

# Semi-supervised Image Captioning via Reconstruction

Bicheng Xu    Weirui Kong    Jiaxuan Chen  
University of British Columbia

{bichengx, weiruik}@cs.ubc.ca    jiaxuan.chen@alumni.ubc.ca

## Abstract

Image captioning is an active research area in the intersection of vision and language. However, previous works all require each training image annotated with at least one ground-truth caption. That is, they are all fully-supervised models. In this work, we propose an end-to-end model that can generate image captions in a semi-supervised setting. This means that our model can be trained using an image set with only part of the images annotated with ground-truth captions. We adopt an idea of reconstruction that helps us to use images without paired captions during training. We conduct experiments on the MSCOCO dataset [8] with image annotation rates as 100%, 50%, 25%, and 12.5%. Results show that when image annotation rates are low, our model achieves better captioning results than standard fully-supervised models.

## 1. Introduction

Image captioning is an important task in the intersection of computer vision and natural language processing that has many valuable applications. The task is mainly given an image, letting a model produce one or several captions for the image. While this task is relatively easy and obvious for humans, it is complicated for machines to understand an image and produce a relevant caption corresponding to the image.

The image captioning task has attracted much research attention. There has been strong progress made on this task. However, previous works all solve this task in a fully-supervised setting, and no prior attempt has been made to generate image captions in a semi-supervised way.

Semi-supervised learning is the idea of making use of unlabeled data for training, typically a small amount of labeled data with a large amount of unlabeled data. In terms of the image captioning task, it is easy to obtain a large image set and a big language set, but it is difficult to build a large image-language dataset where each image paired with several captions. Therefore, it is important to build an image captioning model which can be trained in a semi-supervised

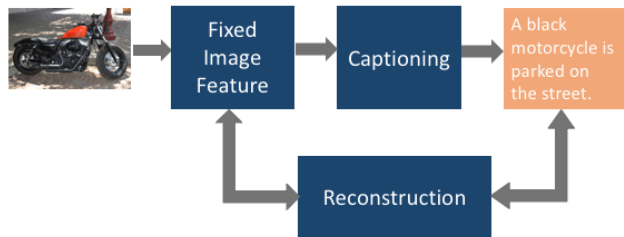


Figure 1: The high-level structure of our model. 1) In the captioning phase, an image feature is first extracted from the given image via a CNN and then fixed. A corresponding caption is generated using a RNN based on the fixed image feature. In the reconstruction phase, the generated caption is first encoded by a RNN and then transformed through a fully-connected layer to reconstruct back the image feature. The training objective is to minimize the mean square error (MSE) between the reconstructed image feature and the fixed original image feature. 2) If a training image comes with a ground-truth caption, a cross-entropy loss is applied between the generated caption and the ground-truth caption.

way.

We introduce a novel semi-supervised approach to generate image captions. The overall high-level structure is shown in Fig. 1. Our model has two phases, a captioning phase and a reconstruction phase. In the captioning phase, given an image, we first use a convolutional neural network (CNN) to extract its feature. Then we fix the image feature, and use a recurrent neural network (RNN) to generate its corresponding caption. In the reconstruction phase, the generated caption is first encoded via a RNN and then transformed using a fully-connected layer to reconstruct back the image feature. Because word samples in captions are discrete, it will be problematic to directly backpropagate gradients through them during training. We apply the Gumbel-Softmax approximation [5] [9] between the captioning phase and the reconstruction phase to tackle this issue. Gumbel-Softmax approximation is a continuous relaxation combined with the re-parametrization of the sampling process which allows backpropagation through sam-

ples from a categorical distribution.

A reconstruction loss is calculated between the reconstructed image feature and the fixed original image feature. This penalizes for the caption not capturing the correct image feature. If an training image is paired with a ground-truth caption, additional supervision is integrated into our model by adding a cross entropy loss penalizing incorrect words in the generated caption directly. At test time, we evaluate how the captions generated by our model are close to the ground-truth captions.

Our model can generate captions associated with images with a small amount of training images paired with ground-truth captions. The reconstruction phase is the key in our solution. We evaluate our model on the MSCOCO dataset [8]. We show that when the image annotation is limited, our model achieves better results than the standard fully-supervised image captioning models.

## 2. Related Work

**Image Captioning.** Image captioning is currently an active research area. [16] presents a generative model maximizing the likelihood of the target description sentence given the training image with a vision CNN and a language generating RNN. [6] proposes an alignment model that maps words and image regions into a common multimodal embedding space. They then use a multimodal RNN to generate image descriptions by using the inferred alignments. [14] tackles the problem of generating a set of image captions that is indistinguishable from human written captions. [2] focuses on producing semantically relevant, natural and diverse sentences for images. They constructed a conditional-GAN based model which jointly learns a generator for image captions and an evaluator to evaluate the quality of the generated descriptions. However, all of these previous works fall into the fully-supervised learning region, while our model can generate image captions in a semi-supervised setting.

**Reconstruction.** The idea of reconstruction makes training with partially annotated data possible. [12] proposes a model to ground texts to images with all levels of bounding box supervision. In their semi-supervised and unsupervised frameworks, reconstructing a given phrase enables the model to work with unlabeled data. In this work, we share the same idea with this paper of using reconstruction. Under the semi-supervised setting, we add a reconstruction component to our model. The reconstruction component enables us to train our model using images without ground-truth captions.

**Gumbel-Softmax Approximation.** The idea of Gumbel-Softmax approximation is first proposed in [5] and [9]. Specifically, [5] presents a efficient gradient estimator that approximates the non-differentiable categorical distribution using a differentiable Gumbel-Softmax distribution.

