Leveraging Past References for Robust Language Grounding

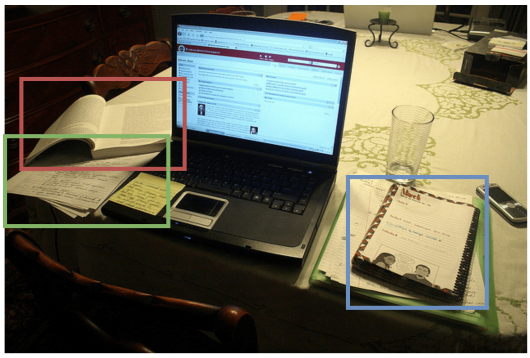
Subhro Roy[*], **Michael Noseworthy**[*], **Rohan Paul, Daehyung Park, Nicholas Roy**
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
{subhro, mnosew, rohanp, daehyung, nickroy}@csail.mit.edu

Abstract

Grounding referring expressions to objects in an environment has traditionally been considered a one-off, ahistorical task. However, in realistic applications of grounding, multiple users will repeatedly refer to the same set of objects. As a result, past referring expressions for objects can provide strong signals for grounding subsequent referring expressions. We therefore reframe the grounding problem from the perspective of coreference detection and propose a neural network that detects when two expressions are referring to the same object. The network combines information from vision and past referring expressions to resolve which object is being referred to. Our experiments show that detecting referring expression coreference is an effective way to ground objects described by subtle visual properties, which standard visual grounding models have difficulty capturing. We also show the ability to detect object coreference allows the grounding model to perform well even when it encounters object categories not seen in the training data.

1 Introduction

Grounding referring expressions to objects in an environment is a key *Artificial Intelligence* challenge spanning computer vision and natural language processing. Past work in referring expression grounding has focused on understanding the ways a human might resolve ambiguity that arises when multiple similar objects are in a scene – for example, by referring to visual properties or spatial relations (Nagaraja et al., 2016; Hu et al., 2017). However, most previous work treats it as an *ahistorical* task: a user is presented with an image and a referring expression and only features from the current referring expression are used to determine which object is being referred to. However, in many scenarios where a grounding system can be deployed, grounding is not an isolated one-off task. Instead, users will repeatedly refer to the same set of objects. In a household environment, a robot equipped with a grounding system will repeatedly be asked to retrieve objects by the people who live there. Similarly, during a cooperative assembly task, a robot will repeatedly be asked for various parts or tools. Even in non-embodied systems, such as conversational agents, a user will repeatedly refer to different relevant entities.



Utterances from a User:
(1) The open text book.
(2) Page with a dark border and images of two people at the bottom.
(3) The open book with typed pages.
(4) Pages of handwritten notes under the open book.

Figure 1: An example where a single user repeatedly refers to objects in the same scene. The color identifies which bounding box is being referred to.

These scenarios present new challenges and opportunities for grounding referential expressions. When people in the same environment commonly interact with each other (as in a household or workplace), *lexical entrainment* will likely occur (Brennan and Clark, 1996). That is, users of the system will come to refer to objects in similar ways to how they have been referred to in the past.

[*] Equal contribution.

Proceedings of the 23rd Conference on Computational Natural Language Learning, pages 430–440
Hong Kong, China, November 3-4, 2019.

Thus, it is important that the grounding system can adapt to the vocabulary used where it is deployed.

Even if each object is being repeatedly referred to by a set of people who do not interact with each other, the agent can still learn general information about the objects by remembering how they have been referred to in the past. This is useful as it provides a way for the model to learn properties of objects that are otherwise hard to detect such as subtle visual properties which cannot easily be captured by standard visual grounding systems.

To include past referring expressions in a grounding model, we formulate grounding as a type of coreference resolution – are a new phrase and a past phrase (known to identify a specific object) referring to the same object? To compare a new referring expression with past referring expressions, we need to learn a type of compatibility metric that tells us whether two expressions can both describe the same object. Detecting object coreference involves distinguishing between mutually exclusive object attributes, recognizing taxonomic relations, and permitting unrelated but possibly co-existing properties.

Our contribution is to demonstrate that grounding accuracy can be improved by incorporating a module that has been trained to perform coreference resolution to previous referring expressions. We introduce a neural network module that learns to identify when two referring expressions describe the same object. By jointly training the system with a visual grounding module, we show how grounding can be improved using information from both linguistic and visual modalities.

