Supplementary Material: Zero-Shot Grounding of Objects from Natural Language Queries

Arka Sadhu[1] Kan Chen[2†] Ram Nevatia[1]
[1]University of Southern California [2]Facebook Inc.
{asadhu|nevatia}@usc.edu kanchen18@fb.com

In this supplementary document, we present some of the details which couldn't be fit in the main paper. We provide details on how the datasets are sampled (Section 1) from Flickr30k Entities [9], Visual Genome [5] and their distributions (Section 2). We also provide (i) proposal recall of baseline method (Section 3) (ii) image blind and language blind ablation of our model (Section 4).

1. Dataset Construction

We re-use the notation introduced in Table 1 of the main paper. We use Flickr30k for Case 0, 1 and Visual Genome for Case 2, 3 (reasons detailed in 1.4).

1.1. Case 0: $Q \notin W$

This is sampled from **Flickr30k Entities** [9].

In Flickr30k Entities each image has 5 associated sentences. The noun phrases (query-phrases) in each sentence are annotated with the bounding box information. Note that query-phrases in different sentences could refer to the same bounding box. Finally, each bounding box has an associated "entity" which we exploit in Case1.

For Case0, we consider the last word in the query phrase and use the lemmatized representation obtained from spacy [3]. This means that words like "car" and "cars" would be considered the same. However, this doesn't consider synonyms so "automobile" and "cars" are considered different.

We sort the lemmatized words in descending order of frequency and consider the $topI = 1000$ words to be always seen. This is reasonable for words like "woman", "sky" *etc*.

Of the remaining words we do a 70:30 split and consider the first part to be in the include (seen) list (S) and the rest to be in the exclude (unseen) list (U). Note that even though S, U are disjoint they could share few similar words. The resulting include list contains $7k$ words and the exclude list contains $3k$ words.

For the test set we use only those images whose annotations have query word $Q \in U$. For the training set we consider the remaining image and remove annotations which have query word $Q \in U$. We also ensure that there is no overlap between the train and test images. The resulting split is called **Flickr-Split-0**.

The main motivation behind **Case0 Novel Words (NW)** is to see how well our model can perform without explicitly having seen the word during training.

1.2. Case 1: $A \notin C$

This is also sampled from **Flickr30k Entities** [9]. We use the entity information of each noun phrase (query-phrase) provided in the dataset. The entities provided are "people", "clothing", "bodyparts", "animals", "vehicles", "instruments", "scene" and "other". "Other" is used to denote those phrases which cannot be categorized into one of the remaining entities.

We extract out all the images with at least one phrase belonging to the "other" category. Of these, we randomly sample 50% and use them for testing. Of the remaining images, we remove the annotations with the "other" category and use them for training.

The main motivation behind **Case1** is to see how well the model generalizes to novel object categories.

1.3. Case 2, 3: $\exists B$ objects semantically close to A

The two cases share the same training images but different test images. We sample the images and queries from the Visual Genome dataset [5]. The dataset creation process has three major components: (i) cleaning the annotations to make them consistent (ii) clustering the objects and creating the train/test splits to satisfy the dataset properties of Case 2, 3 (iii) balancing the resulting splits.

Cleaning Visual Genome Annotations: In visual genome each image has an average of 200 phrases. A phrase refers to a single object but may contain multiple objects in it. Consider the phrase"man holding a pizza"; it is not directly specified if the referred object is a "man" or a "pizza" but there will be a bounding box in the image corresponding to the referred object, let us call it phrase BB; we need to infer the synset for this phrase BB. In addition, for each image, there are also annotated bounding boxes for each object

[†]This work was done while the author was at USC.

type that appears in any of the phrases; in our example, there would be annotations for "man", "pizza" and other objects that may appear in other phrases. To identify the synset for a phrase BB, we find the object bounding box that it has the maximum IoU with and use the object label associated with that bounding box.

