Cap2Det: Learning to Amplify Weak Caption Supervision for Object Detection

Keren Ye*[1], Mingda Zhang[1], Adriana Kovashka[1], Wei Li†[2], Danfeng Qin[2], and Jesse Berent[2]

[1]Department of Computer Science, University of Pittsburgh, Pittsburgh PA, USA
[2]Google Research, Zurich, Switzerland
{yekeren, mzhang, kovashka}@cs.pitt.edu lwthucs@gmail.com {qind, jberent}@google.com

In this document, we include more information and statistics about our Cap2Det model. We also provide additional quantitative and qualitative experimental results.

We first show more statistics about our Cap2Det model. We provide per-class precision and recall of our label inference module (Sec. 3.1 in the paper) in Sec. 1 to help better understand how the module affects the final detection performance. We find that some of the categories benefit from the module more than others. We also qualitatively analyze two strong baseline methods in Sec. 2.

Next, we provide additional details of our implementation of the Online Instance Classifier Refinement (OICR) module (Sec. 3.2.2 in the paper) in Sec. 3. Generally speaking, ours is a simplified version without re-weighing the OICR loss as compared to Tang *et al*. [31].

We show more statistics about our learned models. For our model learned from COCO captions (Sec. 4.2 in the paper), we show more metrics computed by the COCO evaluation server in Sec. 4. In Sec. 5, we measure our model learned from Pascal VOC labels (Sec. 4.4 in the paper), using the Correct Localization scores. This evaluation is also proceeded in [30,31,36].

Finally, to enable a more intuitive understanding of our model, we provide more qualitative results of the models learned from both COCO and Flickr30K (Sec. 4.2 in our paper). For the model learned from COCO (Sec. 6), we side-by-side compare our EM+TEXTCLSF method to the EXACTMATCH baseline. For the model learned from Flickr30K (Sec. 7), we, in addition, show the predicted image-level pseudo label. Both results explain the benefits of our proposed idea of amplifying weak caption supervision.

*Work partially done during an internship at Google
†Now at Facebook Inc.

1. Analysis of per-class precision/recall of label inference method

We provide per-class precision/recall of our label inference method (Sec. 3.1) to see how this method affects the detection performance. We still use the 5,000 COCO *val* examples, but evaluate on only the overlapped classes (20 classes) between COCO and Pascal. We show the comparison between our label inference method EM+TEXTCLSF and the lexical matching method EXACTMATCH.

Fig.1 shows the comparison. Our method does not affect the precision too much, but it has a positive impact on the recall. As compared to the Tab. 1 of our paper, not every percentage of improvements of the recall in the text inference leads to an increase of performance in object detection. However, there are some notable classes that can be explained by the merits of text inference model. For example, using our method, the the recall rate of "cow", "horse", "person", are increased by 47.2%, 26.0%, 54.2% respectively (cow from 53% to 78%, horse from 73% to 92%, person from 24% to 37%). Their detection mAP, accordingly, are increased by 24.9%, 16.5%, 62.5% (cow from 49.0% to 61.2%, horse from 44.2% to 51.5%, person from 10.4% to 16.9%).

