WEAKLY SUPERVISED LEARNING FROM REFERRING EXPRESSION: CHALLENGE AND DIRECTIONS

BY

TAIYU DONG

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Adviser:

Associate Professor Derek Hoiem

ABSTRACT

We explore methods of weakly supervised learning from referring expression. Unlike traditional fully supervised semantic segmentation of object recognition tasks, in which a a small set of discrete class bases is provided, the referring expression task is performed associated with a sentence phrase, *e.g.* "*the dude on the dolphin*". Previous approaches use LSTM and fully convolutional network and have fairly good results under fully supervised setting. However, the fully supervised setting is limited by manual labeling of segmentation masks, which requires a significant amount of human labor. Therefore, we work on an approach to perform segmentation with only image level language descriptions. Under our weakly supervised setting, we are only provided with input images and the corresponding sentence descriptions, without the pixel level labeling for each image as ground truth. In order to get supervision only from language description, we utilize the multiple instance learning loss. We first develop an end-to-end model to localize the image content corresponding to the language expressions. In this model, we use GloVe and ELMo sentence embeddings to get a vector representation for each sentence and combined with image features from a fully convolutional network. However, the sentence level model is hard to interpret hence we also study a more fundamental problem of weakly supervised object localization from referring expressions. We compare the performance of the sentence level model on this task to an alternative word-level model. Our investigation suggests that breaking the referring expressions localization problem into smaller more manageable components is promising.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Image segmentation is a core problem in computer vision. In traditional image segmentation problem, each given image represents an object from a small set of object classes. However, a small set of classes is insufficient to represent all concept present in images. Therefore we are interested in using object localization from referring expressions. More specifically, our question is that if an image and its corresponding sentence description of some image content are given, can we perform segmentation or localization on the image upon the input text query. For example, the input text could be either *“person on raft”* or *“anywhere on the group of people”*.

Previously this problem has been studied using fully supervised method. In fully supervised method, each input data has its the correct segmentation mask for training. However the cost of using human labor to generate segmentation masks is relatively expensive, since people need to read the whole sentence before they mark and find the corresponding contents on the image.

In this paper, we present the results of using weakly supervised learning localization methods to solve this problem. Under our setting, we do not use the pixel-wise ground truth segmentation masks while training the model. The only visible information is a pair of sentence and the corresponding image. At this point it has became a weakly supervised learning localization problem. We develop two models and training techniques for the weakly supervised localization learning problem:

- **Sentence Localization Model** we present an end-to-end model by adjusting from the model Hu *et al.* [4] propose. For a given input data with an image and an associated sentence, we generate an image embedding matrix using FCN[8] and combine it with the associated sentence embedding vector. After we combine image feature and sentence feature, we apply multiple instance learning loss[8] on top of it.

- **Word Localization Model** Our result shows the sentence localization model is not only having a hard time to localize the correct object, but also not interpretable and hence hard to debug. We then simplify our task into a weakly supervised word localization problem. We first extract edge boxes[13] as region proposals for all the input image, and then train separate classifiers for all noun words that ever appear in the sentences. Therefore each noun word has a corresponding classifier when the entire training process ends. The classifier will give the predicted edge box that contains the noun word inside. In inference time, given an input image with a sentence, we locate the correct edge box on the image by extracting the noun phrase via constituency parse and applying the corresponding noun classifier.

Our contributions are as follows: (i) Extend existing fully supervised model for referring expression localization to the weakly supervised case using multiple instance learning. (ii) Propose a soft version of multiple instance learning loss and a size prior to improve stability. (iii) Study the more fundamental problem of weakly supervised noun word localization from referring expressions. Additionally, we also study the roles of negative region sampling, ensemble, and dimension reduction of image features.

CHAPTER 2: BACKGROUND AND RELATED WORK

We begin by discussing prior work on fully and weakly supervised object localization and image segmentation with object labels. Then we review fully supervised phrase grounding problem.

Fully Supervised Localization In fully supervised method setting, semantic segmentation and object detection have been studied systematically. In object detection task[16, 17], region proposals or region proposal network is used to localize object and shows great performance. In semantic segmentation task[7], fully convolutional network[7] has been widely used by adjusting the network architecture from VGG[11] networks. Mask R-CNN[18] gives an end-to-end segmentation model extended from faster R-CNN[16]. Dilated convolution[18] aggregates multi-scale contextual information without losing resolution and also shows great performance on semantic segmentation task.

Weakly Supervised Localization To reduce the expensive annotation cost, weakly supervised object localization uses higher level label information. For example, we can only use image-level label during the training process. Most of weakly supervised object localization approaches[20, 21, 22, 23] take the advantage that discriminative features tend to appear in one class with higher frequency than other classes. Pathak *et al.*[8] propose a way to perform semantic segmentation in a weakly supervised way. In their paper, each image is feed into a fully convolutional network[7] to get 21 channels heatmaps, where each heatmap is corresponding to one of the 20 fixed classes and an additional heatmap represent the heatmap for the background. A multi-class multiple instance learning loss is defined on top of the 21 heatmaps by maximizing the peak pixel value on the corresponding heatmap. This multi-class multiple instance learning gives us the basic inspiration on the more complicated case of referring expressions. In our approach, we analog each referring natural language expression as a fixed class label, and apply multiple instance learning loss in a similar manner.

Fully Supervised Phrase Grounding In recent years, there are several works on referring expressions in fully supervised setting. Hu *et al.*[4] present a model by using LSTM[6] for sentence embedding and fully convolutional network[7] for image feature extraction. After extracting sentence and image features, an additional neural network is designed to be applied on top of the combined sentence and image features to generate an output heatmap. In the fully supervised setting, they make full use of the ground truth segmentation mask and apply weighted binary cross entropy loss on the output heatmap. Furthermore, Liu *et al.*[5] further improve the result by adding an additional LSTM on the sentence embedding and combine the output of LSTM with the extracted image feature.

In chapter 3 and chapter 4, we will explain two proposed models and how different training strategies affect the results on ReferIt game[3] dataset.

CHAPTER 3: SENTENCE LOCALIZATION MODEL

3.1 APPROACH

This approach is an end-to-end model inspired by the fully supervised model proposed by Hu *et al.*[4]. We call it sentence localization model because we encode the entire referring expression in the model. Under the weakly supervised learning setting, we are not allowed to use pixel-level ground truth segmentation masks.