Another difficulty is that if the same object instance is referred to in different phrases, it will have a different phrase BB associated with it. For consistency, we choose one and apply to all phrases. In implementation, we apply a non-maxima suppression algorithm (we use the code provided in [7]); even though, there are no scores associated with the boxes, the algorithm selects on among highly overlapping alternatives. This step provides us with a consistent set of annotations.

Even though the resulting annotations are consistent, the annotations are still spatially imprecise. Due to this reason, we recommend measuring detection accuracy with with IoU threshold of 0.3 instead of the more common value of 0.5.

Clustering Objects: Once we have a clean and consistent set of annotations, we sort all the objects (nearly $5k$ objects) by the number of appearances in the image. However, the objects at the tail end of the distribution are very infrequent so we consider only the top $1k$ objects. Few of these don't have a corresponding word embedding (not available in spacy [3]) so we discard them. This results in a total of 902 objects.

Next, we cluster the GloVe [8] word embeddings of the objects using K-Means clustering (with $K = 20$). We sort the objects in each cluster in descending order with respect to their frequency. For a particular cluster k, we consider the first half to be "seen" (S_k) and the other half to be "unseen" (U_k). This gives us a total of 445 seen objects and 457 unseen objects. For a given cluster k we consider all the images which have at least one object $o_i \in U_k$ to be test images. If there is another object in the same image o_j such that $o_j \in S_k$, we put this image query pair into Case3 else into Case2.

For the remaining images, we remove annotations for any object $o_i \in \cup_k U_k$ and ensure there is at-least one object $o_i \in \cup_k S_k$ and use these to form the training set. However, by construction the training set turns out to be imbalanced with respect to clusters.

Balancing the Dataset To address the above issue we use the following balancing strategy:

- We use Zipf's law approximation that $freq \times rank \approx C$. That is as the rank of the cluster increases the number of annotations for that cluster decreases in a hyperbolic way. We use this to calculate an approximate mean of the clusters. Finally, we also consider $2 \times min\_cluster\_freq$ and take the max of the two.
- Thus, we have an approximate threshold at which we would like to sample. If for a particular cluster this threshold is more than the number of annotations in that cluster, we leave that cluster as it is, else we randomly sample $n = threshold$ annotations for each cluster.
- Note that balancing is only done with respect to the clusters and not with respect to the object names.

Using this balancing strategy we get a balanced train set. We use 25% of it for validation. For test sets we keep both balanced and unbalanced sets.

The main motivation for **Case2, 3** is to see how well the model generalizes to novel objects even if it depends on the semantic distance of the "seen" objects and if it can disambiguate the novel objects from the "seen" objects.

1.4. Choice of Datasets

We note that Flickr30k Entities doesn't provide synset information which is important to disambiguate synonym cases hence it cannot be used for Case2, 3. Visual Genome doesn't contain wide categories like "vehicles" hence it cannot be used for Case 1. For Case0, we could use Visual Genome as well, however, we choose Flickr30k Entities due to its precise bounding boxes.