We evaluate our model on a dataset where users repeatedly refer to objects in the same scene (see Figure 1). Given the same amount of training data, our coreference grounding model achieves an overall increase of 15% grounding accuracy when compared to a state-of-the-art visual grounding model (Hu et al., 2017). We show that the coreference grounding model can better generalize to object categories and their descriptions not seen during training – a common difficulty of visual grounding models. Finally, we show that jointly training the coreference model with a visual grounding model allows the joint model to use object properties not stated in previous referring expressions. As an example application, we demonstrate how the coreference grounding paradigm can be used with a robotic platform.[1]

2 Technical Approach

The task of grounding referring expressions is to identify which object is being described by a query referring expression, Q. The input to this problem is a set of N objects, $O = \{o_1, o_2, \ldots, o_N\}$.[2] Each object, o_i, is represented by its visual features, v_i (i.e., pixels from the object's bounding box), and a referring expression, r_i, that was previously used to refer to that object (r_i may not always be available). The grounding problem can then be modelled as estimating the distribution over which object is being referred to: $p(x|Q, v_{1:N}, r_{1:N})$ where x is a random variable with domain O.

There has been a lot of work in grounding referring expressions (Hu et al., 2017), many of which use only visual features and no interaction history. Most of the proposed models have the form:

$$p(x = o_i|Q, v_{1:N}) = \mathcal{S}\left(\mathrm{W}_{vis} \cdot f_{vis}\left(Q, v_{1:N}\right)\right)_i = \frac{\exp\left(\mathrm{W}_{vis} \cdot f_{vis}\left(Q, v_i\right)\right)}{\sum_{j=1}^{N} \exp\left(\mathrm{W}_{vis} \cdot f_{vis}\left(Q, v_j\right)\right)}$$

where $f_{vis}\left(Q, v_i\right)$ is a low dimensional representation of the visual features v_i and the query Q, W_{vis} is a learned linear transformation, and $\mathcal{S}(\cdot)_i$ is the ith entry of the softmax function output.

We introduce a similar model for coreference grounding which uses past referring expressions to decide which object o_i is being referred to:

$$p\left(x = o_i|Q, r_{1:N}\right) = \mathcal{S}\left(\mathrm{W}_{coref} \cdot f_{coref}\left(Q, r_{1:N}\right)\right)_i$$

where $f_{coref}\left(Q, r_i\right)$ is an embedding of a past referring expression and the query expression, and W_{coref} is a learned linear transformation.

Finally, we introduce a joint model which fuses representations f_{vis} and f_{coref} by a function g:

$$p\left(x = o_i|Q, v_{1:N}, r_{1:N}\right) = \mathcal{S}\left(g\left(f_{coref}\left(Q, r_{1:N}\right), f_{vis}\left(Q, v_{1:N}\right)\right)\right)_i \quad (1)$$

Note that f_{vis} can come from any visual grounding model that associates text to visual features extracted from the objects' bounding boxes.

[1] The dataset, code, and demonstration videos can be found at https://mike-n-7.github.io/coreference-grounding.html.

[2] Object proposal networks (e.g., Faster R-CNN) can be used to extract object bounding boxes from an image.

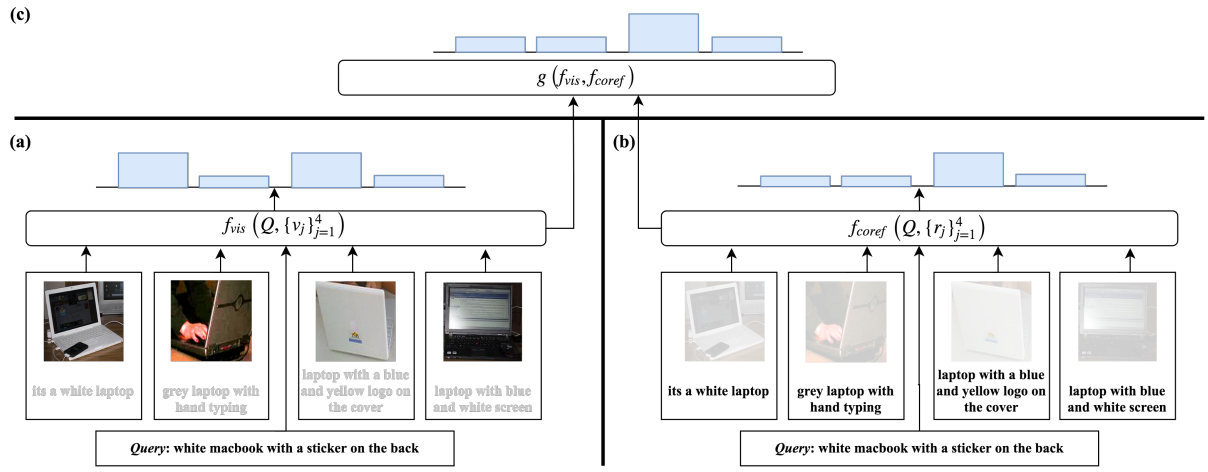


Figure 2: (a) A visual grounding model only takes in images to resolve the query expression and outputs a categorical distribution over the objects. (b) We propose a model that uses past referring expressions to resolve the new expression. (c) These models can be combined to fuse visual and linguistic information.