In our model, when we have an input pair of image and natural language expression, we first extract an image feature map and a language feature vector. Then we combine image and language features before a fully convolutional classification network and an upsample network. The final output will be a pixel-wise segmentation mask. The model can be interpreted as shown in Fig. 3.1.

Since the weakly supervised setting makes the training process difficult in our experiment, we also add PASCAL VOC 2012[1] dataset and use it in a fully supervised manner. We expect fully supervised learning from PASCAL VOC data set to help a great guide the weakly supervised learning on ReferIt[3] dataset. The single class word from PASCAL VOC will be treated as the natural language expression in our model.

3.1.1 Image Feature Extraction

Given a single image with RGB channels, we want to find a feature representation of this image. We simply forward the single image of size $H \times W \times 3$ into a fully convolutional network, more specifically, FCN-32s[7]. The output of feature representation will be in size of $h \times w \times D_{im}$. Since the input image may have different input size, we make zero padding around the image to make all input image with size $512 \times 512 \times 3$.

In our implementation, The FCN-32s is adopted from VGG-16 network architecture[11], the fc6, fc7 and fc8 layers are modified as fully convolutional layers. The output dimension $D_{im} = 1324$ in our case, the number 1324 is from 1024 from ELMo embedding and 300 from GloVe embedding, which we will explain in section in section 3.1.2.

3.1.2 Encoding Natural Language

To simplify the model complexity while maintaining the explained information from the sentence, we use fixed sentence embedding in our model instead of an LSTM network. Peters *et al.*[9] claim that adding ELMo embedding will increase the speed of convergence while maintaining the text information. For each sentence, we concatenate the ELMo sentence embedding vector and GloVe[10] embedding vector to get our sentence embedding vector. Since ELMo sentence embedding vector is trained using a 2-layer LSTM network, it helps maintaining the sentence structure.

For the input of natural language expression, we use two different pre-trained language embedding methods. Firstly, in the ELMo sentence embedding model, We pick the second LSTM layer output from it, and get a sentence embedding vector with dimension $D_{elmo} = 1024$. Secondly, we get a GloVe embedding vector for a single sentence by the average values of each word embedding vectors. We use pre-trained GloVe with 300 dimensions, hence $D_{glove} = 300$. Then the final sentence embedding vector is created by concatenating these two vectors described above, $D_{text} = D_{elmo} + D_{glove} = 1324$.

3.1.3 Classification and upsampling

After extracting the images feature map of size $h \times w \times D_{im}$, and the natural language encoded vector of size D_{text}, we combine the image and language features and classify each pixels as a foreground or background.

To make the shape of image feature and language feature consistent, we tile the text embedding to the size of $h \times w \times D_{text}$, note we have $D_{text} = D_{im} = 1324$. After making the size consistent, we perform a dot product across the third dimension between the images feature map and tiled text encoder matrix. Specifically, we perform elementwise product on the $h \times w \times D_{im}$ image feature and $h \times w \times D_{text}$ text feature, then sum across the third dimension, this will result into a matrix of size $h \times w$.

After getting the $h \times w$ feature map, we add two bi-linear up sampling layers on top of it. The first layer is with scaling factor 4 and the second layer is with scaling factor 8. There is also a one-to-one convolutional layer between the two upsampling layers. Finally, we complete the entire model graph with a heat map of size $H \times W$, which is the same size as the padded input image.

3.1.4 Loss Functions

Once we get the final $H \times W$ heat map for each data, we apply the MIL loss on top of it. There are two different types of MIL loss function we define to perform the weakly supervised learning task. For clarity, we denote P as the $H \times W$ heat map, and P_{ij} represents for the entry in i-th row and j-th column.

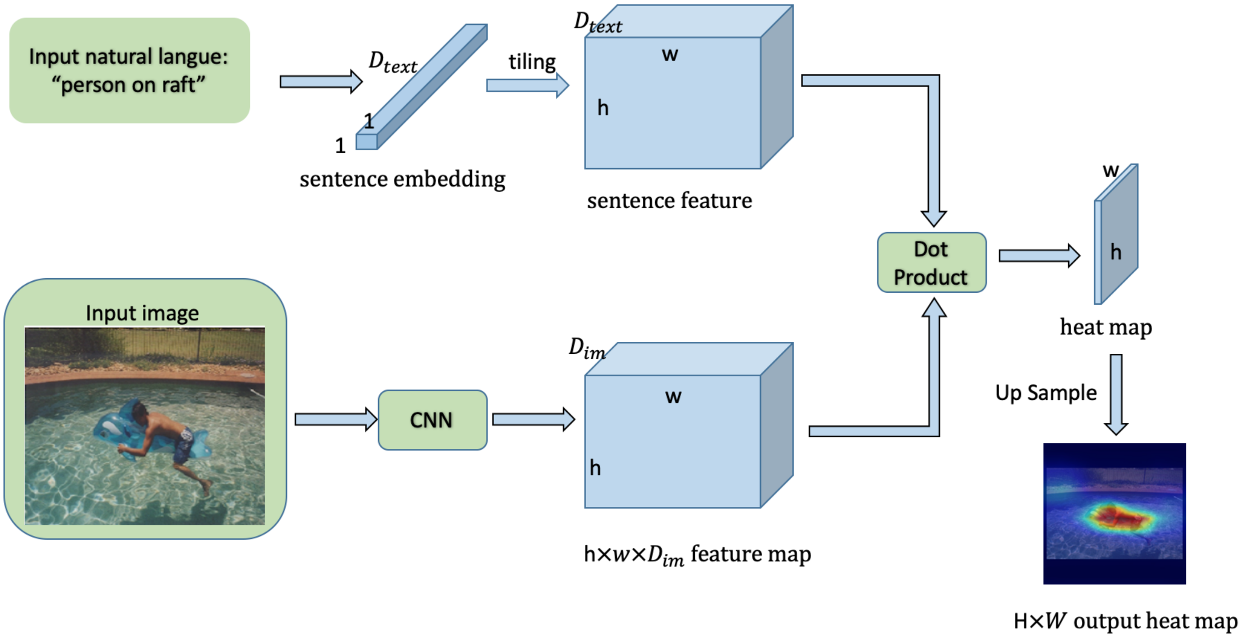

Figure 3.1: sentence localization model for weakly supervised segmentation from natural language expressions.