2. Dataset Distributions

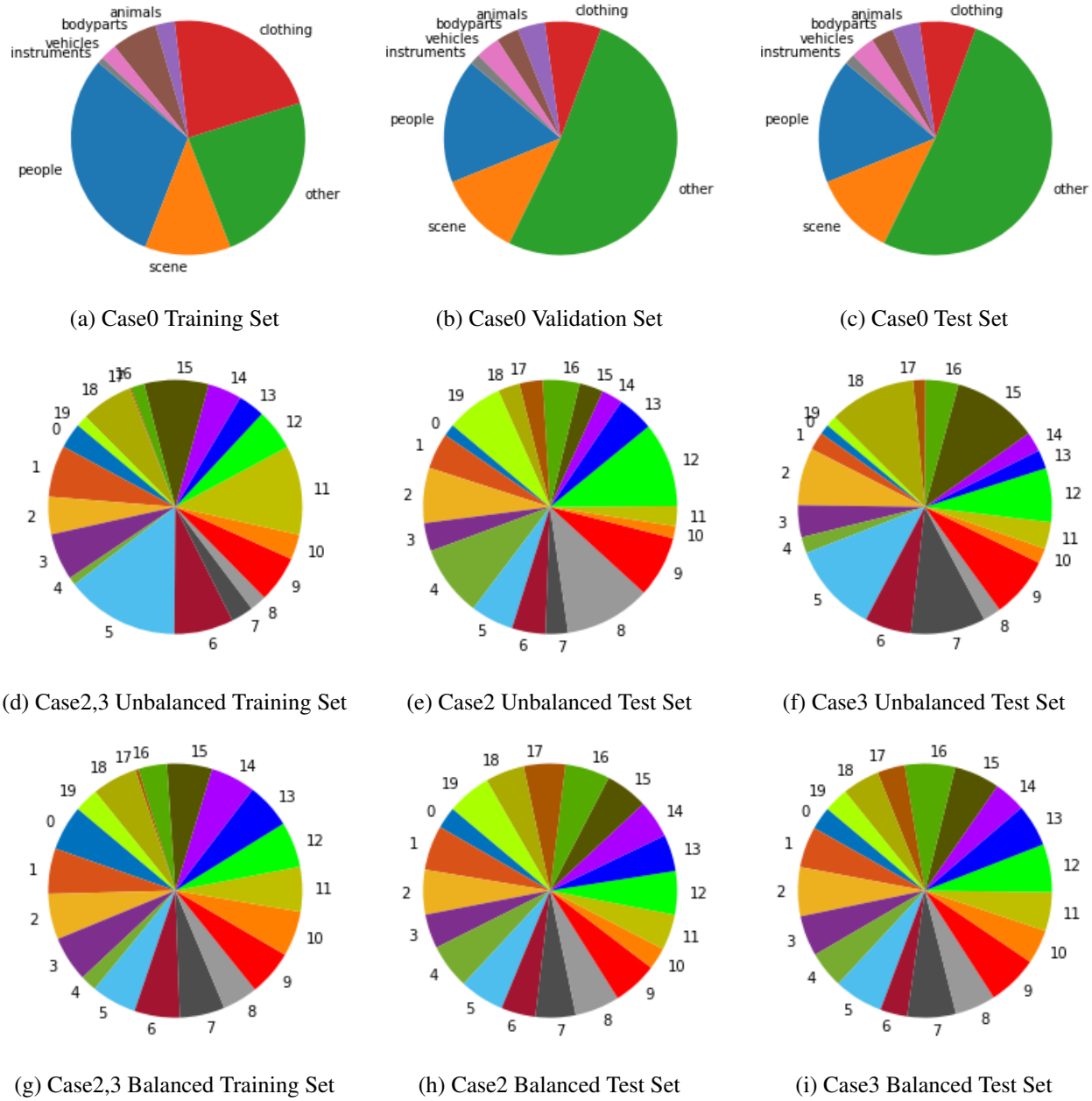
We provide statistics for each dataset in Fig 1. For Case0 we show the entity-wise distribution (a),(b),(c). In particular we note that the "other" category occupies a larger set in the validation and test sets. This is because the "other" category has a more diverse vocabulary and encompasses a larger part of the exclude vocabulary list. For Case1, since it only has "other" category in its validation and test set, the entity-wise distributions are not meaningful and we don't include them here.

For Case2,3 we show the distributions with respect to the clusters formed via K-Means for both the unbalanced [(d),(e),(f)] and balanced cases [(g), (h), (i)]. We don't train on the unbalanced set but do test on the unbalanced set as well. Note that the distribution across clusters in the balanced sets are uniform which means our balancing strategy was successful.

3. Proposals from Pre-Trained Detector(s)

A crucial difference between **ZSGNet** and prior work is the removal of proposals obtained from a pre-trained network. To explicitly analyze the the errors caused due to missing proposals we calculate the proposal recall.