Our contributions are the coreference and joint grounding models. Both models learn to ground a new referring expression by computing compatibility with past referring expressions of candidate objects. We describe f_{coref} in detail in Section 2.1 and the joint model in Section 2.2. For a visualization of the model, see Figure 2.

2.1 Coreference Grounding Model

Given a query referring expression Q and an expression r_i for each object (each represented by a sequence of M words: $\{w_1, w_2, \ldots, w_M\}$), we define a joint representation of these two expressions, f_{coref}, as follows:

$$f_{coref}(Q, r_i) = f_{enc}(Q) \odot f_{enc}(r_i)$$

where f_{enc} produces referring expression embeddings for the given phrase of dimension $l \times 1$, and $\odot$ is the elementwise multiplication operator. Note the same encoder is used to embed both r_i and Q. In Section 4.3, we evaluate various referring expression embedding methods described below.

LSTM Embeddings The final output state of an LSTM (Hochreiter and Schmidhuber, 1997) is used as the referring expression embedding.

BiLSTM Embeddings For the bidirectional LSTM, forward and backward LSTMs are run over the input sequence, and their outputs for each word are concatenated. An expression embedding is computed by the dimension-wise max across words (Collobert and Weston, 2008).

Attention Embeddings Attention encoders (Lin et al., 2017) learn a weighted average of BiLSTM outputs as the referring expression representation. They output an attention score that is used to compute a weighted average of the BiLSTM outputs.

InferSent Embeddings Recently, *InferSent* (Conneau et al., 2017) was proposed as a *general purpose* sentence embedding method. *InferSent* is similar to the BiLSTM model but was trained using the *Natural Language Inference* task with the intuition that this task would require the sentence embeddings to contain semantically meaningful information. The authors showed that their sentence representation generalized well to other tasks. The pretrained encoder from the *InferSent* model is used to embed a referring expression.

2.2 Integrating Vision and Coreference

The coreference model can learn the complexities of coreference with past expressions, but if certain properties are not mentioned in a previous expression, the model will have difficulty deciding which object is being referred to. If the referenced property could have been learned by a visual grounding model, we would like to include this representation in our model. In this section, we show how we can take an existing visual grounding model and combine it with a coreference grounding model. We generate representations f_{vis} from an existing grounding model, and f_{coref} from our coreference grounding model. These representations are fused using a function $g(\cdot)$, which is used to compute the most likely referred object using Equation 1. We experiment with two choices of g, which we describe in the following subsections.

2.2.1 Addition

One approach is to take the sum of the scores of component visual and coreference grounding models. The function $g(\cdot)$ can be written as (arguments of functions removed for brevity):

$$g = \mathrm{W}_{vis} \cdot f_{vis}(\cdot) + \mathrm{W}_{coref} \cdot f_{coref}(\cdot)$$

where $\mathrm{W}_{vis}, \mathrm{W}_{vis} \in \mathbb{R}^{1 \times l}$ are learned parameters which transform their respective feature vectors of size l into scalar scores.

2.2.2 Concatenation

Simple addition might not capture all interdependencies between different modalities. We propose an approach where we concatenate the representations f_{vis} and f_{coref}, and add a two layer feed forward network to output the final score,

$$g = \mathrm{W}_{c1}(\mathrm{ReLU}(\mathrm{W}_{c2}(\mathrm{ReLU}([f_{vis} f_{coref}]).$$

For our visual grounding model, we use the *Compositional Modular Network* (Hu et al., 2017), which is one of the top performing models in several benchmark referring expressions datasets. We train the joint model end-to-end so that it can learn how to properly merge the visual and coreference information.

3 Dataset

Our goal is to ground a referring expression to an object in the environment using past referring expressions and visual features. Existing referring expression datasets do not contain at least two referring expressions for each object. Since this is a requirement to learn and evaluate models that can utilize past expressions, we create two new datasets for the task – a large *Diagnostic* dataset where past expressions are algorithmically assigned, and a smaller *Episodic* dataset created to capture more realistic interaction scenarios.

3.1 Diagnostic Dataset

In this dataset, artificial scenes are created by grouping similar objects, and each referring expression for an object is collected independently. This form of dataset allows us to easily scale up the amount of data used for training and capture more descriptive language by introducing category-level ambiguity into the scene. We use images from the MSCOCO dataset (Lin et al., 2014), which contains bounding boxes for each object in an image. We randomly group together four object images from the same category. We randomly label one object as the goal object and the remaining three as distractor objects. Annotators from the *Figure Eight* platform[3] are shown these images with the goal object labeled by a red bounding box. They are asked to write an English expression to refer to the goal object so that it can be easily distinguished. Two expressions are collected for each object, each time with different distractor objects.

