Learning to Reason from General Concepts to Fine-grained Tokens for Discriminative Phrase Detection

Maan Qraitem, Bryan A. Plummer
Department of Computer Science, Boston University
{mqraitem, bplum}@bu.edu

Abstract
Phrase detection requires methods to identify if a phrase is relevant to an image and then localize it if applicable. A key challenge in training more discriminative phrase detection models is sampling hard-negatives. This is because few phrases are annotated of the nearly infinite variations that may be applicable. To address this problem, we introduce PFP-Net, a phrase detector that differentiates between phrases through two novel methods. First, we group together phrases of related objects into coarse groups of visually coherent concepts (e.g., animals vs automobiles), and then train our PFP-Net to discriminate between them according to their concept membership. Second, for phrases containing fine grained mutually-exclusive tokens (e.g., colors), we force the model into selecting only one applicable phrase for each region. We evaluate our approach on the Flickr30K Entities and RefCOCO+ datasets, where we improve mAP over the state-of-the-art by 1-1.5 points over all phrases on this challenging task. When considering only the phrases affected by our fine-grained reasoning module, we improve by 1-4 points on both datasets.

1. Introduction
In phrase detection the goal is to identify regions from a database of images that are relevant to a phrase. This task is significantly more challenging than the localization-only task addressed in prior work, e.g., [3, 4, 8, 14, 16, 23, 26, 27, 29, 33–35, 38], that only has to localize a phrase within a ground truth image (i.e., ground truth image-phrase pairs are used at test time). Since phrase detectors can distinguish between images that are relevant to a phrase, they are more useful than localization methods in downstream tasks. For example, a phrase detector could tell you if a boat is a yacht or a canoe and indicates its location, but a phrase localization model would only point to the most likely regions the two phrases may appear in. However, distinguishing between semantically similar phrases that refer to different entities is challenging [28]. Methods for related phrase grounding tasks have proposed hard-negative mining methods to make their models more discriminative [12], but these methods suffer from a high rate of false negatives which harms detection performance [28]. This is because these methods, in part, rely on existing structured datasets like WordNet [24] to filter out false negatives, but many semantically similar phrases are not accounted for as illustrated in Figure 1(a).

To address these issues, we introduce Phrase-concepts to Fine-grained Phrase-tokens Net (PFP-Net). To avoid the false negatives from hard-negative mining methods used in prior work (e.g. [12]) we automatically construct coarse concepts: visually coherent groups of entities (e.g., animals vs automobiles). The model then is trained to discriminate between a phrase and its unrelated concepts as shown in Figure 1(b). This presents a significant challenge, since we need to create concept groups that are fine-grained enough to provide informative negatives to improve the discriminative power of our model, while being coarse enough to min-
Our model follows a Faster R-CNN framework \cite{30}. Given an image, we encode it with a ResNet-101 \cite{11} base encoder and generate a set of candidate boxes using a region proposal network. In addition to learning the similarity between regions and phrases (Section 4.1), our approach introduces a way of automatically identifying a set of coarse visually-related concepts (Section 3), and train our model to discriminate between these concepts and joint phrase-image representations as well as regularize our language embeddings using an entity-concept loss (see Section 4.2). We also introduce a fine-grained reasoning module (Section 4.3) that forces the PFP-Net to discriminate between mutually-exclusive set of fine-grained tokens (see Figure 5 for an illustration for how FGM is used at test time). Note that our paper’s contributions are labeled with dotted boxes.

Using our concepts provide two advantages over prior work. First, by creating coarse concepts we minimize false negatives. We find that simply using K-Means clustering over a language embedding like BERT \cite{7} or GLoVE \cite{25} results in noisy concept groups. This is likely because these language embeddings were trained to embed words used in similar contexts near each other, but not to distinguish between visually different entities. For example, they would embed school and teacher nearby since they are related, even though they are not visually similar which is important for phrase detection. In addition, K-means forces all entities to belong to a cluster, even the outliers, which we found often results in incoherent groups. To avoid these issues, we use visually discriminative language features to represent words. Then we use a density-based clustering to create a set of visually coherent concepts, which we assign to phrases based on their semantic similarity.

Using our concepts provide two advantages over prior work. First, by creating coarse concepts we minimize false negatives. Second, comparing a phrase to all of the concepts ensures each image region sees a balanced distribution of easy and hard negative samples, which can improve performance over focusing solely on hard-negatives \cite{36}. However, while knowing when two semantically similar words are mutually exclusive may be challenging to automate in general, we find that we can mine sets of words that typically refer to difficult fine-grained differences (e.g. a red shirt vs. a blue shirt). We take advantage of these sets of mutually exclusive words adding a fine-grained module to our model. The module differentiates between fine grained words and then augment the main model score with its prediction. An overview of PFP-Net is provided in Figure 2.

Our contributions can be summarized as:

- A novel model, PFP-Net, that achieves a 1-1.5 point gain over SOTA on phrase detection over by mining visually coherent concepts and then learning to discriminate between them.
- A fine grained reasoning module that boosts performance by 1-4 points over affected phrases by learning critical visual cues to differentiate between otherwise visually similar instances.
- A novel method for automatically mining visually coherent groups that improves the distribution of our minibatches to better represent the training data.