Proposal Recall: We measure the recall rates (@300) of the region proposal network (RPN) from FasterRCNN [12] pretrained on Pascal VOC [2] and fine-tuned on the target



(a) Case0 Training Set (b) Case0 Validation Set (c) Case0 Test Set

(d) Case2,3 Unbalanced Training Set (e) Case2 Unbalanced Test Set (f) Case3 Unbalanced Test Set

(g) Case2,3 Balanced Training Set (h) Case2 Balanced Test Set (i) Case3 Balanced Test Set

Figure 1. Category-wise distribution of various unseen splits. First row: training, validation and test set splits for Case 0; second row: unbalanced training and test sets for Case2 and Case 3; third row: balanced training and test sets for Case 2 and Case 3. In a row, the colors represent the same entities or the same clusters.

dataset in Table 1. For ReferIt [4] we use the fine-tuned model on Flickr30k Entities [13] to be consistent with QRC [1]. We note that (i) proposal recall significantly improves when we fine-tune on the target dataset (ii) performance of QRG on Flickr30k, case0, case1 follows the same trend as the proposal recall (iii) proposal recall is significantly smaller on Visual Genome [5] due to (a) a large number of classes in visual genome (b) considering the "unseen" classes during training as negatives. These recall scores motivate the use of dense proposals for zero-shot grounding.

4. Image Blind and Language Blind Ablations

Model Ablations: We ablate our model in two settings: language blind (**LB**) (the model sees only the image and not the query) and image blind (**IB**) (the model considers the query but not the image). We provide the results ob-

	Flickr30k	ReferIt	Flickr case0	Flickr case1	VG 2B	VG 3B
FR (no f/t)	73.4	25.4	64.95	62.9	15.87	13.92
FR (f/t)	90.85	58.35	85.18	74.85	26.17	25.07

Table 1. Proposal Recall Rates using top-300 proposals at $IoU = 0.5$ (0.3 for VG) calculated on test sets. FR: FasterRCNN [40], no f/t: pretrained on pascal voc, f/t: fine-tuned on the target training set. For referit we use f/t model on Flickr30k to be consistent with QRC.

Model	Flickr30k	ReferIt	Flickr case0	Flickr case1	VG 2B	VG 3B
LB	0.008	0.0042	0.009	0.0024	0.0084	0.0093
IB	28.07	24.75	24.42	17.15	9.5	9.27

Table 2. Ablation study: Language Blind (**LB**) and Image Blind (**IB**) setting using Images of Resolution 300×300. Metric reported is Accuracy@IoU=0.5 (0.3 for VG)

tained after retraining the model in Table 2. In the **LB** case, our model sees multiple correct solutions for the same image and therefore gives a random box output leading to a very low accuracy across all datasets. In the **IB** case, our model learns to always predict a box in the center. We note that the referred object lies in the center of the image for Flickr30k and ReferIt. This is because Flickr30k Entities contains queries derived from captions which refer to the central part of the image and ReferIt is a two player game with a high chance of referring to the central object, leading to relatively high accuracy $25 - 30\%$.

However, this is substantially lower for Visual Genome $(9 - 10\%)$ which has denser object annotations.

5. Inference Time

Since our proposed model ZSGNet uses dense proposals and doesn't perform RoI pooling, our model is highly efficient in computation like other single-shot architectures [6, 11, 10]. Our overall speed is 30ms for 600×600 image on a single Titan X.

References

[1] K. Chen, R. Kovvuri, and R. Nevatia. Query-guided regression network with context policy for phrase grounding. In *ICCV*, 2017. 3

[2] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 2

[3] M. Honnibal and I. Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 2017. 1, 2

[4] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg. Referit game: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014. 3

[5] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017. 1, 3

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016. 4

[7] F. Massa and R. Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. https://github.com/facebookresearch/maskrcnn-benchmark, 2018. Accessed: [Insert date here]. 2

[8] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 2

[9] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015. 1

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 4

[11] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. *CVPR*, 2017. 4

[12] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2

[13] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2:67–78, 2014. 3