To ensure the model can distinguish between objects of different categories, we randomly select half the data, and replace two distractor objects in the group with two objects from a different category. Since these objects are from a different category, we expect the referring expression to still be able to correctly identify the goal object.

Each instance, or dataset sample, now consists of four objects (represented by images) with a query expression referring to the goal object. To associate each object with a *past expression*, we use expressions that were used to reference that object in other instances. Each instance now has a set of objects associated with a past expression and an image. In addition, the goal object is labeled and a query referring expression is provided for the goal. We randomly split the data into train, development and test sets in a 60/20/20 ratio. We refer to this split as STANDARD.

To evaluate the ability of grounding models to disambiguate objects from categories not seen during training, we create an alternative split of the *Diagnostic* dataset. This split ensures that no object category in the test set is present in the training or development splits. We refer to this split as HARD. More details of dataset construction and verification are present in the supplementary material.

3.2 Episodic Dataset

The *Diagnostic* dataset uses cropped images of objects from MSCOCO images and programmatically assigned past expressions. In order to capture nuances of realistic interaction, we collect a smaller *Episodic* dataset where annotators are shown a full scene and repeatedly asked to refer to objects within that scene. We select scenes from the MSCOCO dataset which have three to six objects of the same category. We prune ob-

[3] https://www.figure-eight.com/

ject bounding boxes with area less than 5% or over 50% of the image area. Extremely small objects are often not distinguishable, and large bounding boxes often correspond to a cluster of objects rather than a single object in cluttered scenes.

Each annotator is shown the same scene 10 times in a row. Each time one of the ambiguous objects is marked with a red bounding box. The annotator is asked to provide an English referring expression to uniquely identify the marked object. Our interface does not allow the user to view referring expressions once it has been entered. As a result, if the same object is marked again in the series, the user will have no way to look up what they had written before. This process simulates how a user will refer to objects in the same environment over a period of time. As the annotators do not communicate with each other, the dataset will not capture between-user entrainment. For each image, we collect two such series of 10 expressions from two different users. We create examples for the *Episodic* dataset in two ways:

SAME USER: For each scene and annotator, we order the expressions in the same order they were provided by the annotator. Each expression forms an example where it is the respective query expression and the candidate objects are all the objects in the scene. All previous referring expressions for this scene and annotator are assigned to the respective objects as past referring expressions. These examples capture the scenario where the interaction history is provided by a single user.

ACROSS USERS: Similar to the SAME USER dataset, we create a new example each time the annotator refers to an object. However, the past expressions come from the other annotator who was displayed the same scene. These examples represent cases where interaction history is acquired from people who do not interact with each other.

We create train and development sets with both types of examples. For testing, we create one set corresponding to each of the previously mentioned types. Validation details are in the supplementary material.

The statistics for both datasets are given in Table 1. We report a metric called *Lexical Overlap* to denote the extent of similarity between training and test data. The *Lexical Overlap* is the fraction of word types in the test set that also appear in the training set. As seen in Table 1, the HARD split has lower Lexical Overlap compared to STANDARD.

Dataset	Diagnostic		Episodic	
	STANDARD	HARD	SAME USER	ACROSS USERS
# Train	10133	9855	2656	2656
# Dev	3889	4170	714	714
# Test	3796	3793	1686	1686
Objects per Example	4.0	4.0	5.64	5.64
Expressions per Object	0.47	0.47	0.5	1.94
Lexical Overlap	0.71	0.63	0.47	0.47

Table 1: Statistics for the various datasets used. Unless otherwise mentioned, the numbers are reported from the test set. Low lexical overlap for HARD indicates more unseen words in the test set. The ACROSS USERS test set of Episodic dataset has more past expressions for each object compared to other splits.

This means that a greater number of novel words appear in the HARD dataset.

4 Experiments

We run multiple experiments to evaluate the proposed grounding models and characterize the advantages of using coreference along with visual features for grounding. Specifically, we consider the following questions:

1. Which method performs best for coreference grounding and the joint model? (Section 4.3)
2. How does grounding with different types of information (visual, coreference) transfer to new object categories? (Section 4.4)
3. How does grounding performance vary when past expressions are acquired from the same (or entrained) users, as opposed to users unknown to each other? (Section 4.5)

All quantitative evaluation can be found in Table 2. As our datasets contain examples where the object being referred to may or may not have a previous referring expression, we report overall test set accuracy (*All*), as well as accuracy grouped by the number of past referring expressions belonging to the ground truth object (*0, 1, or, 2*). This allows us to more accurately evaluate the coreference models as they are not expected to perform well without a previous referring expression.

We show how coreference grounding is used in practice by demonstrating its use on the Baxter robot. Namely, we show how a system can keep track of past referring expressions and associate them with objects.

4.1 Model Descriptions

We compare against the following unsupervised coreference baselines (i.e., not trained on the referring expression task). All these methods use a

similarity score between the input expression and the past referring expression to make a prediction.