2. Related work

Most prior work in phrase grounding has focused on the localization-only tasks, where you are provided a ground truth image-phrase pair and have to identify the relevant image region (e.g., \cite{1, 4, 15, 16, 26, 27, 37}). However, Plummer et al. \cite{28} demonstrated that these methods tend to overfit to the localization task, i.e., they improve performance on localization, but reduce detection performance. This is partly because distinguishing between similar phrases in the localization-only task is unnecessary since most images only contain a reference to a single object of
the same type [28]. Thus, a localization model that is given the phrase a young teenager as input could look for the object category “person” to identify the right object most of the time, whereas a detector also has to determine if it exists at all. Some work falls between detection and prior work in localization, where a ground truth image-phrase pair is not provided [12,41], but they severely limit the number of negative phrases at test time. Thus, methods from these tasks also often do not generalize to phrase detection [28].

Also relevant to our work is Gupta et al. [10], who introduce a sampling method for weakly-supervised phrase grounding. They use the language model BERT [7] to sample negatives that preserve the phrase context (e.g. a walking [Blank], where Blank can not be a car). This limits its application since it requires phrase grounding datasets to be annotated with this context. However, this approach also relies on noisy external language databases like WordNet [24] to identify false negatives. In contrast, our approach creates coherent concepts that boost performance without relying on contextual annotations.

3. Visually coherent concept mining

Vision-language datasets are only annotated with a few examples of the nearly infinite valid annotations. Several methods have been proposed to increase the number of annotations (e.g. [10,12,28,32]), but prior work in identifying good negative examples leads to significant noise that hurts performance [28]. As discussed in the Introduction, this is in part because the structured language datasets used by prior work also contain incomplete annotations, resulting in many false positives. Instead, we automatically identify a set of concepts, sets of entities like clothing or buildings, and then use the concepts that a phrase is not related to as negatives during training (discussed in Section 4). We define our concepts as $C = \{c_j = \{e_k^{L_j = 0}\}_{k=0}^{M} \}$ where $e$ represents an entity (a noun referring to object) in the dataset vocabulary, $M$ is the total number of concepts, and $L_j$ is the total number of entities in concept $c_j$. Note also that $c_1 \cup c_2 \cup \ldots \cup c_M = \emptyset$. We describe how we obtain these concepts in Section 3.1. Then, we describe how we assign each phrase to its set of negative concepts in Section 3.2.

3.1. Obtaining concept groups

We start by extracting the entities (e.g., t-shirt, dog, etc) from our phrases. A straightforward approach for grouping semantically similar phrases would be to perform K-means clustering over some pretrained language model like GLoVE [25] or BERT [7]. However, text-only representations are not trained to capture visual-semantic relationships (e.g., they typically consider a teacher and school to be very similar, even though they are visually different). Thus, prior work has demonstrated that training word embeddings so that they capture visual-semantic relationships is key to good performance on many tasks [2]. While several transformer-based vision-language embeddings have been introduced (e.g., [5, 22, 31]), they are trained in a weakly-supervised setting where related, but semantically different entities which are important for our method are difficult to capture. Therefore, we selected ViCo [9] as it was trained to discriminate between visually different entities.

Simply selecting good language features is insufficient to get good concepts, however. Figure 3(a) shows that using K-means clustering over ViCo features produces noisy concepts. This is because K-means does not impose a constraint over each cluster density. I.e., outliers in the embedding space are still clustered with other concepts even though there is little evidence they belong to the same concept. Instead, we leverage DBSCAN, [6] whose clusters formed from entities that fall within a density threshold $\epsilon$. Thus, only tightly knit clusters are produced and noisy entities are discarded, creating more visually coherent concepts as shown in Figure 3(b).

Concept group quality. We evaluated different methods of creating concept groups via manual inspection. For each set of concepts, we counted the number of incoherent concepts (where 50% of the concept entities or more are not visually coherent). Figure 4 compares different methods of clustering, language embeddings, and hyperparameters affecting the number of clusters ($\epsilon$ for DBSCAN or $K$ for K-means). Our results show that DBSCAN over ViCo embeddings creates the most visually-coherent concepts.
3.2. Assigning phrases to concepts

After creating our concepts in Section 3.1, we now need to assign each phrase to its set of relevant concepts. A simple assignment process is to pair a phrase with a concept with which it shares an entity. We call this simple entity-phrase matching. This approach, however, disregards phrase-concept pairs that don’t share entities but are nevertheless viable pairs. Thus, we make these assignments using semantic similarity as detailed below.

