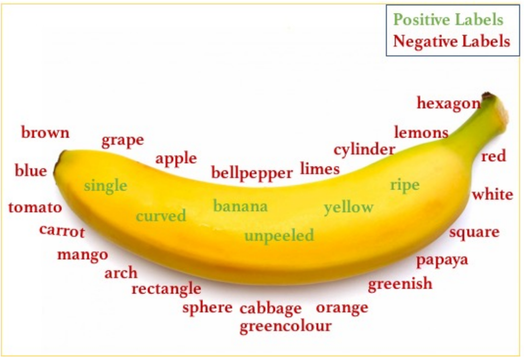

Fig. 4. Positive and negative labels of an object after tf-idf threshold-based filtering. Terms in green will be associated with a classifier that takes this object as a positive example; terms in red, which fell below the tf-idf threshold, will not.

the cosine of angle between the vectors in the vector space. From the figure, we can see that the banana is most similar to another, different banana, which was described as green; then, in descending order, to a yellow cylinder, a lemon, and so on, down to the least similar object, a short red semicylinder. Images of the most distant objects can then be used as negative samples for training the classifier.

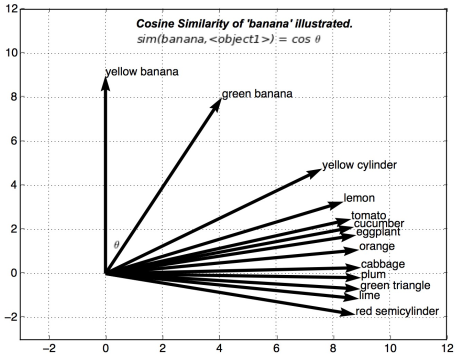

Fig. 5. Cosine similarity of the paragraph vectors of descriptive documents. Vectors represent individual objects, and the angle between vectors denote the similarity of those descriptions. The figure illustrates the similarity of other documents with the descriptive document for one of the bananas in our data set.

Figure 1 shows the selected positive and negative images of some labels. We could see that labels "carrot", "rectangular", and "red" were able to choose perfect negative instances for the learning model. The label "like" is also perfect considering that it does not semantically mean anything in language grounding. Results show that paragraph vector model was able to select good negative samples from the corpus. Instance paragraph vector approach gives good performance as our negative selection model; we evaluate visual classifier performance with these chosen negative samples.

B. *Quality of Trained Classifiers*

The quality of the grounded language model—the learned model of the relationship between language and percepts—is a product of the association between language tokens and the trained visual classifiers. Ideally, attribute descriptions should be associated primarily with a single classifier with good predictive power. Our evaluation was conducted on our corpus of images and descriptions, and classifiers associated with strongly informative terms were trained as described above. Cross validation was used for testing.

Color: Our color classification results show good results on color labels (see Figure 6), although there is some overfitting resulting from the relatively small set of objects. For example, the potato and eggplant objects were frequently described as being on a white background, leading to conflation in classifiers denoted by 'white.' One possible solution for the demand of extensive annotation is using efficient active learning techniques. Previous grounded language acquisition experiments that exercise active learning techniques [14] have shown promising outcomes in reducing annotation efforts without compromising classification accuracy.

The "orange" and "red" classifiers has substantial overlap, in part because users described both tomatoes and carrots using both terms; in addition, polysemy had a negative impact, as the term "orange" can be color or object.

Shape: Training shape classifiers on small RGB-D images is significantly more difficult than color, in part because the shape of an object from different

	Ground truth				
Color classifier denoted by “term”	yellow	red	green	white	orange
“yellow”	0.93	0.20	0.37	0.05	0.02
“building”	0.09	0.11	0.00	0.00	0.17
“red”	0.00	0.89	0.05	0.16	0.35
“green”	0.27	0.00	0.89	0.02	0.00
“tomato”	0.24	0.94	0.00	0.00	0.00
“white”	0.06	0.68	0.55	0.85	0.73
“orange”	0.50	0.93	0.21	0.26	0.66

Fig. 6. Associations between color classifiers created for keywords (y-axis) and ground truth (x-axis). Only a small subset of representative classifiers are shown, since one is created for each keyword in the corpus. The classifiers associated with color words have strong predictive power, as does the color classifier associated with the token “tomato.” The visually uninformative word “building,” by contrast, does not have any strongly performing classifiers associated.

angles can vary considerably. While still performing well, the quality of the results is somewhat less. A few sources of complication included the tendency of annotators not to describe the shape of common objects; cucumbers were frequently referred to as green, but never as cylindrical. In addition, certain terms, such as rectangular, were overused. Figure 7 shows the results of some selected shape classifiers.