Word Overlap Baseline: Compute the *Jaccard* similarity between two expressions.
Word Averaging Baseline: Each expression is represented by the average of the word vectors of constituent words. Similarity is computed as cosine similarity between vectors.
Paragram Phrase: Uses the paraphrase model proposed by Wieting et al. (2016) to compute similarity between past and input expressions.
InferSent Unsupervised: Each expression is represented by its pretrained *InferSent* embedding and similarity is computed as cosine similarity.

We also consider supervised models for the referring expression grounding task:

Vision: The *Compositional Modular Network* (Hu et al., 2017) which grounds an input expression to a bounding box's visual features.
Coreference: The proposed model that grounds an expression to objects represented by previous referring expressions. We evaluate the LSTM, BiLSTM-Max, Attention, and InferSent Encoders.
Joint: Jointly trains the *Vision* and *Coreference* components of the model. This model uses only the InferSent encoder. We evaluate both addition and concatenation methods for information fusion.

4.2 Implementation Details

Since the *Episodic* dataset is much smaller in size, all models (except joint with concatenation) are first trained on the *Diagnostic* training set until the validation error stops decreasing, and then trained on the *Episodic* train set. For the joint model with concatenation, we found that tuning only the fusion parameters, while holding the others fixed, on the *Episodic* data performs better.

The large *Diagnostic* dataset contains at most one past expression for each object. We cannot expect our models trained on the *Diagnostic* data to handle multiple past referring expressions. As a result, when an object is associated with multiple past expressions, we only consider the expression most similar to the query expression according to the Unsupervised InferSent model.

The models are implemented in *PyTorch* (Paszke et al., 2017). All models for the *Diagnostic* dataset are trained with the *Adam* optimizer (Kingma and Ba, 2015), and the best model is selected based on the performance on the development set. We use GloVe vectors (Pennington et al., 2014) and a pretrained VGG19 model to extract visual features (Simonyan and Zisserman, 2015).

4.3 Coreference and Joint Model Evaluation

We evaluate the performance of the coreference and joint models on the STANDARD split of the *Diagnostic* dataset (see column STANDARD of Table 2). First, we observe that all unsupervised coreference methods perform poorly when the goal object does not have past referring expressions. However, when past expressions are present, all the coreference baselines outperform the vision model. Coreference using pretrained InferSent embeddings performs the best among the unsupervised methods, possibly because InferSent embeddings have been pretrained on the SNLI dataset. SNLI was created from image captions, making the domain similar to that of referring expressions.

Learned models of coreference start performing better on cases where the goal object has no past referring expression. Note that in the 0-column, the 0 only refers to the ground-truth object not having a referring expression – the other objects may have expressions associated with them. Thus the supervised models can better determine when two expressions are incompatible, leading to the model choosing an object without any previous referring expression. However, when a past referring expression is present, they are typically informative and the *Word Overlap* model performs the best amongst unsupervised and supervised methods (see the 1-column). *InferSent (Unsupervised)* still achieves strong performance yet fine-tuning to the task still helps. We use the coreference model with the *InferSent* encoder for the joint models.

The jointly trained models achieve high accuracies in both cases where the ground truth object has previously been referred to (1-column) and those where it has not (0-column). This indicates that the models can successfully utilize information from both vision and coreference modalities. The concatenation fusion method outperform simple shallow addition of component scores.[4]

The joint model consistently outperforms the vision model. This is because people usually provide information relevant to how the object can be referred. If this information is available, which is

[4]Note that different objects in a scene might have different number of past expressions. At test time, we will not know the goal object, and hence, we cannot use the number of past expressions to determine which model to use at test time.

		Diagnostic						Episodic (*s. 4.5*)				
		STANDARD (*s. 4.3*)			HARD (*s. 4.4*)			SAME USER				ACROSS USERS
		0 57%	1 43%	All	0 56%	1 44%	All	0 49%	1 33%	2 15%	All	
	Vision	64.2	66.1	65.0	59.6	60.6	60.0	32.1	32.1	29.4	31.5	31.5
Unsupervised	Word Overlap	3.5	74.2	34.1	4.0	72.6	34.3	5.3	85.4	87.3	37.6	58.1
	Word Averaging	3.5	69.6	32.1	4.0	67.8	32.2	5.3	77.8	80.2	42.7	51.8
	Paragram Phrase	3.5	71.3	32.8	4.0	71.3	33.7	5.3	81.1	83.7	44.3	56.9
	InferSent	8.8	73.8	36.9	9.2	73.6	37.6	5.9	85.0	85.7	46.3	47.7
Supervised	LSTM	28.2	41.1	33.8	28.2	40.3	33.5	6.0	69.4	63.9	37.2	46.8
	BiLSTM-Max	30.0	60.8	43.3	27.0	57.8	40.6	7.0	75.7	74.6	41.8	53.1
	Attention	29.2	61.6	43.2	26.8	56.0	39.7	10.1	67.0	63.9	38.4	47.4
	InferSent	27.9	70.5	46.3	28.8	68.6	46.4	5.8	82.5	85.3	45.4	61.3
Joint	Addition	65.3	69.4	67.1	62.0	62.3	62.1	22.3	63.8	60.3	42.6	48.7
	Concatenation	68.6	71.3	**69.8**	58.0	70.7	**63.6**	19.0	84.7	86.1	**52.7**	**62.7**