First, given a phrase \( p \), and phrase entities \( p_e \), we consider the phrase concepts \( p_{c1} \) which are the result of a simple entity-phrase matching: \( p_{c1} = \{ e : e \cap p_e \neq \emptyset, c \in C \} \). We also consider concepts \( p_{c2} \) which are the result of a similarity based assignment process. This method is designed for when a phrase is still relevant to a concept but they do not explicitly share any entities (e.g., hoodie with concept: shirt and sweatshirt). More concretely, for each entity \( e \in p_e \), we look up the entity ViCo representation. We will denote this representation as \( \tilde{c} \). We also compute a concept \( e \) representation by averaging its entities ViCo feature vectors. We will denote this as \( \tilde{c} \). Now assume \( \text{sim}(\tilde{e}, \tilde{c}) = \frac{\tilde{e} \cdot \tilde{c}}{|\tilde{e}| |\tilde{c}|} \), we compute an “association” score between the entity \( e \) and a concept \( c \) as follows:

\[
\text{Assoc}(e, c) = \frac{\exp(\text{sim}(\tilde{e}, \tilde{c})/\tau)}{\sum_{j=0}^{M} \exp(\text{sim}(\tilde{e}, \tilde{c})/\tau)} \tag{1}
\]

From here, \( p_{c2} = \{ c : \text{Assoc}(e, c) > \gamma, e \in p_e, c \in C \} \). This way, the phrase is assigned to concepts with which it shares the most statistical ViCo based similarity. The final set of phrase concepts is therefore, \( p_c = p_{c1} \cup p_{c2} \). Thus, a phrase that is near multiple concepts will be assigned to all of them, and so we will only contrast phrases against groups that we are confident are negatives.

4. Learning From General Phrase-concepts to Fine-grained Phrase-tokens

In phrase detection the task is to determine what images regions \( r \) can be described by phrase \( p \). Prior work addressing phrase localization only selects image regions from the ground truth image, but in phrase detection regions from all images in the dataset are considered. To address this problem, we introduce a neural network that learns from Phrase-concepts to Fine-grained Phrase-tokens (PFP-net). For a fair comparison to prior work [12, 28], PFP-net adopts a Faster R-CNN architecture [30] to obtain a set of image regions, which we describe briefly in Section 4.1. Then, we use the concept mining procedure described in Section 3 in two ways. First, in Section 4.2 we use the mined entity based concepts to identify the general object type of a phrase which provides a broad coverage informative negatives. Second, we obtain the initial set of concepts described in Section 3.1 over adjectives, where the concepts typically denote mutually-exclusive fine-grained differences in phrases (e.g., a red truck vs. a green truck). Thus, in Section 4.3 we use these adjective-concepts in our model to differentiate between mutually exclusive phrases.

4.1. Encoding phrases and image regions

Following [12, 28], we adopt a Faster R-CNN architecture [30] using a ResNet-101 [11] encoder as a backbone in order to encode regions and mean-pooled HGLMM Fisher vectors [19] to represent our phrases. However, we note that the contributions of our approach, namely the concept-conditioned training procedure and fine-grained module , described in the subsequent sections, are adaptable to any detection architecture. We obtain a final region and phrase representation using a pair of 2-layer multi-layer perceptrons (MLP), one that takes the HGLMM features as input and the other that takes the ROI-pooled features from the Faster R-CNN encoder. After obtaining a good feature representation for the image regions \( r \) and phrase (or entity/concept) features \( p \) from their respective MLPs, we compute the similarity between them via:

\[
P_{RSim}(r, p) = \text{MLP}_{3\text{Layer}}(r \odot p), \tag{2}
\]

where \( \odot \) denotes an elementwise product. Note that this function is represented as the “Joint MLP” in Figure 2. Following Plummer et al. [28], partway through training we re-initialize the 2-layer region and text MLPs via canonical correlation analysis (CCA) [13], which Plummer et al. found greatly improved phrase detection performance. We also apply an L2 regularization (\( L_{reg} \)) with the initial CCA weights when fine-tuning these MLPs to avoid catastrophic forgetting. Our model is trained using a phrase-region similarity loss (PR). Formally, let \( K \) be the number of image region-query pairs in a batch, \( s^p \) be a region-phrase pair.
score, $l^p$ be its $-1/1$ label indicating whether it is a negative/positive pair, $a$ be the conditional embedding weights before the softmax, then the base loss is:

$$L_{\text{base}} = \sum_{i=1}^{K} \log(1 + \exp(-l^p_i s^p_i)) + \lambda_{\text{reg}} L_{\text{reg}}$$

(3)

where $\lambda_{\text{reg}}$ is a scalar parameter.

### 4.2. Expanding batch coverage using concepts

One of the primary contributions is using the generated concepts described in Section 3 to provide a set of informative negatives that represent our entire training dataset. We use these concepts in two ways: (1) as additional samples for the output of the Eq 2 computed between the regions and concepts (referred to as “Concept-region Loss in Figure 2), and (2) to regularize the MLP textual representation (referred to as “Entity-Concept Loss in Figure 2). Note that our concepts are encoded using text features. See our detailed explanations below.