3.1.4.1 Hard MIL Loss

The MIL loss is inspired by the Deepak *et al.*[8]. On the $H \times W$ heat map, we pick top k_f pixel values as foreground pixels, and the lowest k_b pixel values as background. We denote the set of pixels as has top k_f values as S_{kf}, and similarly for S_{kb}. Based on the MIL loss, we want to maximize the foreground pixel values in S_{kf} and minimize the background pixels values in S_{kb}. The reason to use k_f and k_b instead of only maximizing the peak value on the heatmap is to increase the stability of the model. In the case, as long as the actual true foreground pixel appears in S_{kf}, the learning will towards the right direction as we increase the correct pixel value. In our setting, $k_f = 3$ and $k_b = 1000$ are used to achieve better performance.

$$\text{Hard MIL Loss} = L_{hard} = -\frac{1}{k_f}\sum_{p^+ \in S_{kf}} \log(p^+) - \frac{1}{k_b}\sum_{p^- \in S_{kb}} \log(1 - p^-) \tag{3.1}$$

The loss is saying if a pixel in the final heatmap has a high value as in S_{kf}, we will maximize it to reduce the loss. Similarly, if a pixel in the final heatmap has a low value as in S_{kb}, we will minimize it to reduce the loss.

3.1.4.2 Soft MIL Loss

To even further increase the stability of the model, we adjust the loss function to be in a soft version manner. In this case, instead of only increasing the top k_f pixel values and decreasing lowest k_b, we use model predicted probability as soft labels. More precisely, a pixel with a higher value will have a higher weight to increase and vice versa. The weight above is defined by the current value on the pixel. However, we are not going to backpropagate through this weight while training.

$$\text{Soft MIL Loss} = L_{soft} = -\sum_{i,j} W_{ij} \log P_{ij} - \sum_{i,j} (1 - W_{ij}) \log (1 - P_{ij}) \tag{3.2}$$

where the value $W_{ij} = P_{ij}$.

3.1.4.3 Prior Restriction on Object size

Based on the ReferIt dataset, we found there are roughly average 15% of pixels are foreground and 85% of pixels are background. Therefore, we add a prior constraint to the final loss to ensure there are roughly θ_{prior} percent of pixels are foreground. In our setting, θ_{prior} is set to be different values as $\{0.1, 0.15, 0.2, 0.25, 0.5, 0.75\}$

$$\text{Prior Restriction} = L_{size} = |\frac{1}{WH} \sum_{i,j} P_{ij} - \theta_{prior}| \tag{3.3}$$

3.1.4.4 Final Loss Total

Therefore, as we discuss from 3.1.4.1 to 3.1.4.3, final total loss function is defined as:

$$\text{Total Loss} = L_{total} = L_{hard/soft} + L_{size} \tag{3.4}$$

The MIL Loss can be either hard MIL loss or soft MIL loss described in section 3.1.4.1 and 3.1.4.2.

3.2 EXPERIMENT AND RESULTS

In the training process, we use the VGG-16 pre-trained weights on Imagenet 1000 classes classification challenge as the initialization weights on the FCN image feature extraction. The entire FCN weights are trained through the entire process.

In the evaluation time, once we have the final $H \times W$ heap map for a single data, we apply a sigmoid function on each of the pixels and choose a classification threshold value θ to mask out the foreground pixels. More specifically, we mark all pixels with a value greater than θ as foreground and vice versa.

The evaluation metric used is intersection over union(IoU) between the predicted segmentation masks and the ground truth segmentation masks. In order to fully explain the result, we both compute IoU values for foreground and background. To compute the IoU for background, we compute the IoU between all predicted background region and the ground truth background region. We compare and explore the effect of different choice of MIL loss functions as well as difference choice of prior values.

3.2.1 **Effect of Losses**

Table 3.1 shows the computed IoU evaluation metric on the validation set with different choice loss functions. Prior threshold $\theta_{prior} = 0.15$ and classification threshold $\theta = 0.15$ are used in this test.

	Intersection over Union (IoU)	
MIL Loss Type	IoU Foreground(%)	IoU Background(%)
Hard MIL	14.19	64.53
Soft MIL	14.82	84.46

Table 3.1: Intersection over Union on validation set when using different type of loss function. IoU is both on foreground and background.

Based on the IoU results from the validation set, we can see that Soft MIL loss and prior threshold help with the performance on the weakly supervised segmentation task. This is because when we use hard MIL loss, if a ground truth foreground pixel is not in S_{kf}, the model may not be updated towards an expected direction since it will not increase the probability being a foreground. However, the soft MIL loss always keep a weight that the true foreground value needs to be increased.

3.2.1.1 Effect of Prior Restriction

Table 3.2 shows the IoU evaluation metric on the validation set with different combination of θ_{prior} and θ. From the IoU numbers in the table, $\theta_{prior} = 0.2$ and $\theta = 0.1$ give the best IoU foreground on the ReferIt it validation data.

	classification threshold		
prior threashold	**0.10**	**0.15**	**0.20**
0.10	0.161	0.156	0.144
0.15	0.171	0.152	0.130
0.20	0.172	0.166	0.154
0.25	0.150	0.145	0.137
0.50	0.134	0.128	0.123
0.75	0.160	0.161	0.165

Table 3.2: Foreground intersection over Union on validation set when using different values of prior threshold and classification threshold.

3.2.2 Segmentation Results

The model gives some reasonable results showing in Fig. 3.2. But for examples show in Fig. 3.3, it still has a hard time localizing the correct objects in the images.

The model is more likely to predict people as foreground no matter what the input natural language is. This is very likely caused by the fully supervised part from PASCAL VOC dataset as there is a class called "people" in PASCAL VOC and the full supervision takes most part of the learning. However, if we remove the fully supervised part on PASCAL VOC data set, the result outputs become totally random and unpredictable.

In general, this end-to-end sentence localization model using MIL loss does not perform as we expected. Our best foreground IoU is 17.2%, which does not show a significant improvement when we use the baseline strategy, which has IoU 15%. And the baseline method is simply predict everything as foreground. This is mainly because of the noisiness of the input sentence encoding vector. In the case of multi-class multiple instances learning proposed by Pathak *et al.*[8], the study is on PASCAL VOC data set. However, the natural language sentence encoded vector lies on a continuous space. Therefore we need to even further simplify the problem in order to get better performance in the weakly supervised setting.