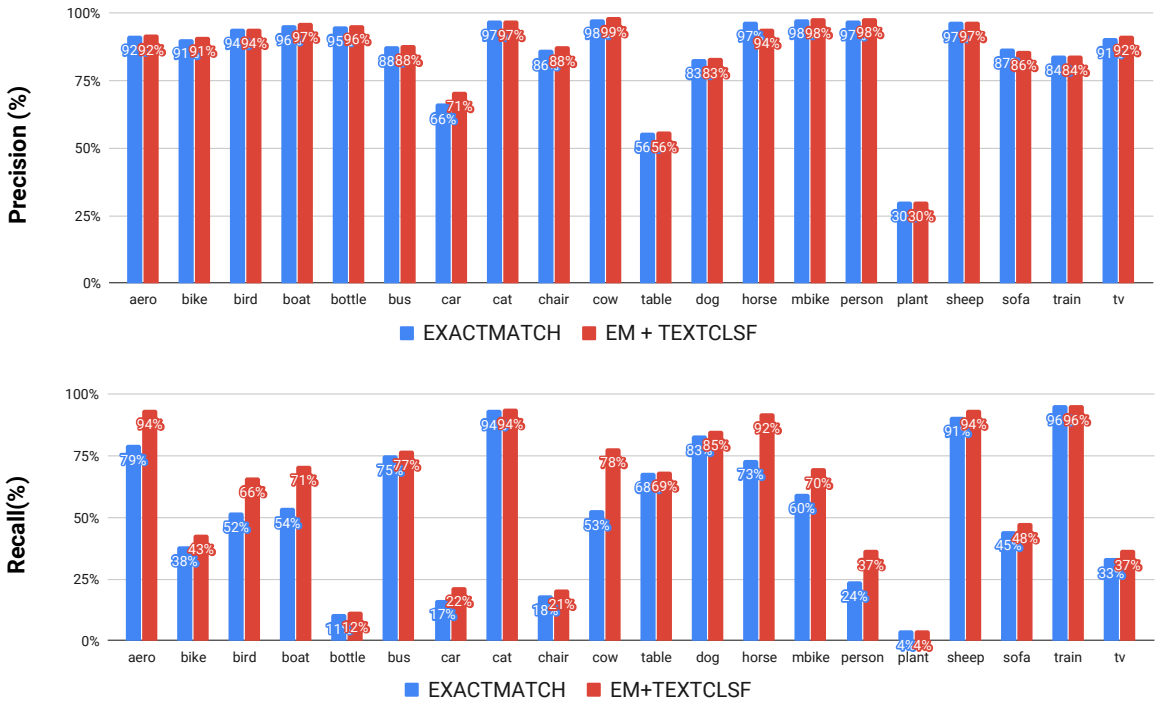
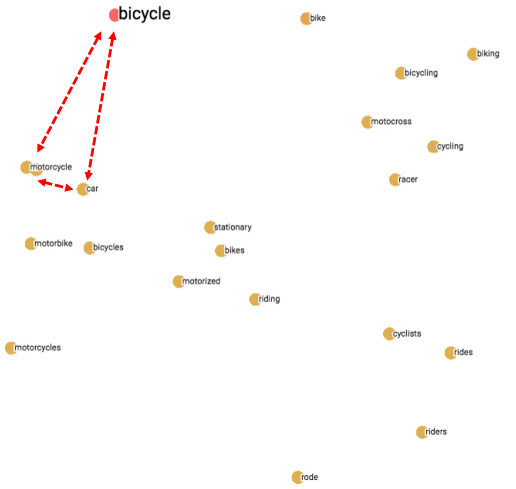


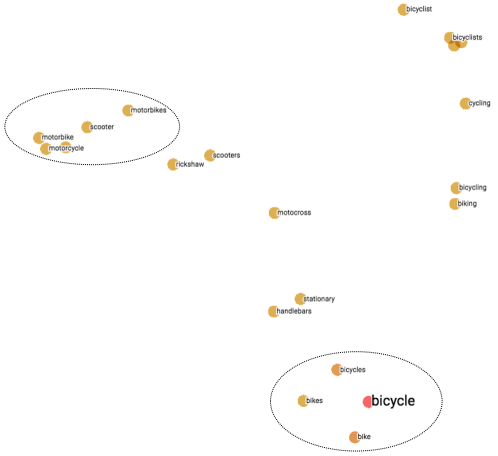
Figure 1: **Precision/recall of the Pascal labels.** Similar to the Fig. 3 in our paper we evaluate the precision and recall, but we focus on the subset of the 20 Pascal VOC classes instead of the performance on the 80 COCO labels.

2. Qualitative analysis of the word embedding based methods

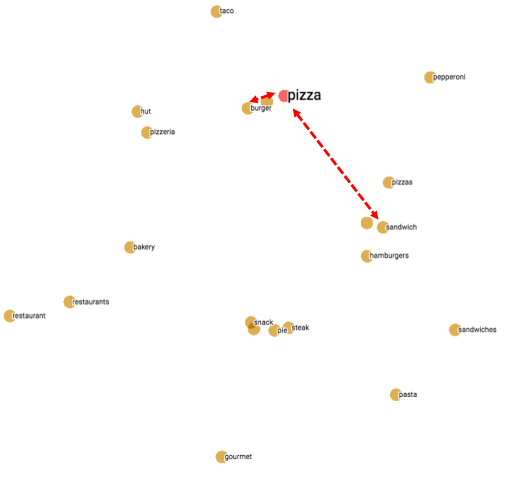
We show some qualitative analysis of the GLOVEPSEUDO and LEARNEDGLOVE, by visualizing their word embedding feature space. Intuitively, Fig.2 shows that the Glove embedding optimized on the general purpose textual corpus is unable to distinguish the nuance such as bicycle and motorcycle, pizza and sandwich, etc. This explains the improved performance of LEARNEDGLOVE compared to GLOVEPSEUDO (Tab. 1 in the paper).



(a) "bicycle" in Glove space: visually different objects such as "car" and "motorcycle" are near to "bicycle".



