Abstract

Localizing natural language phrases in images is a challenging problem that requires joint understanding of both the textual and visual modalities. In the unsupervised setting, lack of supervisory signals exacerbate this difficulty. In this paper, we propose a novel framework for unsupervised visual grounding which uses concept learning as a proxy task to obtain self-supervision. The simple intuition behind our idea is to encourage the model to localize to regions which can explain some semantic property in the data, in our case, the property being the presence of a concept in a set of images. We present quantitative and qualitative experiments to demonstrate the efficacy of our approach and show a 5.6% improvement over the state of the art on Visual Genome dataset, a 5.8% improvement on the ReferItGame dataset and comparable to state-of-art performance on the Flickr30k dataset.

1 Introduction

Utilizing arbitrary length phrases for visual grounding overcomes the limitation of using a restricted set of categories for localization and provides a more detailed description of the region of interest as compared to single-word nouns or attributes. The problem of supervised visual grounding (i.e localizing) of phrases has hence gathered a lot of interest in recent vision literature [4, 7, 2]. However, supervised approaches require expensive bounding box or pixel level annotations which are difficult to scale. In this paper we address the problem of visual grounding of textual phrases with an unsupervised approach where no bounding box annotation exists during training. Given the lack of supervision, we develop a self-supervised proxy task which can be used to guide the learning. The general idea behind self-supervision is to design a pretext task which involves explaining some regularity about the input data. Instead of directly optimizing the localization objective, the model is trained with a surrogate loss which tries to optimize for the proxy task. A good proxy improves performance on the final task when the surrogate loss is minimized. We propose concept-learning as a substitute task for visual grounding. During training, we create concept batches of size \( k \), consisting of \( k \) different phrase-image pairs, all containing a common concept. We exploit this presence of semantic commonality within the concept batch to generate supervisory signals. We hypothesize that to predict these commonalities, the model must localize them correctly within each image of the set. We induce a parametrization in the form of attention which, given the input text and image, can localize the phrase. These localized regions are then used to predict the common concept. The entity to be grounded in most phrases is a single-word occurring in varying contexts and learning high-level semantic representations for these concept can improve visual grounding. In summary, the main contributions of our work are as follows:

- We propose a novel framework for visual grounding of phrases through semantic self-supervision where the proxy task is formulated as concept learning. We introduce the idea of a concept batch to aid learning.
Figure 1: An overview of our model. The encoder takes in a set of image-phrase pairs, indexed by $i$, all sharing a common concept and embeds them to $V_i$ and $t_i$ respectively. These features are used to induce a parametrization for spatial attention. Next, the decoder uses the visual attention map to predict the common concept. In addition, the decoder also predicts the common concept independently for each pair ($i$). For details, see Section 2.

- We evaluate our approach on the Visual Genome and ReferIt dataset and achieve state-of-art performance with a gain of 5.6% and 5.8% respectively. We also get performance comparable to the state-of-art on Flickr30k dataset.

- We analyze the behavior of our surrogate loss and the concept batch which gives an insight into the functioning of our approach. We also analyze the correlation of our model’s performance with factors like bounding box size and concept similarity.

2 Grounding through semantic self-supervision

Proxy task formulation. Our model is trained for the proxy task of concept-learning. A concept is defined as the entity which is to be grounded in the image. For example, in the phrase ‘white towel on the counter’, the highlighted word ‘towel’ is the concept. We hypothesize that if we induce a parametrization for localization of the phrase and use the localized regions to predict the concept present in an image, the parametrization will converge to the ground truth localization of the phrase. Given this proxy task, we’re faced with two main challenges: 1) How do we identify the concept in a phrase? and 2) How do we learn concept representations in an unsupervised setting? For the first part, we assume the concept is a single-word entity and exists within the phrase, and naively pick a random noun from the phrase. For the second problem, we introduce the notion of a concept batch. A concept batch is one training instance for our model, which itself consists of $k$ phrase-image pairs, all containing a common concept. The proxy task is now formulated as jointly decoding the common concept using all $k$ localized feature representations in addition to independently decoding the same concept. The intuition behind training with a concept batch is that for decoding the common concept, $k$ phrase-image pairs should encode a localized representation which is invariant to the difference in context across the $k$ pairs. On the other hand, the proxy task of decoding independent concept (for each image in the batch) ensures two things: a) Individual and common representations are consistent b) Model cannot find a shortcut by using only few inputs from the concept batch to decode the common concept (see section 4). Note that using a concept batch with random sampling of nouns can be interpreted as generating weak supervision, albeit noisy in nature. Since the same image-phrase pair can be chosen with different sampled concepts during training, it is this random sampling of nouns which ensures that the model doesn’t only learn a simple concept-identifier, but also generates discriminative information about a concept in different contexts.