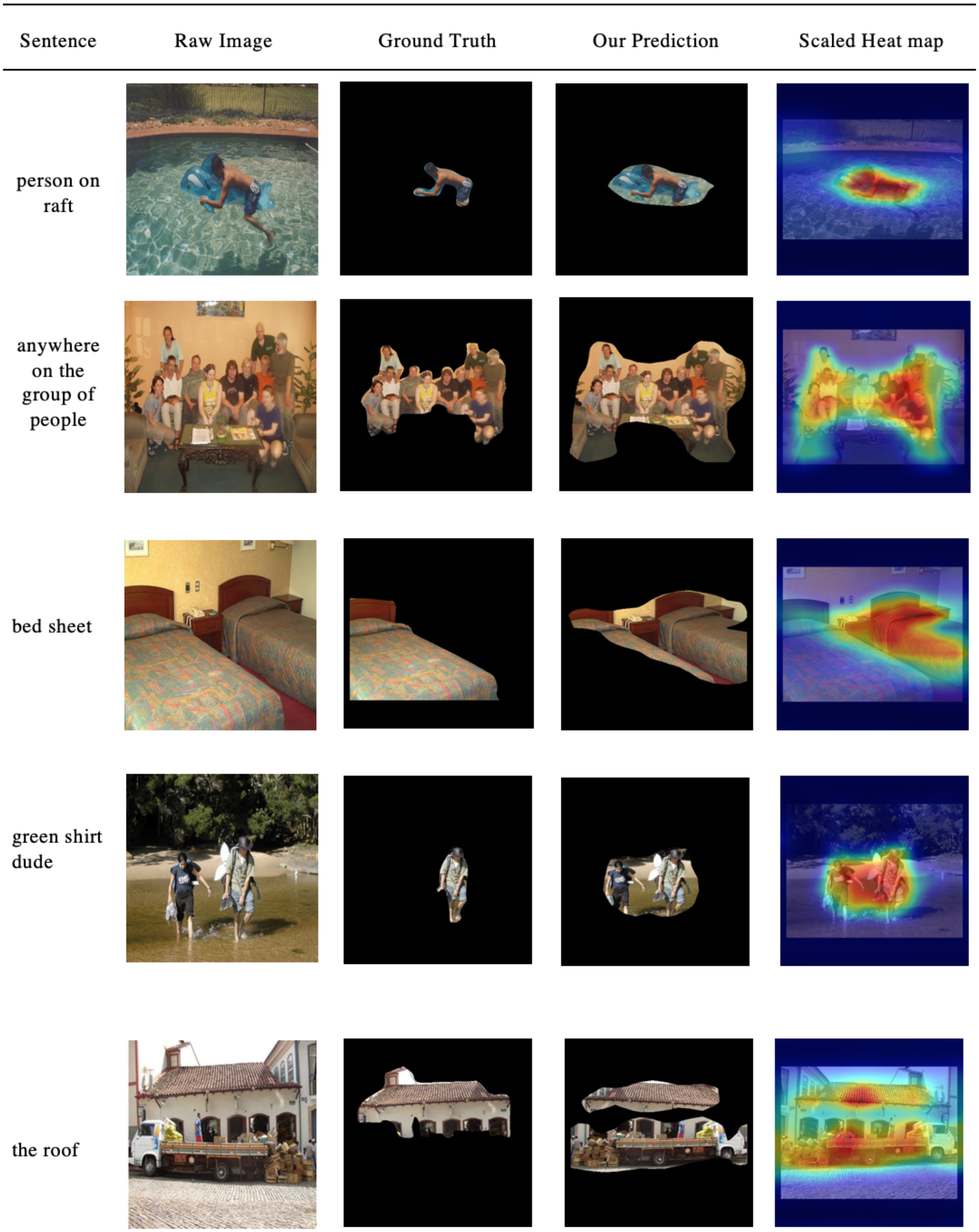

Figure 3.2: positive localization results.

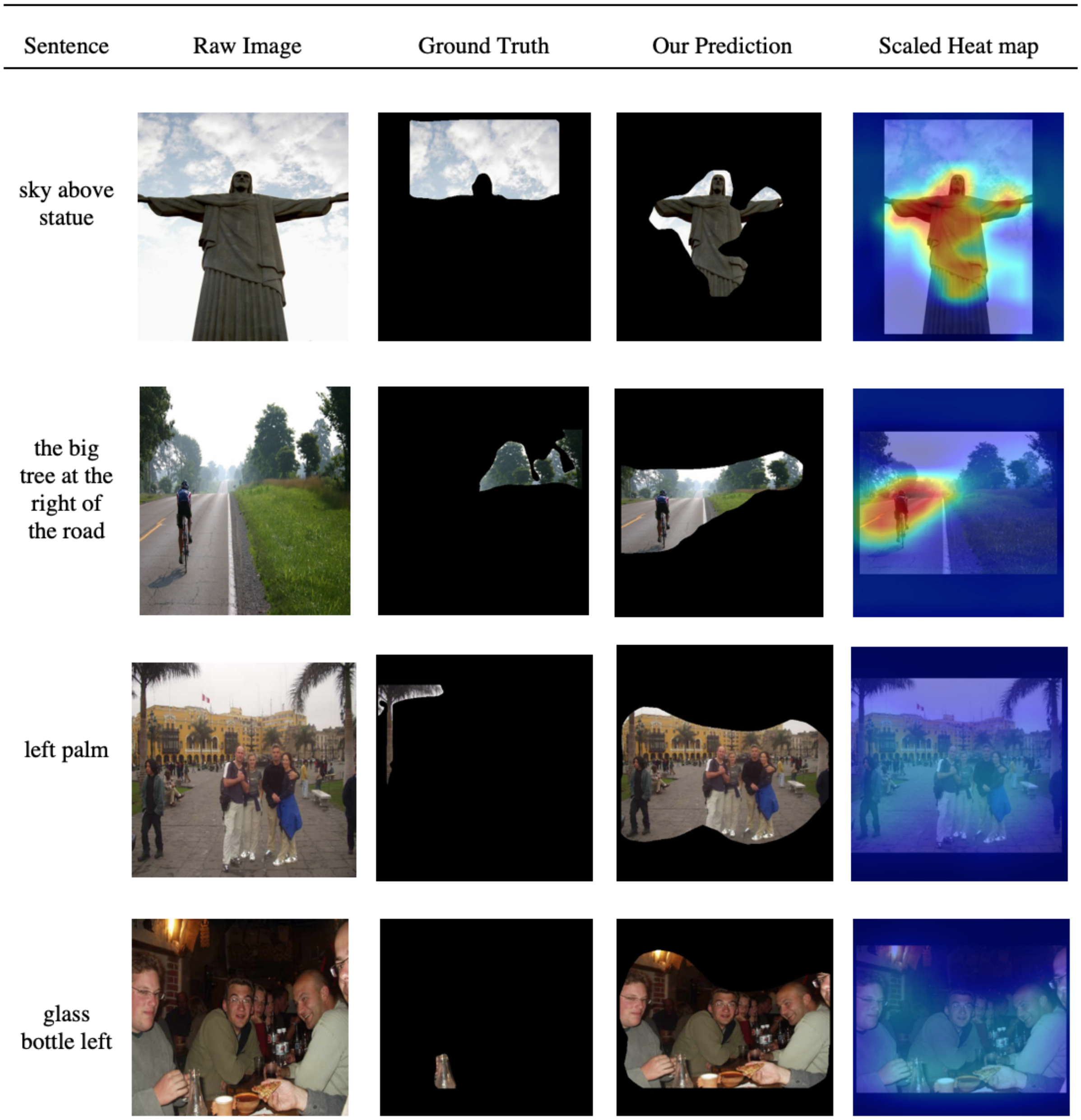

Figure 3.3: failure cases.