(b) "bicycle" in Learned Glove space: visually similar objects are clustered.



(c) "pizza" in Glove space: visually different object "burger" is almost indistinguishable.

(d) "pizza" in Learned Glove space: "pizza" related words such as "pepperoni", "toppings" are clustered.

Figure 2: **Visualization of the embedding space for two strong baselines GLOVEPSEUDO and LEARNEDGLOVE**. We show the 20 nearest neighbors to the query word "bicycle" and "pizza". Both (a) and (c) visualize the original Glove feature space while (b) and (d) visualize the learned Glove embedding.

3. Online instance classifier refinement

We provide more details regarding our Sec. 3.2.2, namely, the Online Instance Classifier Refinement (OICR). Generally speaking, the instance-level label $\hat{\boldsymbol{y}}^{(k+1)}$ at the $(k+1)$-th iteration is inferred from both the image level label $[y_1, \ldots, y_C]$ and the detection score $\hat{\boldsymbol{s}}^{(k)}$ at k-th iteration using Algorithm 1. Then, it is used to guide the learning of the $\hat{\boldsymbol{s}}^{(k+1)}$ using Eq.5 in the paper.

Algorithm 1: Online Instance Classifier Refinement - Generating Pseudo Instance Level Labels at the $(k+1)$-th Iteration

Input : Proposals $[r_1, \ldots, r_m]$;
Image level labels $[y_1, \ldots, y_C]$;
Detection scores at the k-th iteration $\hat{\boldsymbol{s}}^{(k)} = [s^{(k)}_{1,1}, \ldots, s^{(k)}_{m,C+1}]$;
Iou threshold $threshold$.

Output: Instance level labels at the $(k+1)$-th iteration $\hat{\boldsymbol{y}}^{(k+1)} = [\hat{y}^{(k+1)}_{1,1}, \ldots, \hat{y}^{(k+1)}_{m,(C+1)}]$.

$\hat{\boldsymbol{y}}^{(k+1)} \leftarrow \vec{\mathbf{0}}$;
for $c' \leftarrow 1$ ***to*** C **do**
  if $y_{c'} = 1$ **then**
    $j \leftarrow \arg\max_i s^{(k)}_{i,c'}$;
    for $i \leftarrow 1$ ***to*** m **do**
      if $IoU(r_i, r_j) > threshold$ **then**
        $\hat{y}^{(k+1)}_{i,c'} \leftarrow 1$; `// Assign foreground target.`
for $i \leftarrow 1$ ***to*** m **do**
  $t \leftarrow \sum_c \hat{y}^{(k+1)}_{i,c}$;
  if $t = 0$ **then**
    $t \leftarrow 1$;
    $\hat{y}^{(k+1)}_{i,C+1} \leftarrow 1$; `// Assign background target.`
  for $c \leftarrow 1$ ***to*** C **do**
    $\hat{y}^{(k+1)}_{i,c} \leftarrow \hat{y}^{(k+1)}_{i,c}/t$; `// Make the probabilities sum to 1.`
return $\hat{\boldsymbol{y}}^{(k+1)}$

4. Using captions as supervision

We provide more metrics to measure our model on the COCO dataset, including the Average Recall (AR) given different numbers of detected boxes (max=1,10,100), and the Average Recall Across Scales (size=small, median, large). Please check the COCO object detection challenge to see the details of these standard metrics. The observations are still similar to the discussion of the Sec. 4.2 of our paper. The following table shows the result.

Methods	Avg. Precision, IoU			Avg. Precision, Area			Avg. Recall, #Dets			Avg. Recall, Area		
	0.5:0.95	0.5	0.75	S	M	L	1	10	100	S	M	L
GT-LABEL	10.6	23.4	8.7	3.2	12.1	18.1	13.6	20.9	21.4	4.5	23.1	39.3
EXACTMATCH (EM)	8.9	19.7	7.1	2.3	10.1	16.3	**12.6**	**19.3**	**19.8**	3.4	20.3	37.4
EM + GLOVEPSEUDO	8.6	19.0	6.9	2.2	10.0	16.0	12.2	18.7	18.9	2.9	19.0	37.6
EM + LEARNEDGLOVE	8.9	19.7	7.2	2.5	10.4	**16.6**	12.3	19.1	19.6	**3.5**	20.0	37.7
EM + EXTENDVOCAB	8.8	19.4	7.1	2.3	10.5	16.1	12.1	19.0	19.5	3.4	20.3	37.5
EM + TEXTCLSF	**9.1**	**20.2**	**7.3**	**2.6**	**10.8**	**16.6**	12.5	**19.3**	**19.8**	**3.5**	**20.6**	**37.8**

Table 1: **COCO test-dev results (learning from COCO captions)**. We report these numbers by submitting to the COCO evaluation server. The best method is shown in **bold**.