**Concept-region Loss (CR)** Given a phrase $p$, and its set of related concepts $p_c$, we use the unrelated concepts $p_c = \{ c : c \not\in p_c, c \in C \}$ as negative samples. We then pair those concepts with the region $r$ associated with phrase $p$ to obtain negative concept-region pairs $\tau_r$. We then simply concatenate the concept-region pairs to the phrase-region pairs and then feed their scores through the logistic loss described in Eq 3. Let $s^c$ be region-negative concept score, $L$ be the total number of negative concept-region pair, then the loss:

$$L_{CR} = \sum_{i=1}^{L} \log(1 + \exp(s^c_i))$$

(4)

**Entity-concept Loss (EC).** In addition to ensuring that each image region is embedded near its related concepts, we also encourage phrases to embed near its associated concepts and far from others. For each phrase in our batch, we extract their entities (nouns). We randomly sample one entity for each phrase. We then compute an entity-concept score $s^e$ by taking the dot product between the $l^c_2$ normalized entity features and the concept representations followed by a Sigmoid activation function. Now assuming $l^c$ is the entity-concept $-1/1$ label indicating whether it is a negative/positive pair, and $S$ is the total number of entity-concept samples, then the entity regularization loss can be computed as:

$$L_{EC} = \sum_{i=1}^{S} \log(1 + \exp(-s^e_i l^c_i))$$

(5)

### 4.3. Fine-Grained reasoning Module (FGM)

After obtaining our initial phrase-region similarity scores using Eq 2, we modify it by taking into account fine-grained differences between phrases. In particular, we have the model predict the scores of sets of mutually exclusive fine-grained tokens, and then use them to augment the main model region-phrase scores. More formally, given a set of tokens sets $F$ where each $f \in F$ is itself a set of fine grained tokens, our novel fine grain module learns to discriminate between each element of $f$. We obtain $f$ using the same bottom up approach used to mine for concepts over adjectives. However, in this case, we train the model to discriminate between each concept tokens rather than the concepts themselves, since adjectives typically group together words that denote fine-grained differences between phrases, as mentioned earlier. While other parts-of-speech could also be used (e.g., verbs also denote a mutually-exclusive state of an object), we found that our datasets did not contain enough samples for training a FGM module over other concept types.

The FGM module encodes image regions using a set of convolutional layers and then performs multi-label classification on each set of fine grained tokens $f$. Now, assume $t$ is a fine grained token such that $t \in f$, $R$ be the number of region-fine grained token pairs, $s^t$ be the region-token score, and $l^t$ be its 0/1 label indicating whether it is a positive/negative region-token pair, then:

$$L_{FGM} = \sum_{i} l^t_i \log s^t_i + (1 - l^t_i) \log(1 - s^t_i)).$$

(6)

With this, the final loss for PFP-net is:

$$L_{\text{final}} = L_{\text{base}} + \lambda_{CR} L_{CR} + \lambda_{EC} L_{EC} + L_{FGM}$$

(7)

### 4.4. Model inference

At test time the FGM module’s scores are augmented with the base model’s phrase-region scores. Given a phrase $p$, the phrase tokens $p_t$ and $t \in p_t$, a phrase-region pair score $s^p$, and token-region score $s^t$, then the final score $s^f$:

$$s^f = (1 - \lambda_f)s^p + \lambda_f s^t,$$

(8)

where $\lambda_f$ is a scalar parameter that applies for every $t \in f$. See Figure 5 for an illustration.

## 5. Experiments

**Datasets:** We evaluate our PFP-net on the two phrase grounding datasets which are the best suited for benchmarking phrase detection. First, we use Flickr30K Entities [29] that consists of 276K bounding boxes in 32K images for the noun phrases associated with each image’s descriptive captions (5 per image) from the Flickr30K dataset [39]. We use the official splits [29] that consist of 30K/1K/1K train/test/validation images. Second, we evaluate on RefCOCO+ [40], which consists of 19,992 images from the
Table 1. mAp Split by frequency of training instances. (a) contains results reported in prior work or produced using their code. (b) contains ablations of our model that compares the performance of our three novel components (EC, ER, and FGM). See Section 5.1 for discussion.

| #Train Samples | Flickr30K Entities | RefCOCO+ |
|----------------|---------------------|----------|
|                | zero shot | few shot | common | mean | zero shot | few shot | common | mean |
| (a) QA R-CNN [12] QA R-CNN + NPA [12] SimNet w/CCA [28] | 3.9 4.3 8.9 5.7 | 3.8 4.1 9.7 5.9 | 9.7 11.2 17.3 12.7 | 6.0 10.2 20.1 12.1 |
| (b) PFP-Net (CR) PFP-Net (CR+EC) PFP-Net (CR+EC+FGM) | 10.2 11.9 18.1 13.5 | 10.4 12.0 18.6 13.7 | 10.6 12.4 19.0 14.0 | 6.2 10.5 21.1 12.6 |

Table 2. Performance of the phrases impacted by our FGM module compared between previous SOTA (SimNet w/CCA) and our model. See Section 5.1 for discussion.

| #Train Samples | Flickr30K Entities | RefCOCO+ |
|----------------|---------------------|----------|
|                | zero shot | few shot | common | mean | zero shot | few shot | common | mean |
| SimNet w/CCA | 13.0 14.7 12.0 13.2 | 7.0 10.3 11.0 9.4 |
| PFP-Net (CR+ER) | 14.3 15.0 15.8 15.0 | 7.0 10.3 11.4 9.6 |
| PFP-Net (CR+ER+FGM) | 15.0 17.2 18.8 17.0 | 7.7 10.8 14.3 10.9 |

Figure 5. PFP-Net inference overview During inference, the model takes in three inputs: the image, the phrase, and any applicable fine grained token within the phrase. The model then produces two scores: a phrase-region score and a fine grained token-region score. Finally, the weighted sum of the scores is then computed following Eq 8 to produce the final score $s^f$.