CHAPTER 4: WORD LOCALIZATION MODEL

4.1 APPROACH

The sentence localization model does not seem to localize correct objects in the image. And the model is not interpretable and the source of failure is hard to diagnose and fix. One possible reason is that the natural language encoding vector lies on a continuous space, and can be hardly analog to the work proposed by Pathak *et al.*. For example, for a single image with same ground truth segmentation, the sentence can either be "person in front with white shirt", or "white shirt guy", sometimes even more different ways. In this case, the output heat map P may vary a lot while the input images and segmentation ground truth are the same.

To overcome this fundamental challenge caused by the sentence localization model, we develop a new model that completely discards the language encoding. We use POS Tagger[12] to extract all noun words in each sentence and then perform multiple instance learning on each noun word separately. To even further reduce the complexity of the problem, we want to first solve object localization from natural language, instead of performing segmentation task directly. To perform object localization task, we need to extract region proposals for each image before apply classification on those regions.

The overview of the new procedure are: (i) Extract all noun words among all sentences. (ii) Extract region proposals for each image. (iii) Train individual classifiers for each noun word. (iv) In inference time, given an image and a sentence, find the most important noun word or noun phrase, and use the classifier associated with the noun word to find the most relative region of proposal in the image.

4.1.1 Extracting Noun Words

To extract the noun words, we use Stanford Log-linear Part-Of-Speech Tagger(POS Tagger)[12] to select all NN and NNs for each sentence and then save the indexes of NN and NNs in a dictionary.

4.1.2 Extracting Region Proposals and Image Features

For each image, we need to extract region proposals that contain objects. Laurence *et al.*[13] propose edge box extraction, which has showed significant performance in selecting objects in the image. Taking advantage of this previous work, we extract 100 edge boxes from each of the raw images, then we feed the selected edge box regions into ResNet152[14] network pre-trained on Imagenet classification task. To extract the image features, we output the feature vectors before feeding to the final classification layer in ResNet152. Since the size and shape of the edge box may vary, we resize and pad all edge boxes to be with a fixed shape of 224 × 224 before feeding into ResNet152. After the feature extraction, for every single image, we will have 100 number of feature vectors with size D_{im}. Based on the architecture of ResNet152, $D_{im} = 2048$ is the default value.

4.1.3 Multiple Instance Learning on Edge Boxes

Section 4.1.1 and 4.1.2 describe how we get pre-processing data. Now we need to find a way to train classifiers for each noun separately. What the classifier tells is that among all the given 100 edge boxes, which of these is containing correct content related to this noun word. For simplicity, we will use "dog" as an example of the single noun word we are training.

As we are training on the noun word "dog", we will only look at all the sentences that contain "dog" and discard all the other data for now. We denote the set of all images that have a natural language description containing "dog" as S^{+}_{dog}. On the other hand, we construct another set of images that none of their natural language descriptions contains "dog", denoted by S^{-}_{dog}.

To prevent from overfitting, we keep the model as simple as a single hidden layer neural network. Fig. 4.1 shows the flow graph of our model on a single data a "dog" classifier. In the figure, the edge box feature vectors are with size D_{im} and they are fed into a multi-layer perceptron(MLP) with only one hidden layer. Each edge box input will get a single probability interpreted as how likely it contains a "dog" or not. The negative feature vectors are edge box feature vectors that are randomly sampled from the image in S^{-}_{dog}. In our experiment, we sampled a total amount of 1000 random edge boxes that from S^{-}_{dog}.

4.1.3.1 Loss Functions

Let us take a single training data in S^{+}_{dog}. Since there are 100 edge box feature vectors, the input matrix feed into MLP is of size $100 \times D_{im}$, hence the output will be a vector of size 100×1 with probabilities saying how likely each edge box contains a dog, denoted by P^{+}. Similarly, the negative feature vectors input is of size $1000 \times D_{im}$ and the output is a vector with size 1000×1, denoted by P^{-}.