Encoder-Decoder model. We adopt an encoder-decoder architecture for learning to ground as illustrated in Figure 1. The encoder uses an attention mechanism similar to [10] using the joint features from visual and textual modalities. Similar to previous work, we use a pre-trained VGG16 and a language model [1] and freeze their weights during training. For the $i^{th}$ index in the concept batch, given visual features $V^i = f_{VGG}(I^i)$ and textual features $t^i = f_{LM}(P^i)$, the attention over visual regions is given by:

$$f_{attn}^i = \text{softmax}(f_{\text{joint}}(V^i, t^i)).$$

$$f_{\text{joint}}(V^i, t^i) = \Phi_s(\Phi_r(\Phi_q([V^i, t^i]))).$$
We find that aggregating the visual attention across regions, which is commonly done in the past with the pointing game metric [11]. Out of the 49 regions in the attention map, we choose the center where $V_n$ (Since our attention is over VGG16 features, visual-only baseline.)

The second is the center baseline which always picks the center-most point of the grid which is maximally activated, as the maximum pixel for the pointing game. We also test diverse datasets in terms of their phrase and image statistics, namely Visual Genome, Flickr30k of the 49 grids. The second is the center baseline which always picks the center-most point of the concept vocabulary instead of image-phrase samples during training. In the encoder, the values from Equation 2 are taken as $\Phi_{\text{vac}}(X) = ReLU(W_p X + b_p)$, (3) where $ReLU(x) = \max(x, 0)$, $W_p \in \mathbb{R}^{p \times d}$, $b_p \in \mathbb{R}^{p \times 1}$ and $X \in \mathbb{R}^{d \times n}$. Here $n$ is the number of regions over which attention is defined and $d$ is the dimensionality of each region with respective to $X$. Thus, we use a 4 layered non linear perceptron to calculate attention for each of the $n$ regions (Since our attention is over VGG16 features, $n = 7 \times 7$). The four $\Phi(\cdot)$ layers gradually decrease the dimensionality of the concatenated joint features from $(m + l) \rightarrow q \rightarrow r \rightarrow s$ where $s = 1$.

Note that the attention module is shared across all $V^i$ and $t^i$ and the encoder is common for all pairs in the concept batch. Given the attention weights $f_{\text{attn}}^i \in \mathbb{R}^{1 \times n}$, the visual attention for common concept prediction ($f_{\text{vac}}^i$) is computed by taking the weighted sum with the original visual features.

$$f_{\text{vac}}^i = \sum_{i=1}^{k} f_{\text{attn}}^i V^i$$

(4)

We find that aggregating the visual attention across regions, which is commonly done in the past attention literature degrades performance for our task. Therefore we retain the spatial information and only aggregate the features across the concept batch. Similarly, the visual attention for independent concept prediction, $f_{\text{vai}}^i$ is given by the element-wise product of the attention weights and visual features.

$$f_{\text{vai}}^i = f_{\text{attn}}^i V^i$$

(5)

Finally, both the attended features are flattened and separately connected to a fully connected layer, leading to a softmax over the concepts. In practice, we also down-sample the dimensionality of $f_{\text{vai}}^i$ using 1 + 1 convolutions before we aggregate and flatten the features.

$$y_{\text{common}} = \text{softmax}(W_{\text{vai}} f_{\text{vac}} + b_{\text{vac}}).$$

(6)

$$y_{\text{independent}} = \text{softmax}(W_{\text{vai}} f_{\text{vai}} + b_{\text{vai}}),$$

(7)

where $y_{\text{common}}$ is the network prediction for the common concept and $y_{\text{independent}}^s$ is the independent concept prediction for the $s^{th}$ index in the concept batch. Our surrogate loss is the sum of common concept cross-entropy loss and the concept-batch averaged independent concept cross-entropy loss.