COCO dataset [21] that have been labeled with 141,564 region descriptions. We use the official split [40], which splits the train/val and testing sets 16K/1.5K/1.5K. Both datasets are licensed under creative commons.

Metrics: We follow the evaluation protocols of Plummer et al. [28]. For every image we obtain the most likely region and confidence score for every phrase in our test split. For any ground truth phrases in an image, we consider a phrase successfully localized if the predicted bounding box has at least 0.5 intersection-over-union with its associated ground truth bounding box. Then, we compute average precision (AP) for each phrase and then split them into zero-shot, few-shot, and common sets, based on if they didn’t occur in our training split, if they had between 1-100 occurrences, or if they occurred more than 100 times, respectively. We then report an overall mAP for each set of phrases, as well as the average of them for an overall performance score. This procedure ensures that the zero-shot and few-shot phrases are not over represented compared to the common phrases, since the zero- and few-shot sets have more unique phrases, but represent a smaller portion of overall instances.

Table 3. mAp split by frequency of training instances where augmented positive phrases (PPA) from [28] is used for evaluation. The table compares our model three novel components (EC, ER, and FGM) to state of the art (SimNet w/CCA). See Section 5.1 for discussion.

| #Train Samples | Flickr30K Entities | RefCOCO+ |
|----------------|---------------------|----------|
|                | zero shot | few shot | common | mean | zero shot | few shot | common | mean |
| SimNet w/CCA [28] | 9.4 11.2 19.8 13.5 | 6.2 10.3 20.5 12.3 |
| PFP-Net (CR) | 10.6 12.1 20.4 14.3 | 6.8 10.4 21.9 12.7 |
| PFP-Net (CR+EC) | 10.6 12.2 20.9 14.6 | 6.5 10.7 21.3 12.8 |
| PFP-Net (CR+EC+FGM) | 11.3 12.6 21.2 15.0 | 6.8 10.9 22.5 13.4 |
Figure 6. Comparison between our model vs prior work model on positive (ground truth) and randomly sampled negative phrases scores. Our model is significantly better able to separate positive vs negative phrases scores than prior work. This result is key in enhancing model discriminative ability which in turn translates to improvement on the mAP scores as documented in Table 1. See Section 5.2 for additional discussion.

Implementation details. We train PFP-net with an ADAM [18] optimizer using the hyperparameter settings of Plummer et al. [28] for a fair comparison except where stated. We set all hyperparameters introduced by our work via grid search. Specifically, when creating our concept groups using DBSCAN we set $\epsilon$ (the threshold to determine clustering tolerance), and the $\gamma, \tau$ in Eq 1. The used values were $\gamma = 0.2, \tau = 0.01$ for both datasets. $\epsilon = 0.43$ for Flickr30K and $\epsilon = 0.53$ for RefCOCO+. For entity regularization, we set $\lambda_{ER} = 0.1$ and $\lambda_{CR} = 1$ for both Flickr30K Entities and RefCOCO+. We trained our model using a single NVIDIA RTX 8000 GPU using an internal cluster.

Figure 7. Effect of the concept set size $|C|$ has on detection performance. See Section 5.2 for discussion.

Figure 8. Effect of changing $\gamma$ in Eq 1 has on detection performance. See Section 5.2 for discussion.

Figure 9. Effect of changing $\lambda_{EC}$ in Eq 7 has on detection performance. See Section 5.2 for discussion.

5.1. Results

Table 1 compares ablations of PFP-Net with the current state-of-the-art on phrase detection. Comparing the last line of Table 1(b) to the results from prior work in Table 1(a) we get 1-1.5 point gain in mAP over the state of the art. Much of this gain came from common phrases, where we achieved a 2 point gain over prior work. In Table 1(b) we also note the contribution of each component of our model, with the biggest gain coming from using our concept-region loss (CR). Note that in Table 1(a) the hard-negative min-
ing approach of Hinami and Satoh [12] achieved a much more modest 0.2 point gain. Note that the results in our tables are the results of 5 different runs to ensure our reported gains are not due to model variance. We note that CR produces consistent improvements on both datasets across all phrase types (zero shot, few shot, common). EC results in small, but largely consistent improvements on both datasets. These results indicate that our negative samples are effective in improving our model discriminative ability. In addition to the gains from using our concepts, we note that Table 1(b) reports that the FGM module from Section 4.3 further boosts performance by a small but consistent gain for both datasets. Moreover, in Table 2, we report performance on only the phrases affected by our FGM module, which reports a significant 3.8 point improvement on Flickr30K Entities, while also obtaining a 1.5 point improvement on RefCOCO+. Finally, Table 3 reports the performance of PFP-Net using positive phrase augmentation (PPA) [28], which reduces annotation sparsity by pairing ground truth phrases with plausible positive phrases using WordNet [24]. We note that PPA does not change the relative gains of the detection methods, but obtains higher absolute performance.