Table 2: Grounding accuracy under different conditions. Best scores in bold. The columns labeled **0**, **1**, and **2** correspond to test examples where the goal object has 0, 1 and 2 past expressions respectively (percentage indicates fraction of test examples applicable for that column). **All** refers to the score on the entire test set. Most examples in ACROSS USERS have large number of past expressions, so we only report score on the entire test set.

true in many realistic scenarios, it is beneficial to utilize coreference grounding.

With more data, the vision model could achieve higher performance. However, we argue that as ambiguity between objects increases, the properties that distinguish objects become more subtle and a large dataset would be necessary to learn these intricacies.

When does Joint perform better than Vision and Coreference? Our *Joint* model performs better than models trained on single modality (*Vision*, *Coreference*). The joint model can use visual features when the past referring expressions are not sufficient to discriminate between objects. In 25% of examples, the *Coreference* model predicts the wrong object whereas the *Joint* model selects the correct object. A majority of these cases are examples where the goal object has no past expression associated with it. In 8% of examples, the *Joint* and *Coreference* models are correct even though the *Vision* model is wrong. Finally, for only 7% of the examples, the *Joint* model predicts the correct object when both the *Vision* and *Coreference* models predict incorrectly. This indicates that the *Joint* model is primarily merging the gains of the *Vision* and *Coreference* models; most correct decisions of *Joint* correspond to correct decisions either from *Vision* or *Coreference*. Figure 3 shows a breakdown of how often the various models outperform each other. Figure 4 shows examples where Joint outperforms models trained on a single modality.

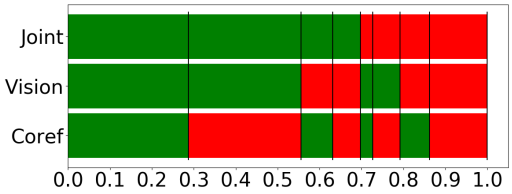


Figure 3: Proportion of examples of the STANDARD test set where different subsets of the systems ground correctly (green) or incorrectly (red). Instances are arranged along the x-axis.

4.4 Generalizing to New Object Categories

To evaluate how the various models handle generalization to unseen object categories, we use the HARD split of the *Diagnostic* dataset. The test set of this split contains object categories which have not explicitly been seen during training. We hypothesize that due to pretrained word embeddings (which include embeddings for words describing the unknown categories), the coreference models will be able to successfully ground to new object categories. On the other hand, the performance of the *Vision* model will decrease in the HARD split, because the pretrained visual features of the new objects are not well aligned with representations of unseen words. As seen in Table 2, this is indeed the case as the *Vision* model's performance drops between the STANDARD and HARD datasets. On the other hand, the coreference models' performance on the HARD split is comparable to those

of the STANDARD split. Although the aggregate performance is low, the coreference models perform strongly on examples where the goal object has past expressions (see 1-column). In particular, *Coref Supervised with InferSent* achieves 70.5% on the STANDARD split, which reduces only to 68.6% on the HARD split. This can be explained by observing that the representation of the object (its past referring expression) and a new referring expression are already aligned within the same vector space (pretrained InferSent embeddings).

4.5 Performance on Episodic Dataset

As the *Diagnostic* dataset is somewhat artificial in the way objects and properties are grouped, we also evaluate these models on the *Episodic* dataset. As described in Section 3.2, the key differences in this dataset are that all candidate objects in an example are from the *same* scene, and previous referring expressions are added sequentially as the user refers to more objects in the scene (thus a previous referring expression truly did occur *previously*).

We find similar trends in the performance on the *Episodic* dataset (see the *Episodic* columns of Table 2). Since the same user is referring to the same object multiple times, we expect expressions for the same object to be similar. This leads to the high performance of coreference models in the SAME USER split of the *Episodic* dataset (see the 1-column). Even the word-overlap model does particularly well due to the correlation between expressions from the same user (over 85% accuracy when the goal object has past expressions).

If the past expressions were not provided by the same user, but by users unknown to each other, we can still see improvement over not having any past expressions (see the ACROSS USERS column). In this split, past expressions are always associated with objects as we assume other users have interacted with the system. In the ACROSS USERS split, even though the *Coreference* models are less effective than in the SAME USER split, they still outperform the *Vision* model (e.g., the *Supervised InferSent* model achieves 61.3%, whereas a pure vision model achieves only 31.5%). This is because users unknown to each other still provide useful knowledge about the properties of objects.