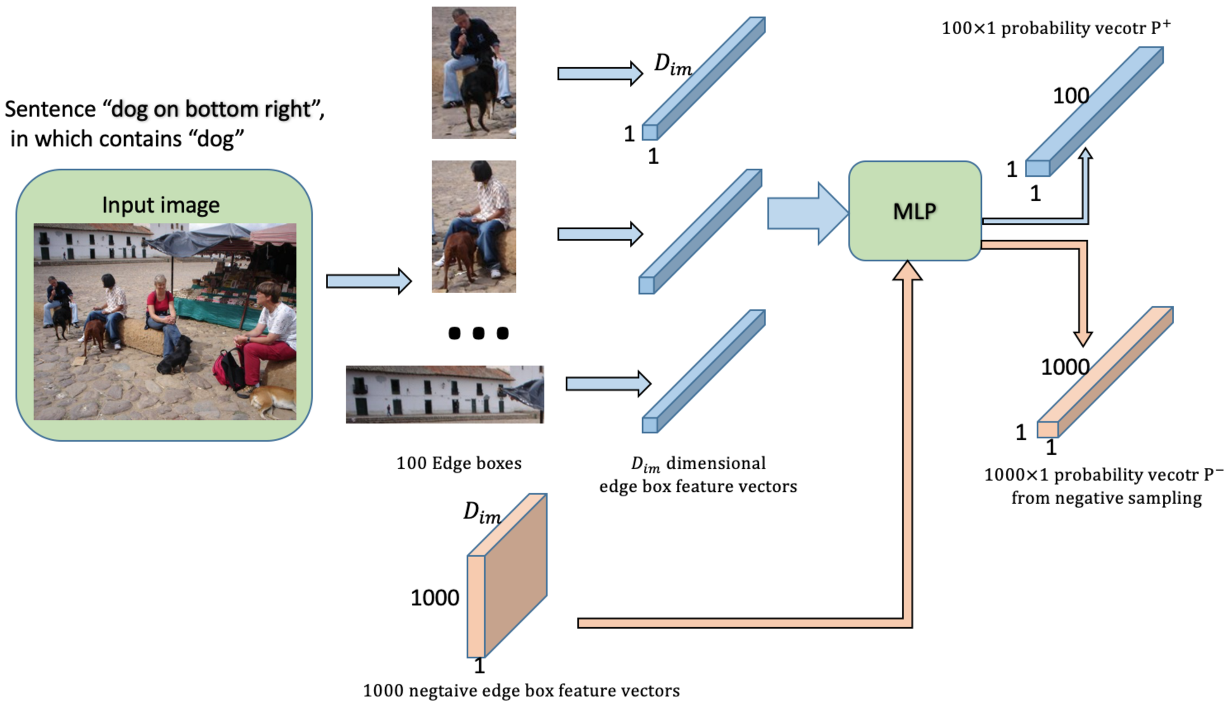

Figure 4.1: Our model for edge box classification on a single noun word "dog".

MIL Loss. The key information we know is that at least one of the edge box feature in the S^+_{dog} contains a dog, and none of the edge box in the S^-_{dog} has a "dog". Now we can utilize our multiple instance learning loss function by

$$i^* = \operatorname*{argmax}_i P^+_i$$

$$\text{MIL LOSS} = -\log P^+_{i^*} - \sum_{i \neq i^*} \log\left(1 - P^+_i\right) - \sum_i \log\left(1 - P^-_i\right) \tag{4.1}$$

where the subscript i denotes the i-th value in the vector P^-_i or P^+_i. The MIL loss is basically saying that pick the highest value in P^+_i as the potential edge box that contains a "dog" and maximize its probability while minimizing all the other probabilities from edge boxes in P^+_i and P^-_i .

Conservative MIL Loss. However, one problem of defining MIL loss using the way above is that we also minimize other probabilities in P^+_i. In the setting of weakly supervision, it is common that the model misses the true edge box that contains the true object "dog". Once it misses the true edge box, the MIL loss defined above will also minimize its probability, which may make the model even more difficult to find the correct edge box. Therefore, we remove the P_i^+ part and define a more stable version of MIL loss, called conservative MIL loss. In this case, we are not going to reduce the probability of the true edge box even though it does not get the maximum probability.

$$i^* = \underset{i}{\operatorname{argmax}}\ P_i^+$$

$$\text{CONSERVATIVE MIL LOSS} = -\log P_{i^*}^+ - \sum_i \log\left(1 - P_i^-\right) \tag{4.2}$$

4.1.3.2 Feature Ablation

We extract the edge box feature by feeding the edge box into ResNet152 pre-trained on the Imagenet classification problem. To make it easier for multiple instance learning to discover useful image features, we perform dimension reduction to get rid of redundant features.

To reduce the dimension, we define an encode weight vector W_{en} of shape 128×2048 and a decode weight vector W_{de} of shape 128×2048. Let S_{eb} denote the set of all edge boxes and we minimize

$$\text{Loss}_{recon} = \frac{1}{|S_{eb}|} \sum_{X_{im} \in S_{eb}} |W_{en} X_{im} W_{de}^T - X_{im}^T| + |W_{en}| + |W_{de}| \tag{4.3}$$

The absolute values on W_{en} and W_{de} are used by regularization. After the Loss_{recon} converges, we construct the low dimensional feature vector by computing $W_{en} X_{im}$, the new low dimension D_{im} is now 128 instead of 2048.

4.1.3.3 Model Ablation

Even though we propose all setting above to improve the stability of multiple instance learning tasks, a single multi-layer perceptron is still easy to have random performance since it is easy to latch on to incorrect regions. To increase the model robustness, we add some parallel MLPs with random initialization. These parallel MLPs perform in an ensemble way to make the entire model stable.

Assume we have k different MLPs with same structure, and the outputs probabilities from the k MLPs are denoted by $P^{1+}, P^{2+}, ..., P^{k+}$, and $P^{1-}, P^{2-}, ..., P^{k-}$. The final P^+ and P^-

are defined as

$$P^{+} = \frac{1}{k} \sum_{i=1}^{k} P^{i+} \tag{4.4}$$

$$P^{-} = \frac{1}{k} \sum_{i=1}^{k} P^{i-} \tag{4.5}$$

And the MIL loss or conservative MIL loss are still computed as described in 4.1.3.1.

4.2 EXPERIMENT AND RESULTS

Once we finish training, each noun word will have its own classifier. In the inference time, when evaluating a single data, we use a pre-trained constituency parsing model from ELMo[9] to extract the noun words. More precisely, given a sentence, we recursively find the last noun phrase, then treat all NN and NNs in this final noun phrase as the nouns we want to look at. To simplify the evaluation process, we create separate evaluation sets for each noun word. For example, we collect all data that have the word "dog" appears in the last noun phrase, denote S_{dog}^{eval}. However, since there are around 8000 nouns in the word dictionary, we only train 97 noun words that have the highest frequency in the entire ReferIt[3] dataset.

The evaluation metric is the recall value at top $\{1, 5, 30\}$ edge boxes. We consider the prediction is true if among the k edge boxes that with highest predicted probabilities, there is at least one of edge boxes has IoU value greater than some threshold with the ground truth bounding box. In order to find the best model for our problem, we then explore the effect on different feature dimensions, loss functions and number of MLPs being used.

4.2.1 Effect of Dimension Reduction

Table 4.1 shows the top 1, top 10 and top 30 recall values at different threshold level using only single MLP and MIL loss function setting. For example, in the section recall@1, we compute top 1 recall values with threshold 0.1, 0.2, and 0.5. That is, if the IoU between predicted edge box and ground truth is greater than a threshold, we consider it as correctly classified.

From the reported values in the table, it can be observed that with 2048 dimension edge box feature vector, the performance is significantly better. Therefore, our auto encoder-decoder dimensional reduction method loses important features and is not going to get better results.

	recall@1			**recall@10**			**recall@30**		
Feature Dimension	**0.1**	**0.2**	**0.5**	**0.1**	**0.2**	**0.5**	**0.1**	**0.2**	**0.5**
128	0.19	0.11	0.03	0.61	0.42	0.13	0.84	0.70	0.30
2048	0.44	0.28	0.06	0.80	0.64	0.24	0.90	0.78	0.42

Table 4.1: top 1, top 10, top 30 recall value at IoU threshold 0.1, 0.2, 0.5 using different edge box feature dimension.