	Ground truth				
Shape classifier denoted by “term”	cube	cylinder	sphere	arch	triangle
“cylinder”	0.32	0.87	0.06	0.29	0.29
“rectangular”	0.82	0.43	0.51	0.78	0.30
“circle”	0.25	0.25	0.75	0.26	0.21
“archshaped”	0.29	0.27	0.12	0.82	0.33
“triangle”	0.54	0.60	0.52	0.31	0.82

Fig. 7. The confusion matrix showing performance of shape classifiers for objects in shape categories (x-axis) against selected words (y-axis). The confusion between rectangles and arches is a product of our data set, as the blocks usually described as arch-shaped have a rectangular top.

Object: Object classifiers, which are intended to determine the class an object belongs to, are trained using a combination of color and shape features. While our object classification has good results on our data set, this is partly due to the strong influence of color in classification; both the toys and the food objects in our data set tended to be primarily a single strong color.

	Ground Truth				
Object classifier denoted by “term”	corn	semi-cylinder	banana	eggplant	tomato
“corn”	0.92	0.01	0.77	0.04	0.00
“building”	0.08	0.61	0.30	0.02	0.03
“banana”	0.00	0.15	1.00	0.00	0.04
“tomato”	0.00	0.00	0.05	0.00	0.94
“wedge”	0.49	0.30	0.00	0.43	0.00
“eggplant”	0.26	0.24	0.01	0.84	0.11

Fig. 8. Associations between object classifiers created for keywords (y-axis) and ground truth objects (x-axis). Only a small subset of representative classifiers are shown, since one is created for each keyword in the corpus.

Overall: The overall goal of this work is to allow robots to improve their ability to learn semantic representations of their perceived environments, using unconstrained natural language as the training signal. While not a complete metric, one way of considering whether this work makes progress towards that goal is to verify that the most obvious terms for the intended ground truth have been identified as having important semantic relevance, and how accurately the classifiers associated with those terms perform on the complete dataset. By this metric, we find that all of our ground truth labels have been discovered; classifier performance is shown in Figure 9.

VI. CONCLUSION AND FUTURE WORK

While a number of different approaches have explored how to acquire semantic representations of perceptual data, the need for negative natural language exemplars recurs throughout the literature. Our results demonstrate that using semantic similarity measures on corpora of mixed language and perceptual data can be used to automatically identify terms that should be considered as candidates for learning groundings for, and to select negative examples automatically for training classifiers that instantiate the semantic meaning of perceptual data. The most immediate steps for this work include running a more thorough evaluation of the results by using Mechanical Turk to evaluate the quality of each step of the process: finding positive labels,

blue:	0.995	arch:	0.532
green:	0.947	cube:	0.590
orange:	0.720	cylinder:	0.725
purple:	0.499	rectangle:	0.621
red:	0.844	triangle:	0.649
white:	0.772		
yellow:	0.918		

banana:	0.942	lemon:	0.777
cabbage:	0.879	lime:	0.936
carrot:	0.887	orange:	0.921
corn:	0.922	potato:	0.715
cucumber:	0.615	tomato:	0.926
eggplant:	0.646		

Fig. 9. Average performance of the classifiers associated with each word shown in cross-validation. In general, color classifiers (top left) perform the best; the outlier, purple, reflects the color differences between the objects described as purple (typically eggplants, red cabbage, and plums). Shape classifiers (top right) perform worst, stemming from the fact that people do not provide a shape description as often as the other two classes. Classifiers for object types (bottom left and right) perform well in general.

selecting positive and negative examples for classifiers, and evaluating the behavior of those classifiers on held-out data sets.

One possible baseline to determine the performance and effectiveness of the model will be to compare this negative sample selection model with a traditional model that randomly chooses objects which are not explained by the same keywords. Including questions to explicitly ask about the dissimilarity of objects would also be a valid measurement scenario that can be incorporated in human-robot interaction trials.