The *Joint* models benefit from using both modalities, which is particularly important in realistic scenarios as most objects initially will not have any previous referring expressions. Note

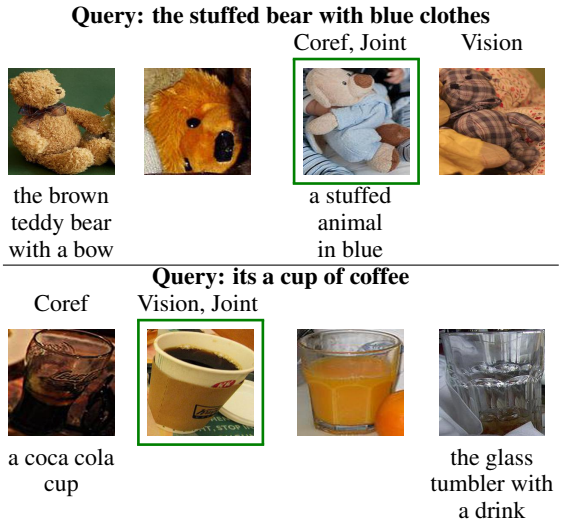


Figure 4: Examples where the Joint model accurately grounds the query, but either Vision or Coreference does not. The past expression is mentioned below each object image. The correct object has a green border and a model's name appears above its predicted object.

that the performance of the *Vision* model on the *Episodic* dataset is much lower than its performance on the *Diagnostic* dataset. This loss in performance is due to the images in the *Episodic* dataset being more cluttered and annotators often using spatial relationships to refer. As the *Diagnostic* training dataset does not contain spatial information, our models cannot handle these cases. We can train on existing referring expression datasets to learn spatial relationships, but we do not explore this as it is not central to this paper. More analysis is added to the appendix.

4.6 Robot Demonstration

One of the motivations of this work was to enable robots to use past referring expressions to aid grounding in human-robot interactions. In this section, we provide a demonstration of how our coreference grounding system can be integrated with the Baxter robotic platform. To benefit from coreference grounding, the system must have a natural way to keep track of past referring expressions of an object. We demonstrate such an interface in Figure 5.

We focus on a scenario where a human asks the robot to pick up specific objects. If the robot incorrectly identifies the object being referred to, on the next turn, the user can correct the robot by indicating the correct object (Figure 5b). The referring expression can then be associated with the correct

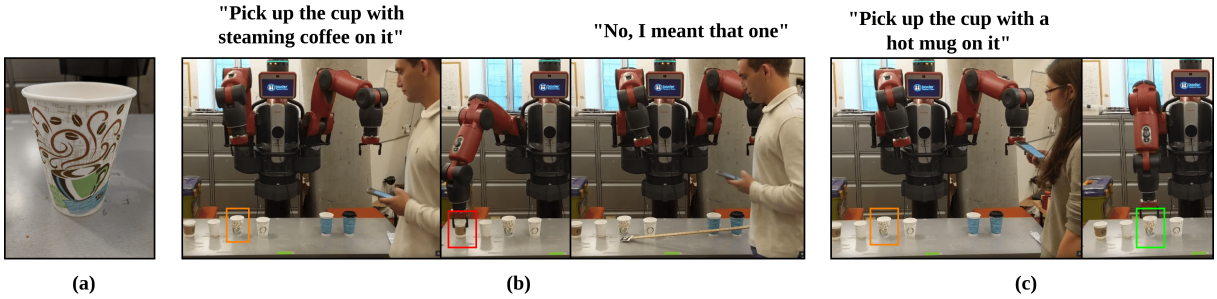


Figure 5: Demonstration of a coreference grounding system on the Baxter robot. (a) The cup being identified. (b) A user refers to the cup and the robot grounds incorrectly. The user corrects the robot and associates the referring expression with that object. (c) When a new user refers to the same object, the system's output is correct.

object and this association can be maintained with a tracking system. Future users (with no knowledge of the first user) can then successfully refer to the object (Figure 5c).

5 Related Work

Grounding Referring Expressions There has been a lot of recent work on resolving referring expressions to objects (Mao et al., 2016; Yu et al., 2016; Shridhar and Hsu, 2018; Nagaraja et al., 2016; Cirik et al., 2018). In contrast to these approaches which are static once trained, our model leverages past referring expressions, which permits an interface to add information during execution. Our task is related to *Visual Dialogue* (Das et al., 2017; Kottur et al., 2018), where an agent interactively answers questions about visual properties of a scene where the answers only exist in the image and not in the dialogue context. In contrast, we focus on using complementary knowledge from past referring expressions. There has also been work to learn a compatibility metric dictating whether two words can refer to the same object (Kruszewski and Baroni, 2015). Our work focuses on coreference between entire referring expressions instead of atomic words.