4.2.2 Effect of Ensemble

In this section, we want to look at how the number of MLPs affects the final result. Table 4.2 shows the top 1, top 10 and top 30 recall values at different threshold level using feature dimension $D_{im} = 2048$ and MIL loss to train the models.

	recall@1			**recall@10**			**recall@30**		
Number of MLPs	**0.1**	**0.2**	**0.5**	**0.1**	**0.2**	**0.5**	**0.1**	**0.2**	**0.5**
1	0.44	0.28	0.05	0.80	0.64	0.24	0.90	0.79	0.41
5	0.42	0.27	0.06	0.78	0.62	0.24	0.90	0.78	0.41
10	0.44	0.29	0.06	0.81	0.65	0.26	0.92	0.81	0.42

Table 4.2: top 1, top 10, top 30 recall value at IoU threshold 0.1, 0.2, 0.5 using different number of MLPs.

From the reported values, we can conclude that increasing the number of MLPs can make the entire model slightly more robust and give better results.

4.2.3 Effect of Training on Negative Region in Positive Bag

Finally, we will explore the effect of the loss function, and how MIL loss and conservative MIL loss can affect the results. Table 4.3 shows the top 1, top 10 and top 30 recall values at different threshold using edge boxes feature dimension $D_{im} = 2048$ and 10 number of MLP classifiers.

	recall@1			**recall@10**			**recall@30**		
Type of Loss function	**0.1**	**0.2**	**0.5**	**0.1**	**0.2**	**0.5**	**0.1**	**0.2**	**0.5**
MIL Loss	0.30	0.19	0.04	0.75	0.58	0.20	0.89	0.77	0.39
Conservative MIL Loss	0.44	0.29	0.06	0.80	0.65	0.26	0.90	0.79	0.42

Table 4.3: top 1, top 10, top 30 recall value at IoU threshold 0.1, 0.2, 0.5 using different loss function.

From reported IoU values in the Table 4.3, we observe that the convervative MIL loss is significantly better than MIL loss. This result is also as expected because by using original

MIL loss, it decrease the potential corrected edge box probability value, hence the model is more likely to converge something random. However, the adjusted conservative MIL loss only backpropagate the negative edge boxes, which are total disjoint with any possible correct edge box.

4.2.4 Qualitative Results

Fig 4.2 shows some qualitative sample outputs edge box predictions in the test set using our best model, which is using edge box feature vectors dimension $D_{im} = 2048$, 10 MLP classifiers and conservative MIL loss function definition.

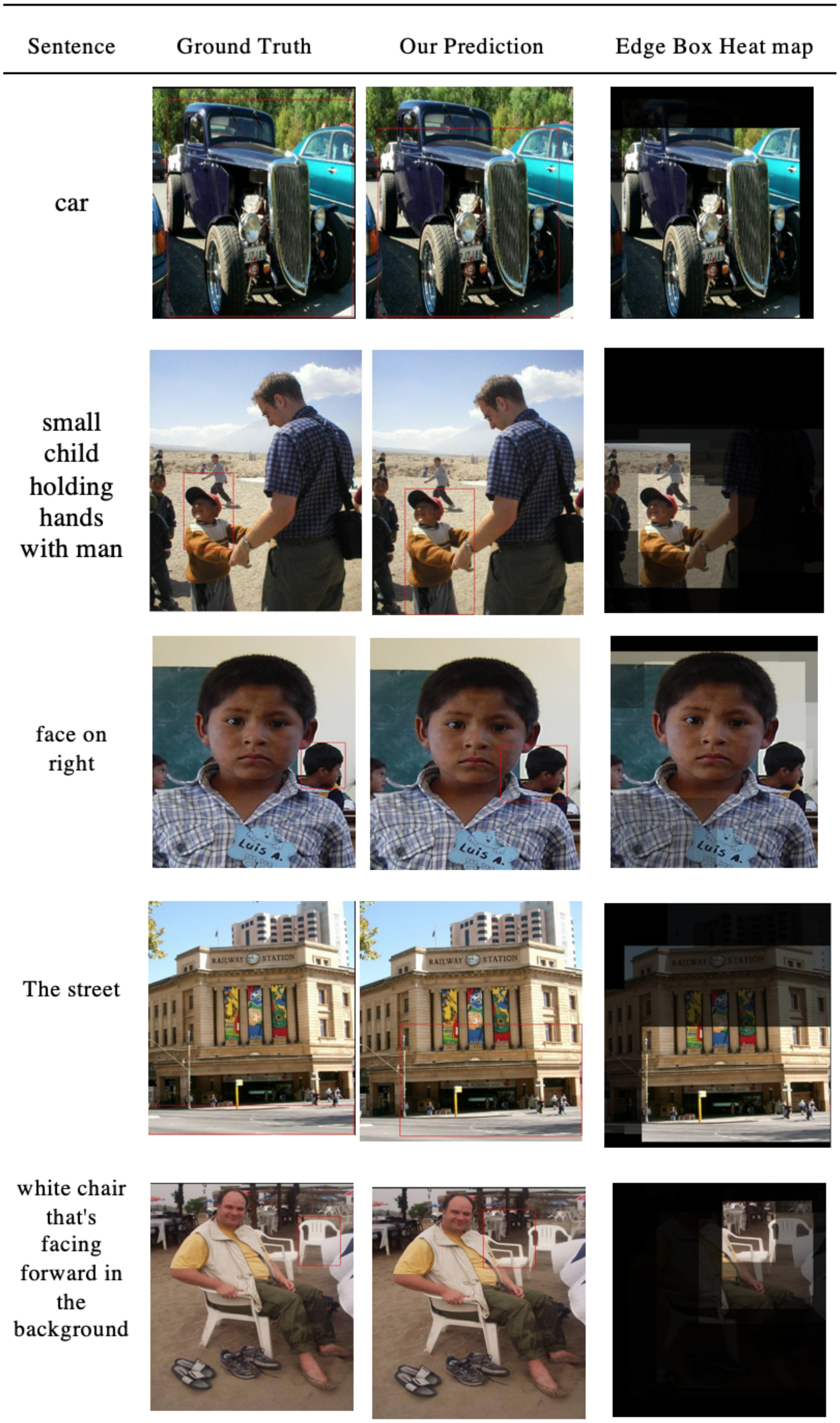
Sentence
Ground Truth
Our Prediction
Edge Box Heat map
car
small child holding hands with man
face on right
The street
white chair that's facing forward in the background

Figure 4.2: Localization results. Predicted edge box and ground truth is marked as red box.

