Leveraging Past References for Robust Language Grounding: Supplementary Material

In the Supplementary Material, we include a detailed description of dataset creation (Section 1) and further analysis of the coreference models (Section 2).

1 Dataset Construction Details

Here, we provide details about the diagnostic dataset creation. We use images from the MSCOCO dataset (Lin et al., 2014), which contains bounding boxes for each object in an image. We consider object categories related to inanimate objects (52 categories), resulting in a total of $48,000$ unique object images. We split these images randomly into train, development, and test sets in a 60/20/20 ratio (maintaining this ratio for each category). For each of the train and development sets, we perform the following steps:

1. 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[1] 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 from the remaining three distractor objects. We create a separate annotation task to check the quality of the data[2]. To ensure that every object has two associated referring expressions, each object is used as a goal object twice (each time with a different set of distractor objects). See Figure 1a.

2. 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.

3. The third step is associating objects with past referring expressions. Each object is randomly assigned one of the expressions used to reference it in other examples (see Figure 1b). Each instance now has a set of objects, each 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 object (see Figure 1c).

4. For the test set, we remove each past expression with a probability of 0.5. The train and development set still have past expressions for all objects. In order to train a robust model, we use dropout of past expressions during training.

For the rest of the paper, 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. We randomly split object categories into train, development and test sets using a 60/20/20 split. We then repeat the above four steps. The resulting test set does not contain any object category seen in the training data, making this split a challenging test for generalization. We refer to this split as HARD.

We check the quality of the *Episodic* dataset annotations with the same method as for the *Diagnostic* dataset described in the footnote.

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

[2] To check the quality of the data, we create a separate task where annotators are shown the 4 objects and the referring expression collected in the first annotation task. If 2 out of 3 annotators fail to identify the correct object, we remove the referring expression from our dataset

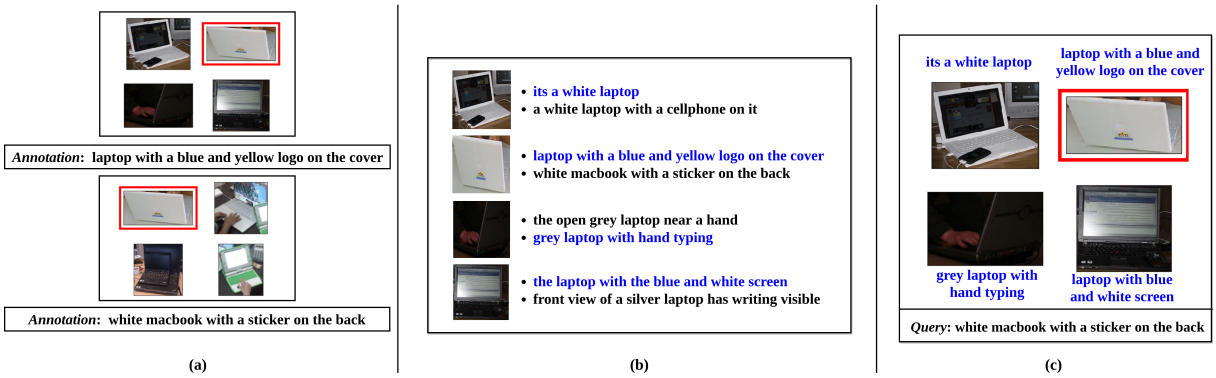

Figure 1: The diagnostic dataset creation process. (a) Annotators provide a referring expression for the object highlighted in red that distinguishes it from the remaining objects of the same category. (b) Each object is annotated twice (with different object sets) resulting in two annotations for each object. (c) Instances are built by selecting previous referring expressions to represent each object (blue text).

Query: white motorcycle with black trimming
(1) [No past expressions]
(2) its a red motorcycle in the street [InferSent]
(3) the police motorcycle with no lights on [Coreference]
(4) motorcycle with a yellow runner and another blue one

Query: the silver car next to a black gate
(1) the white vehicle
(2) the large blue car with brake lights on [InferSent]
(3) [No past expressions] (Coreference)
(4) the rear of a blue car with a man in a red shirt walking behind it

Query: a light brown teddy bear wearing nothing but a smile
(1) [No past expressions]
(2) happy teddy bear [Coreference]
(3) toy bear with yellow and red raincoat in the center
(4) the teddy bear with a blue and white striped shirt [InferSent]

Table 1: Examples where the learned Coreference model (with InferSent encoder) predicts the correct object (green), but the unsupervised InferSent method does not. Each example mentions the query, as well as the past expression associated with each object. Images of objects are not shown for brevity. Predictions of both models are also provided after the corresponding past expression.

2 Analysis

When is Coreference better than InferSent Unsupervised? Our Coreference model is trained using pretrained InferSent embeddings. Although InferSent Unsupervised performs well, we achieve improvements by training on task-specific data. Table 1 shows a few examples where the learned coreference model outperforms the unsupervised InferSent model. We found that around 70% of the gains were from examples in which the goal object did not have any past expressions associated with them. In all these examples, the coreference model needs to decide that the past expressions of the non-goal objects are incompatible with the query expression. We find that learning is important to detect this incompatibility and leads to a significant improvement on these examples over the Unsupervised InferSent method.

Error Analysis About half of the errors of the *Joint* model are examples where the goal object is not associated with any past expressions. In the errors where the goal object has past expressions, we often found them to be unrelated to the query expression. For example, the goal object past expression *"the car by the parking meter"* is unrelated to the query expression *"front view of a white car"*. In these cases, the model has to rely only on visual features for grounding the expression, which can be challenging when subtle or uncommon visual properties are referenced. We also found that for around 18% of examples, Joint predicts the wrong object even when one of Vision or Coreference is correct. This indicates that the fusion of knowledge from vision and past expressions is challenging, and there is room for improvement to better utilize the multiple modalities.

References

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)*.