<table>
<thead>
<tr>
<th>Method</th>
<th>Visual Genome</th>
<th>RefeIt (mask)</th>
<th>RefeIt (bbox)</th>
<th>Flickr30k</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random baseline</td>
<td>11.15</td>
<td>10.48</td>
<td>24.30</td>
<td>27.24</td>
</tr>
<tr>
<td>Cross baseline</td>
<td>20.55</td>
<td>17.04</td>
<td>30.40</td>
<td>49.20</td>
</tr>
<tr>
<td>VGG baseline</td>
<td>18.04</td>
<td>15.04</td>
<td>29.98</td>
<td>35.37</td>
</tr>
<tr>
<td>Fang et al. [11]</td>
<td>14.03</td>
<td>23.93</td>
<td>33.52</td>
<td>29.03</td>
</tr>
<tr>
<td>Zhang et al. [8]</td>
<td>19.31</td>
<td>21.94</td>
<td>31.97</td>
<td>42.60</td>
</tr>
<tr>
<td>Ramamurthy et al. [8]</td>
<td>24.80</td>
<td>24.80</td>
<td>-</td>
<td>50.10</td>
</tr>
<tr>
<td>Xiao et al. [12]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Semantic self-supervision</td>
<td>30.03</td>
<td>29.72</td>
<td>39.98</td>
<td>49.10</td>
</tr>
</tbody>
</table>

Table 1: Grounding evaluation using the pointing game (left) and the ablation analysis of different surrogate losses while varying the concept batch size (right).
4 Results and analysis

Results. We report the comparison of our method with the baselines and previous methods in this section. Table 1 summarizes the performance of our best model on the three different datasets. To highlight our generalization, we train our model on Visual Genome since it’s the largest dataset out of the three and directly evaluate on the other two without fine tuning. Surprisingly, the VGG16 baseline fares decently well even though it does not take any phrase-related information into account owing to the fact that many phrases refer to commonly occurring objects. The center baseline performs very well for Flickr30k showing the centered nature of the phrases with respect to spatial location. Our model outperforms all baselines on Visual Genome and Refer-It and is just 1% less than the state-of-art work of [8] on Flickr30k.

Concept batch size and surrogate loss. We perform ablative studies on the two loss terms and the concept batch size $k$. We use the shorthand $IC$ (independent concept only), $CC$ (common concept only) and $ICC$ (independent and common concept) for the three loss types from Table 1. We train our model with the $IC$ and $CC$ loss separately, keeping everything else in the pipeline fixed. For all three settings, we vary the concept batch size $k$ and observe some interesting trends. For a fixed loss type, the performance increases as we increase $k$, the CC loss being the exception to this trend. The performance for $CC$ loss increases up to $k = 7$, but goes down with $k = 9$. This points to a very common problem with self-supervised techniques where the model finds a shortcut to reduce the loss value without improving on the performance. With only the common concept loss, the network can learn a majority voting mechanism such that not all $k$ concept representations are consistent with the common concept and the loss can be optimized even though some instances aren’t grounded correctly. This is corroborated with the fact that we also observe a faster convergence of $CC$ loss for $k = 9$ than the lower concept batch sizes. These results empirically highlight the importance of the $IC$ loss term as a regularizer and also highlight the usefulness of our concept batch formulation since it improves performance.

Performance variation across concepts. To better understand the variation in performance across the chosen concepts, we also compute the performance across each of the 2000 concept classes and observe variation in performance. We investigate two possible causes for this variability. The first is the average bounding box size associated with each of these concepts for which we expectedly find a strong positive correlation ($\rho = 0.85$), explaining the lower performance for concepts like ‘screw’ and ‘doorknob’. The second is the existing knowledge of concept labels present in the ImageNet classes which our model obtains through the VGG16 based visual encoder. For computing the correlation of concept performance with the knowledge from ImageNet classes, we use a trained word2vec model [6] and compute the maximum similarity of a particular concept across all the ImageNet classes. We find no noticeable correlation between the ImageNet similarity of our concepts and their performance ($\rho = -0.02$). This further strengthens the case for our approach since our concept performance isn’t biased towards the ImageNet labels.

Improvement over a noun-based concept detector. We also conduct a simple experiment to verify that the model isn’t simply working as a noun-based concept detector instead of modeling the complete phrase. For this, we replace the full phrase with a single noun, randomly sampled from the phrase, as the input to the textual encoder. We note a 4.7% drop in performance on Visual Genome. Since training of the original model enforces only concept-level discrimination, it’s interesting to see the presence of complete phrases being useful for model performance and shows that our model learns more than just word-level concepts.
References