5.2. PFP-Net Analysis

To further understand the behaviour of our model, we examine several qualitative images in Figure 6. Compared to prior work, our model maintains its confidence in positive ground truth phrases and correctly localize them while also having significantly lower scores for negative phrases. For example, in Figure 6(b) the negative phrase "sofa" in the left image has a significantly lower score than prior work in Figure 6(a). This is evidence that simply optimizing for phrase localization only does not translate to discriminative detection power. As shown in the center image of Figure 6 prior work fails to differentiate between white and black, resulting in an incorrect localization of the "white dog". Our PFP-Net, on the other hand, with the help of our FGM module, correctly localizes the phrase.

We further investigate our model behaviour via sensitivity analysis of its hyperparameters. First, we examine the effect of changing the number of used concepts in Figure 7. For each dataset, we progressively increase DBSCAN’s density threshold until we can not generate more concepts. We observe that overall more concepts improve performance on both datasets. This is probably because more concepts split the language distribution into finer portions, thus improving the accuracy of our phrase-concept assignment in Section 3.2 which translate to better performance overall. Second, we examine the effect of changing Eq 1 $\gamma$ parameter in Figure 8 that controls whether a given concept is assigned to a phrase (i.e., they are related). We vary the parameter between 0 (a phrase is assigned to every given concept) and 1 (a phrase is only assigned to concepts that share tokens with). We note that performance is at best for values of $\gamma$ between (0.2, 0.6). Performance drop for $\gamma < 0.2$ as we assign wrong concepts to phrases. Performance also drops when $\gamma > 0.8$ which indicates the importance of our similarity based matching component. In other words, simply assigning phrases to concepts that only share tokens with is not sufficient. This is because, as noted in Section 3.2, there are many concepts that are still relevant to certain phrases but also do not share tokens with.

Third, and finally, we study the effect of changing $\lambda_{EC}$ used in Eq 7 in Figure 9 that controls the contribution of our Concept-Entity loss. Overall, we note that performance for both datasets is best at 0.1. We note that both the $\gamma$ from Eq 1 and $\lambda_{EC}$ from Eq 7 achieved the highest performance at the same parameter value, demonstrating that these hyperparameter val generalize across datasets, whereas PFP-Net tends to favor larger number of concepts.

6. Limitations and societal impact

Phrase detection methods can help us understand the content of images. This is useful for downstream tasks like image captioning and visual question answering. This is useful for some tasks like answering questions about images posed by people who have visual impairments. However, these benefits also come with some risks, such as enabling some applications in tasks like surveillance by allowing a user to quickly locate specific entities in a database of images. Although our approach provides significant performance improvements, absolute performance on this task is still poor. While this suggests there are ample opportunity for researchers to improve performance on this task, it also indicates that the results of these systems should not be trusted blindly. We also note that Flickr30K Entities and COCO may contain personally identifiable information, but they remain standard benchmarks making them an important comparison.

7. Conclusion

In this work, we introduced a new phrase detection model (PFP-NET) that significantly improves performance by 1-1.5 points on two phrase detection data-sets. The model does so by incorporating visually coherent clusters (concepts) to sample negative concept-region as well concept-entity pairs that effectively improve the model discriminative abilities when compared to prior work. Our model further improves performance by incorporating a novel fine grained module that learns to discriminate between adjective fine grained tokens. Note that although our experiments used the Faster R-CNN framework to fairly compare to prior work, the contributions made by our paper (concept-based sampling and fine grained module) are modular and so can be adapted to any underlying detection framework.
References