5. Using image labels as supervision

Similar to [30,31,36], we also report the Correct Localization (CorLoc) scores (in %) of our method, using the Pascal VOC *trainval* set. We employ the same threshold of IoU ($>= 0.5$) as that in Tab.3 of our paper. The results are shown in the following table.

Methods	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
VOC 2007 results:																					
OICR VGG16[31]	81.7	80.4	48.7	49.5	32.8	81.7	**85.4**	40.1	40.6	79.5	35.7	33.7	60.5	88.8	21.8	57.9	**76.3**	59.9	75.3	81.4	60.6
PCL-OB-G VGG16[30]	79.6	**85.5**	62.2	47.9	37.0	**83.8**	83.4	43.0	38.3	**80.1**	50.6	30.9	57.8	**90.8**	27.0	**58.2**	75.3	68.5	**75.7**	78.9	62.7
TS^2C[36]	**84.2**	74.1	61.3	**52.1**	32.1	76.7	82.9	66.6	42.3	70.6	39.5	57.0	61.2	88.4	9.3	54.6	72.2	60.0	65.0	70.3	61.0
OICR Ens.+FRCNN[31]	*85.8*	82.7	62.8	45.2	43.5	*84.8*	*87.0*	46.8	15.7	*82.2*	51.0	45.6	83.7	91.2	22.2	*59.7*	75.3	65.1	76.8	78.1	64.3
PCL-OB-G Ens.+FRCNN[30]	83.8	85.1	65.5	43.1	*50.8*	83.2	85.3	59.3	28.5	*82.2*	57.4	50.7	*85.0*	*92.0*	27.9	54.2	72.2	65.9	*77.6*	82.1	66.6
Ours	82.4	64.6	**70.0**	50.3	**46.7**	77.4	78.7	**78.0**	**56.6**	77.3	**69.5**	**66.7**	**69.0**	81.2	**33.3**	49.8	76.0	**70.3**	70.9	**86.3**	**67.8**
VOC 2012 results:																					
OICR VGG16[31]	86.2	**84.2**	68.7	**55.4**	46.5	82.8	74.9	32.2	46.7	**82.8**	42.9	41.0	68.1	89.6	9.2	53.9	81.0	52.9	59.5	83.2	62.1
PCL-OB-G VGG16[30]	77.2	83.0	62.1	55.0	**49.3**	83.0	**75.8**	37.7	43.2	81.6	46.8	42.9	**73.3**	**90.3**	**21.4**	**56.7**	**84.4**	55.0	62.9	82.5	63.2
TS^2C[36]	79.1	83.9	64.6	50.6	37.8	**87.4**	74.0	74.1	40.4	80.6	42.6	**53.6**	66.5	88.8	18.8	54.9	80.4	60.4	**70.7**	79.3	64.4
OICR Ens.+FRCNN[31]	*89.3*	86.3	75.2	*57.9*	53.5	84.0	*79.5*	35.2	47.2	*87.4*	43.4	43.8	77.0	91.0	10.4	60.7	*86.8*	55.7	62.0	84.7	65.6
PCL-OB-G Ens.+FRCNN[30]	86.7	*86.7*	74.8	56.8	*53.8*	84.2	80.1	42.0	36.4	86.7	46.5	*54.1*	*87.0*	*92.7*	*24.6*	*62.0*	86.2	63.2	*70.9*	84.2	*68.0*
Ours	**87.9**	70.1	**76.6**	54.7	48.9	80.8	72.8	**76.5**	**51.9**	69.6	**64.7**	49.8	63.7	83.1	21.1	55.2	80.9	**75.5**	62.3	**85.9**	**66.6**

Table 2: **Correct localization (in %) on the Pascal VOC trainval set**. The top shows VOC 2007 and the bottom shows VOC 2012 results. The best single model is in **bold**, and best ensemble in *italics*.

6. Training with COCO captions: qualitative examples

We provide more qualitative examples on the COCO *val* set. We compare the EXACTMATCH and our EM+TEXTCLSF (see paper Sec. 4.2 for details) in a side-by-side manner in the following figure. Qualitatively, our proposed method EM+TEXTCLSF provides better detection results than the baseline EXACTMATCH.

Figure 3: **Visualization of our Cap2Det model results**. We show boxes with confidence scores $> 5\%$. Green boxes denote correct detection results ($IoU > 0.5$) while red boxes indicate incorrect ones.