Lexical Choice in Interactions A large body of work in psycholinguistics (Clark and Wilkes-Gibbs, 1986; Brennan and Clark, 1996; Pickering and Garrod, 2004) show that participants in a conversation collaboratively come up with lexical terms to refer to objects and consequently, get *entrained* with each other and start using similar expressions to refer to objects. These works form the motivation for our problem formulation. Recently, the *PhotoBook Dataset* has been proposed to investigate shared dialogue history in conversation (Haber et al., 2019). The dataset differs from ours in that expressions refer to entire scenes instead of objects within a scene. The authors' conclusions support our findings that using past expressions is useful for resolving referring expressions.

Interaction History for Human Robot Interaction Paul et al. (2017) maintains knowledge provided by users, using a closed set of predicates. In constrast, we use raw past referring expressions to handle an open domain of knowledge. There has been work on grounding in dialogue for robots (Tellex et al., 2014; Whitney et al., 2017; Thomason et al., 2017; Padmakumar et al., 2017). In contrast to our work, they focus on refinement or clarification, and hence past expressions only help within the same dialogue episode.

6 Conclusion

In this work, we reformulated the grounding problem as a type of coreference resolution, allowing for the inclusion of past referring expressions that are typical in many real-world scenarios. We proposed a model that can use both linguistic features from past expressions and visual features of the object to ground to a new expression. We showed that this model outperforms a purely vision-based model as it can use past descriptions of salient features that the vision-based model may have difficulty learning with limited data. It further benefits from having a vision model that can fill in information not provided in past expressions.

Acknowledgements

We gratefully acknowledge funding support in part by the Honda Research Institute and the Toyota Research Institute. However, this article solely reflects the opinions and conclusions of its authors.

References

Susan E. Brennan and Herbert H. Clark. 1996. Conceptual pacts and lexical choice in conversation. In *Journal of Experimental Psychology: Learning, Memory, and Cognition*.

Volkan Cirik, Taylor Berg-Kirkpatrick, and Louis-Philippe Morency. 2018. Using syntax to ground referring expressions in natural images. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*.

Herbert H. Clark and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22(1):1–39.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M.F. Moura, Devi Parikh, and Dhruv Batra. 2017. Visual Dialog. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*.

Janosch Haber, Tim Baumgärtner, Ece Takmaz, Lieke Gelderloos, Elia Bruni, and Raquel Fernández. 2019. The photobook dataset: Building common ground through visually-grounded dialogue. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. 2017. Modeling relationships in referential expressions with compositional modular networks. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Satwik Kottur, José M. F. Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. 2018. Visual coreference resolution in visual dialog using neural module networks. In *Proceedings of the 15th European Conference on Computer Vision (ECCV)*.

Germán Kruszewski and Marco Baroni. 2015. So similar and yet incompatible: Toward the automated identification of semantically compatible words. In *HLT-NAACL*.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Proceedings of the 13th European Conference on Computer Vision (ECCV)*.

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*.

Varun K. Nagaraja, Vlad I. Morariu, and Larry S. Davis. 2016. Modeling context between objects for referring expression understanding. In *Proceedings of the 14th European Conference on Computer Vision (ECCV)*.

Aishwarya Padmakumar, Jesse Thomason, and Raymond J. Mooney. 2017. Integrated learning of dialog strategies and semantic parsing. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

Rohan Paul, Andrei Barbu, Sue Felshin, Boris Katz, and Nicholas Roy. 2017. Temporal grounding graphs for language understanding with accrued visual-linguistic context. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Martin J. Pickering and Simon Garrod. 2004. Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, 27(2):169–190.

Mohit Shridhar and David Hsu. 2018. Interactive visual grounding of referring expressions for human-robot interaction. In *Proceedings of Robotics: Science and Systems (RSS)*.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Stefanie Tellex, Ross Knepper, Adrian Li, Daniela Rus, and Nicholas Roy. 2014. Asking for Help Using Inverse Semantics. In *Proceedings of Robotics: Science and Systems (RSS)*.

Jesse Thomason, Aishwarya Padmakumar, Jivko Sinapov, Justin Hart, Peter Stone, and Raymond J. Mooney. 2017. Opportunistic active learning for grounding natural language descriptions. In *Proceedings of the 1st Conference on Robot Learning (CoRL)*.

David Whitney, Eric Rosen, James MacGlashan, Lawson Wong, and Stefanie Tellex. 2017. Reducing Errors in Object-Fetching Interactions through Social Feedback. In *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*.

Licheng Yu, Patric Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. 2016. Modeling context in referring expressions. In *Proceedings of the 14th European Conference on Computer Vision (ECCV)*.