[1] M. Bajaj, L. Wang, and L. Sigal. G3raphground: Graph-based language grounding. In ICCV, 2019.
[2] A. Burns, R. Tan, K. Saenko, S. Sclaroff, and B. A. Plummer. Language Features Matter: Effective language representations for vision-language tasks. In The IEEE International Conference on Computer Vision (ICCV), 2019.
[3] K. Chen, R. Kovvuri, J. Gao, and R. Nevatia. Msfc: Multimodal spatial regression with semantic context for phrase grounding. In Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, ICMR ’17, page 23–31, New York, NY, USA, 2017. Association for Computing Machinery.
[4] K. Chen, R. Kovvuri, and R. Nevatia. Query-guided regression network with context policy for phrase grounding. In 2017 IEEE International Conference on Computer Vision (ICCV), pages 824–832, 2017.
[5] Y.-C. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu. Uniter: Universal image-text representation learning. In The European Conference on Computer Vision (ECCV), 2020.
[6] D. Deng. Dbscan clustering algorithm based on density. In 2020 7th International Forum on Electrical Engineering and Automation (IFEEA), pages 949–953, 2020.
[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, June 2019.
[8] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 457–468, Austin, Texas, Nov. 2016. Association for Computational Linguistics.
[9] T. Gupta, A. Schwing, and D. Hoiem. Vico: Word embeddings from visual co-occurrences. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2019.
[10] T. Gupta, A. Vahdat, G. Chechik, X. Yang, J. Kautz, and D. Hoiem. Contrastive learning for weakly supervised phrase grounding. In European Conference on Computer Vision (ECCV), 2020.
[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.
[12] R. Hinami and S. Satoh. Discriminative learning of open-vocabulary object retrieval and localization by negative phrase augmentation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2605–2615, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics.
[13] H. Hotelling. Relations Between Two Sets of Variates, pages 162–190. Springer New York, New York, NY, 1992.
[14] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4555–4564, 2016.
[15] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. In CVPR, 2016.
[16] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg. ReferItGame: Referring to objects in photographs of natural scenes. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 787–798, Doha, Qatar, Oct. 2014. Association for Computational Linguistics.
[17] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg. Referit game: Referring to objects in photographs of natural scenes. In EMNLP, 2014.
[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017.
[19] B. Klein, G. Lev, G. Sadeh, and L. Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4437–4446, 06 2015.
[20] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. In The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[21] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, Computer Vision – ECCV 2014, pages 740–755, Cham, 2014. Springer International Publishing.
[22] J. Lu, D. Batra, D. Parikh, and S. Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In Advances in Neural Information Processing Systems, 2019.
[23] J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 11–20, 2016.
[24] G. A. Miller. Wordnet: A lexical database for english. Comm. ACM, 38(11):39–41, Nov. 1995.
[25] J. Pennington, R. Socher, and C. Manning. GloVe: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, Doha, Qatar, Oct. 2014. Association for Computational Linguistics.
[26] B. A. Plummer, P. Kordas, M. H. Kiapour, S. Zheng, R. Piremuthu, and S. Lazebnik. Conditional image-text embedding networks. In Computer Vision – ECCV 2018, pages 258–274, Cham, 2018. Springer International Publishing.
[27] B. A. Plummer, A. Malliya, C. M. Cervantes, J. Hockenmaier, and S. Lazebnik. Phrase localization and visual relationship detection with comprehensive image-language cues. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017.
Table 4. Phrase Localization accuracy comparison using Positive Phrase Augmentation (PPA) [28] between prior work (SimNet w/CCA) and our model three novel components (EC, ER, and FGM). See Section A for discussion.

| Model                      | Flickr30K Entities | RefCOCO+ |
|----------------------------|--------------------|----------|
| SimNet w/CCA               | 71.7               | 57.5     |
| PFP-Net (CR)               | 70.6               | 56.2     |
| PFP-Net (CR+ER)            | 70.7               | 56.3     |
| PFP-Net (CR+ER+FGM)        | 70.8               | 56.6     |

Table 5. Phrase Localization accuracy comparison using Positive Phrase Augmentation (PPA) [28] between prior work (SimNet w/CCA) and our model three novel components (EC, ER, and FGM). See Section A for discussion.

| Model                      | Flickr30K Entities | RefCOCO+ |
|----------------------------|--------------------|----------|
| SimNet w/CCA               | 66.8               | 59.5     |
| PFP-Net (CR)               | 58.8               | 57.6     |
| PFP-Net (CR+ER)            | 58.9               | 57.7     |
| PFP-Net (CR+ER+FGM)        | 60.3               | 58.3     |

A. Phrase localization

Our work makes advances on phrase detection. The problem stands in contrast to localization-only tasks where detection is not required (i.e. ground truth phrases are given at test time). As prior work [28] demonstrated, it is too easy to overfit to the localization-only task. In other words, localization performance and detection performance are not causally related, so localization performance is of very limited use for evaluating our detection approach. However, to be complete, we report localization performance of our model without PPA in Table 4 and with PPA in Table 5. We note that our methods’ localization numbers are on par with previous work, with the only exception of when PPA is applied on Flickr30K. We would argue that this is likely due, in part, to the fact that the PPA augmentations are noisy, as [28] themselves acknowledged. For example, using PPA augmentation on Flickr30K, any region labeled as “swimmers head cap” would also be labeled as a positive for “roof” or “jacket,” although these are clearly incorrect associations. Since our work more accurately distinguishes between phrases, we would more likely label the correct image region for such spurious associations.

B. Concept generation ablations

In our work, we make use of visually coherent groups (concepts) to provide negative samples that cover a broad spectrum of the training distribution and minimize false negatives. We argue that minimizing visual coherence noise in our concepts is key to minimizing false negatives. As noted in Section 3.1 in our paper, DBSCAN + ViCo re-
result in concepts with the least amount of visual coherence noise. Thus, we expect this combination to have the best performance. To verify this claim, we provide an ablation study that document the impact of varying the embedding/clustering algorithm combination on the CR component of our PFP model. The component is responsible for using the concepts as negative samples. For each combination, we select the clustering algorithm hyper parameters that produce the least visual coherence noise. As we can see in Table 6, DBSCAN+ViCo perform the best on both datasets. While other combinations have equal performance on Flickr30k entities, their performance align in RefCOCO+ dataset with our visual coherence noise metric. More concretely, K-means overall perform worse than DBSCAN and ViCO improves performance on either of the clustering algorithms when compared to GLoVE.