7. Training with Flickr30K captions: qualitative examples

We provide qualitative examples on the Flickr30K dataset in the following figure. We show the pseudo labels predicted by our label inference module. As a comparison, the EXACTMATCH fails to recall most of the image-level labels, while image-level supervisions are still accurate if we use the method in Sec. 3.1 (EM+TEXTCLSF). Please note that there is NO image-level label on the Flickr30K dataset and our inference module purely transfers textual knowledge from the COCO, with NO training on Flickr30K.

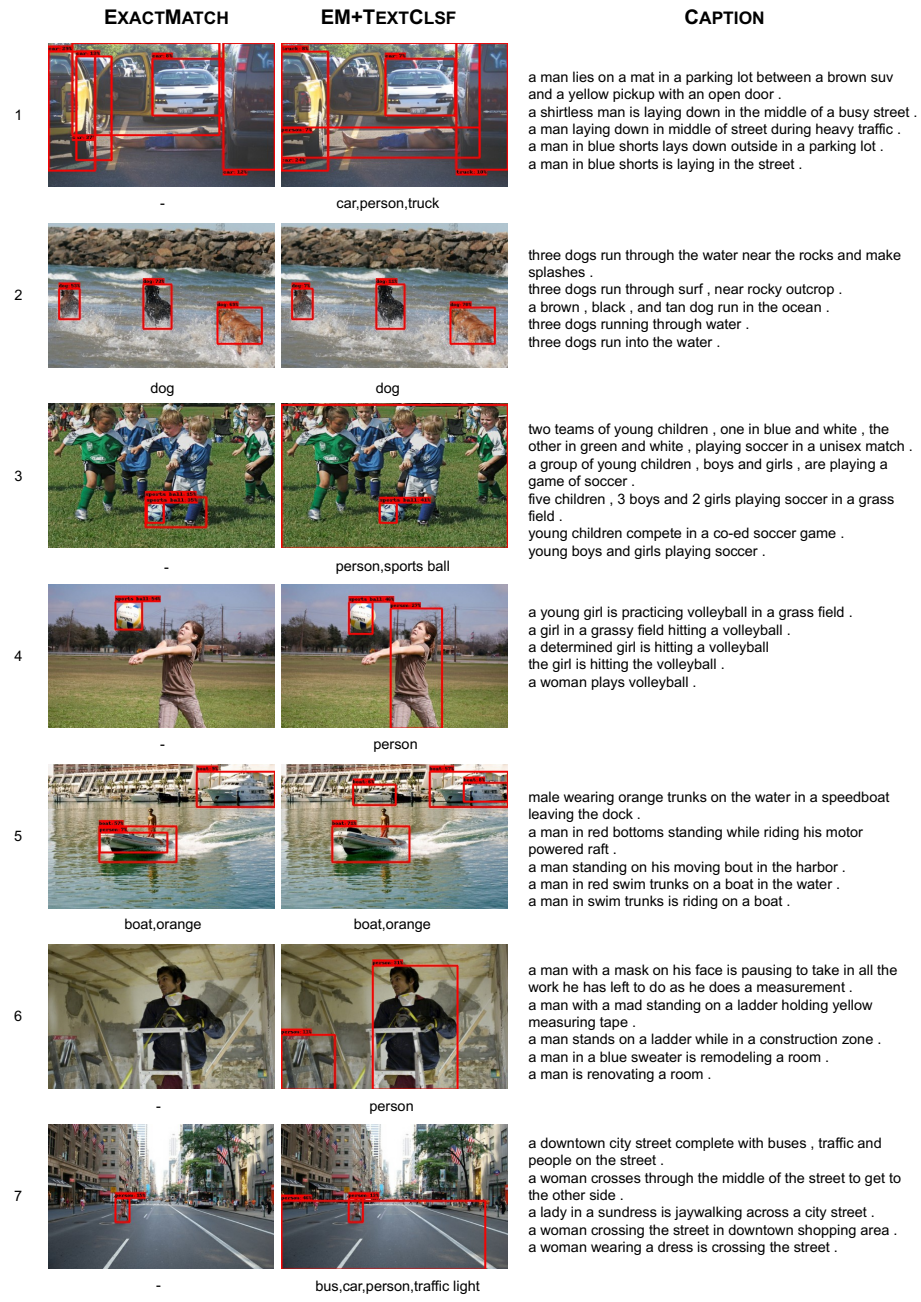


Figure 4: **Visualization of our Cap2Det model results**. We show boxes with confidence scores $> 5\%$. We also show pseudo labels extracted from textual descriptions. Please note that there is neither instance-level nor image-level object labels in Flickr30K, but our label inference module fills in this gap.