In future, our intention is to extend this work to a more varied set of objects, additional kinds of classifiers, and complex visual classification tasks, as well as to apply the identification of negative grounding examples to ongoing work on grounded language acquisition tasks. Ultimately, the goal is to have robot systems that actively learn from descriptions of objects and instructions, from non-specialists, after deployment into complex, novel environments. Being able to learn from natural human descriptions will be an important part of that process, and this work demonstrates initial steps towards solving an outstanding problem in doing so.

References

[1] Liefeng Bo, Kevin Lai, Xiaofeng Ren, and Dieter Fox. Object recognition with hierarchical kernel descriptors. 2011.

[2] Melissa Bowerman. The'no negative evidence'problem: How do children avoid constructing an overly general grammar? In *Explaining language universals*, pages 73–101. Basil Blackwell, 1988.

[3] Grzegorz Chrupała, Lieke Gelderloos, and Afra Alishahi. Representations of language in a model of visually grounded speech signal. 2017.

[4] Haris Dindo and Daniele Zambuto. A probabilistic approach to learning a visually grounded language model through human-robot interaction. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 790–796. IEEE, 2010.

[5] Peter M Hastings and Steven L Lytinen. The ups and downs of lexical acquisition. *Ann Arbor*, 1001:48109, 1994.

[6] Ross A Knepper, Stefanie Tellex, Adrian Li, Nicholas Roy, and Daniela Rus. Recovering from failure by asking for help. 2015.

[7] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. RGB-D object recognition: Features, algorithms, and a large scale benchmark. In Andrea Fossati, Juergen Gall, Helmut Grabner, Xiaofeng Ren, and Kurt Konolige, editors, *Consumer Depth Cameras for Computer Vision: Research Topics and Applications*, pages 167–192. Springer, 2013.

[8] Howard Lasnik. On certain substitutes for negative data. In *Learnability and linguistic theory*, pages 89–105. Springer, 1989.

[9] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196, 2014.

[10] Cynthia Matuszek*, Nicholas FitzGerald*, Luke Zettlemoyer, Liefeng Bo, and Dieter

Fox. A Joint Model of Language and Perception for Grounded Attribute Learning. In *29th International Conference on Machine Learning (ICML)*, Edinburgh, Scotland, June 2013.

[11] Cynthia Matuszek, Liefeng Bo, Luke Zettlemoyer, and Dieter Fox. Learning from unscripted deictic gesture and language for human-robot interactions. In *Proc. of the 28th National Conference on Artificial Intelligence (AAAI)*, March 2014.

[12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR Workshop*, 2013.

[13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[14] Nisha Pillai, Karan K Budhraja, and Cynthia Matuszek. Improving grounded language acquisition efficiency using interactive labeling. In *Robotics: Science and Systems workshop on Model Learning for Human-Robot Communication*, 2016.

[15] Deb K Roy. Learning visually grounded words and syntax for a scene description task. *Computer speech & language*, 16(3):353–385, 2002.

[16] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[17] Gerard Salton and Michael J McGill. Introduction to modern information retrieval. 1986.

[18] Connor Schenck and Dieter Fox. Towards learning to perceive and reason about liquids. In *Proceedings of the International Symposium on Experimental Robotics (ISER)*, Tokyo, Japan, October 2017.

[19] Carina Silberer, Vittorio Ferrari, and Mirella Lapata. Visually grounded meaning representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[20] S. Tellex, T. Kollar, S. Dickerson, M.R. Walter, A. Banerjee, S. Teller, and N. Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the National Conference on Artificial Intelligence*, 2011.

[21] Stefanie Tellex, Ross A Knepper, Adrian Li, Daniela Rus, and Nicholas Roy. Asking for help using inverse semantics.

[22] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. Approaching the symbol grounding problem with probabilistic graphical models. *AI magazine*, 32 (4):64–76, 2011.

[23] Stefanie Tellex, Pratiksha Thaker, Joshua Joseph, and Nicholas Roy. Learning perceptually grounded word meanings from unaligned parallel data. *Machine Learning*, 94(2):151–167, 2014.

[24] Jesse Thomason. 1. continuously improving natural language understanding for robotic systems through semantic parsing, dialog, and multi-modal perception. 2016.

[25] Justin Zobel and Alistair Moffat. Exploring the similarity space. In *ACM SIGIR Forum*, volume 32, pages 18–34. ACM, 1998.