| #Train Samples   | zero shot | few shot | common | mean   | zero shot | few shot | common | mean   |
|------------------|-----------|----------|--------|--------|-----------|----------|--------|--------|
| Baseline [SimNet w/CCA [28]] | 9.7       | 11.2     | 17.3   | 12.7   | 6.0       | 10.2     | 20.1   | 12.1   |
| K-means + GLoVE  | 10.0      | 11.8     | 18.5   | 13.4   | 6.0       | 9.9      | 20.4   | 12.1   |
| K-means + ViCo   | 10.1      | 11.9     | 18.3   | 13.4   | 6.0       | 9.8      | 20.7   | 12.2   |
| DBSCAN + GLoVE   | 9.9       | 11.9     | 18.4   | 13.4   | 6.2       | 10.4     | 20.7   | 12.4   |
| DBSCAN + ViCo    | 10.4      | 11.8     | 18.3   | 13.5   | 6.1       | 10.4     | 21.0   | 12.6   |

Table 6. mAP Split by frequency of training instances. The table contains ablations of our model that compares the performance of its CR component using different concept generation methods (i.e. clustering algorithm/embedding combinations) to state of the art (SimNet w/CCA). See Section B for discussion.

C. Evaluation dataset selection

The sparsity of phrase detection datasets annotations poses significant challenges on evaluation. Real world datasets annotations simply can not cover all the possible positive cases. For example, while a region might be annotated with only the phrase blue shirt, it can also be correctly labeled with clothing. Thus, if a model assigns a region the phrase clothing, the evaluation process will incorrectly classify the case as negative (i.e. false negative). Prior work [28] attempted to mitigate this problem by introducing Positive Phrase Augmentations (PPA) where structures like WordNet [24] are used to derive additional positive samples for a given annotation. However, this problem is not limited to issues with synonyms. Phrases might have different structures but can convey the same meaning (e.g. frisbee that is round vs a round frisbee), thus resulting in the same false negatives problem in evaluation. While the authors of datasets like Flickr30K [29] limited the structure of their annotations such that this problem would not arise, this is not the case for datasets like Visual Genome [20] or Referit [17]. Moreover, both datasets were used in prior work to evaluate phrase detection algorithms [28], but given the aforementioned problem with false negatives, their validity for phrase detection evaluation is not clear. To quantitatively document this issue, we sampled 30 random phrases from each dataset and considered the top 5 most similar phrases using the visual based language representation ViCo [9]. For each of these top 5 phrases, we manually counted the number of false negatives. We report the average results in Table 7. As we can observe, both Referit and Visual Genome suffer from significantly higher false negative rates when compared to Flickr30k Entities. Thus, they are not viable evaluation datasets for phrase detection and so we choose not to use them in our paper. We instead use Flickr30k Entities as well as RefCOCO+ [40]. RefCOCO+ was collected using the same underlying game as Referit but the authors improved the data collection standards. For example, they disallowed players from using location words in their referring expressions by adding “taboo” words to the ReferIt Game which earlier would result in many false negatives. Thus, as the authors note, the resulting expressions were more appearance focused and more concise. This is further evident in the lower false negative rate in Table 7 when compared to that of Referit.

Even though Flickr30k and RefCOCO+ are significantly less noisy in terms of false negatives than Referit and Visual Genome, they still exhibit a significant amount of noise. Thus, future work can benefit from developing better evaluation procedures and datasets to improve the the evaluation accuracy of phrase detection models.

| Dataset            | False Negative Rate |
|--------------------|---------------------|
| Referit [17]       | 74%                 |
| Visual Genome [20] | 72%                 |
| Flickr30k Entities [29] | 31%             |
| RefCOCO+ [40]      | 40%                 |

Table 7. Dataset annotations’ false negative rate. From each of the following datasets: Flickr30k, Visual Genome, RefCOCO+ and Referit, we sample 30 random phrases and manually calculate the average percentage of false negatives in each phrase top 5 most similar phrases. See Section C for more details and discussion.
In Figures 10 and 11, we provide additional qualitative results. As we discuss in the paper, the figures demonstrate how our model is better able at expanding the gap between the positive phrases (i.e. groundtruth) and the negative phrases scores. For example, consider Figure 11 (a) middle image, prior work model assigns 0.995 score for the phrase edge mountain while our model in Figure 11 (b) middle image assigns 0.39 score while maintaining high confidence in the ground truth phrases. This improvement is a direct result of using our concepts in both the CR and EC modules. The concepts expose the model to a wider scope of the language distribution while minimizing false negatives, and hence increase the model confidence against negative phrases. We also note that this increased discriminative power translates to tighter and better fit boxes. For example, consider the box around the waterfall in Figure 11 right image (a) [prior work] compared to (b) [our work]. Our work model boxes fit the target phrase beautiful waterfall more tightly. This improvement can be due to the model’s increase confidence in the phrase with respect to other negative phrases.