CHAPTER 5: DISCUSSION

5.1 COMPARISON

In our work we use both sentence and word localization method to solve image segmentation problem. For sentence localization method, our best model achieves IoU 17.2%, which is a slight improvement comparing to the baseline strategy with IoU 15%, where the baseline strategy is simply predicting all pixels as foreground. Afterwards, we construct a word localization model to find the bounding box that contains corresponding objects.

To make the two models comparable, we create bounding boxes using the segmentation outputs from the sentence localization model. Specifically, we take the most top, left, bottom, right pixels in the segmentation output and generate a bounding box based on those boundary pixels. We choose the best model from each method and compute the top 1 recall values with different IoU threshold values as described in section 4.2.1.

Based on Table 5.1, when IoU threshold is 0.1, the sentence localization model gives better top 1 recall value. However, the word localization model gives better top 1 recall values at higher IoU thresholds. Therefore, the word localization method tends to give higher quality bounding boxes and thus has better performance on object localization task since the recall values at higher IoU thresholds are higher.

	Recall@1(%)		
Methods	**0.1**	**0.2**	**0.5**
Sentence Localization	50.3	26.6	4.2
Word Localization	44.4	29.3	6.4

Table 5.1: Top 1 recall values by sentence localization model and word localization model.

5.2 LIMITATION

Edge box bottleneck While we use edge boxes as the region proposals, the final selected bounding box is limited by the fixed edge boxes. In the worst possible case, none of the edge boxes captures the actual object in the image.

Sentence bottleneck Since we train and classify the edge box from single noun word, our method may not process complex sentence structure sufficiently accurate. For example, if the input sentence is “the person on the left”, our model will give the edge box that contains “person”, but not necessary to be the person on the left.

CHAPTER 6: CONCLUSION

We propose a weakly supervised end-to-end sentence localization method to perform image segmentation task from natural language expression, in which case we are only given pairs of input image and natural language expression. However, the sentence localization model can hardly correctly localize the correct object in the images. The main reason is that the vector encoding is too high-dimensional and easy to mislead the entire training process.

To simplify the task, we omit the language encoding part and design a word localization model. In this case, same words will be associated with multiple images, which is much easier to train in the weakly supervised setting. We use edge boxes as region proposals and train individual classifiers for each extracted noun words using multiple instance learning loss.

We compare the performance of sentence and word localization methods. The results show that word localization method tends to give better object localization result since it gives higher top 1 recall values. We also find and conclude that for a complicated multiple instance learning or weakly supervised task, we may further divide the task into several simpler tasks to achieve more stable results. In addition, our ablation study also shows the importance of using negative sampling and ensemble techniques on the weakly supervised or multiple instance learning tasks.

REFERENCES

[1] Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results..* `http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html`

[2] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollr, C. Lawrence Zitnick, *Microsoft COCO: Common Objects in Context.* `https://arxiv.org/abs/1405.0312`

[3] Sahar Kazemzadeh, Vicente Ordonez Mark Matten, Tamara Berg, *ReferItGame: Referring to Objects in Photographs of Natural Scenes.* `http://tamaraberg.com/referitgame/`

[4] R. Hu, M. Rohrbach, and T. Darrell. *Segmentation from natural language expressions.* In ECCV (1), volume 9905 of Lecture Notes in Computer Science, pages 108124. Springer, 2016. 1, 2, 3, 5

[5] C. Liu, Z. Lin, X. Shen, J. Yang, X. Lu, A. Yuille. *Recurrent Multimodal Interaction for Referring Image Segmentation.* In ICCV 2017.

[6] S. Hochreiter, J. Schmidhuber. *Long short-term memory.* Neural Computation year:1997 vol:9 iss:8 page:1735 -1780 ISSN:0899-7667

[7] Long, J., Shelhamer, E., Darrell, T. *Fully convolutional networks for semantic segmentation.* In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 34313440

[8] D. Pathak, E. Shelhamer, J. Long, T. Darrell. *Fully Convolutional Multi-Class multiple instance learning.* In ICLR 2015 worshop.

[9] Peters, Matthew E. and Neumann, Mark and Iyyer, Mohit and Gardner, Matt and Clark, Christopher and Lee, Kenton and Zettlemoyer, Luke. *Deep contextualized word representations.* In Proc. of NAACL 2018.

[10] Jeffrey Pennington and Richard Socher and Christopher D. Manning. *GloVe: Global Vectors for Word Representation.* In Empirical Methods in Natural Language Processing (EMNLP) 2014, 1532-1543 `http://www.aclweb.org/anthology/D14-1162`

[11] Karen Simonyan, Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition.* In CoRR 2014.

[12] Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. *Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network.* Proceedings of HLT-NAACL 2003, pp. 252-259.

[13] Zitnick, C. Lawrence, Dollar, Piotr. *Edge Boxes: Locating Object Proposals from Edges*. In ECCV 2014.

[14] Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun. *Deep Residual Learning for Image Recognition*. In CoRR 2015.

[15] Zhirong Wu, Yuanjun Xiong, Stella Yu, Dahua Lin. *Unsupervised Feature Learning via Non-Parametric Instance-level Discrimination*. In CoRR 2018.

[16] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. In CoRR 2015.

[17] Ross Girshick. *Fast R-CNN*. In CoRR 2015.

[18] Kaiming He, Georgia Gkioxari, Piotr Dollr, Ross Girshick. *Mask R-CNN*. In CoRR 2017.

[19] Fisher Yu, Vladlen Koltun. *Multi-Scale Context Aggregation by Dilated Convolutions*. In CoRR 2015.

[20] H. Bilen, M. Pedersoli, and T. Tuytelaars. *Weakly supervised object detection with posterior regularization*. In BMVC 2014.

[21] R.Cinbis,J.Verbeek,andC.Schmid. *Multi-foldMILTraining for Weakly Supervised Object Localization*. In CVPR 2014.

[22] H. O. Song, Y. J. Lee, S. Jegelka, and T. Darrell. *Weakly supervised discovery of visual pattern configurations*. In NIPS 2014.

[23] R. Cinbis, J. Verbeek, and C. Schmid. *Weakly supervised object localization with multi-fold multiple instance learning*. In arXiv:1503.00949, 2015