[9] introduces continuous relaxations of discrete random variables to re-factor discrete stochastic nodes of a computation graph into one-hot bit representations. [14] uses the Gumbel-Softmax approximation to backpropagate the gradients from the discriminator to the generator in their GAN-based image captioning model. Similarly, we incorporate the Gumbel-Softmax approximation into our model to tackle the discreteness problem between the captioning phase and the reconstruction phase.

Initially, we also considered applying a caption discriminator with Maximum Mean Discrepancy loss as in [17] on the generated caption. Ideally, this will help regularize the generated caption to be realistic-looking. However, due to the time limit, we are not able to implement the caption discriminator for now, which will be our future work.

## 3. Model

Our model is shown in Fig. 2, which generally has two phases, a captioning phase and a reconstruction phase.

Given an image, we first use the ResNet-152 [4] model to extract the image feature. The image feature is fixed. The captioning phase contains two parts, one projection layer and one caption decoder. We first use the projection layer to change the dimension of the image feature and then feed it to the caption decoder. The caption decoder will generate a caption based on the projected image feature. The reconstruction phase also contains two parts, one caption encoder and one transforming layer. The caption generated by the caption decoder will be encoded as a feature vector by the caption encoder. Then we use the transforming layer to transform the caption feature to something like an image feature, which we call reconstructed image feature. A mean square error (MSE) loss is applied between the reconstructed image feature and the fixed extracted image feature to train the network. We use the Gumbel-Softmax approximation [5] [9] between the caption decoder and the caption encoder to allow backpropagate gradients among the discrete word samples during training.

If the given image is annotated with a ground-truth caption, we add another cross entropy loss between the generated caption (after Gumbel-Softmax approximation) and the ground-truth caption. We only use the captioning phase at test time. The model detail is described below.

### 3.1. Data Preprocessing

**Image feature representation.** We use the image feature extracted right before the last softmax layer from the ResNet-152 [4] model pre-trained on ImageNet [13]. The dimension for the image feature vector is 2048. We fix the image feature for training and testing our model.

**Word vocabulary and preprocessing.** We put all the words appeared in the training sentence set more than 4

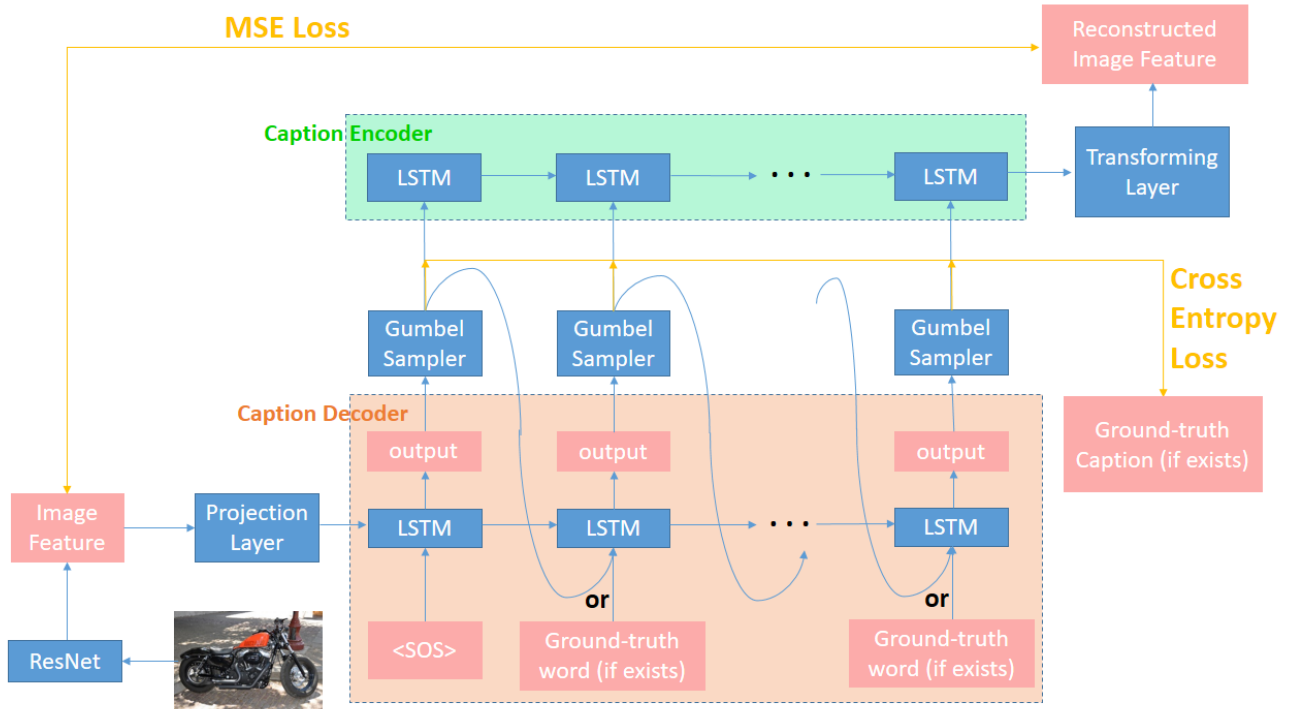


Figure 2: Our model has two phases, a captioning phase and a reconstruction phase. The captioning phase includes a projection layer and a caption decoder. The reconstruction phase consists of a caption encoder and a transforming layer.

times into our word vocabulary. We also add a special token  $\langle \text{SOS} \rangle$  to indicate the start of the sentence and a special token  $\langle \text{EOS} \rangle$  to indicate the end of the sentence. We prepend the  $\langle \text{SOS} \rangle$  token and append the  $\langle \text{EOS} \rangle$  token to each training sentence. After the vocabulary is built, we use the pre-trained Word2Vec [10] embedding model to represent the words. For the word representations, the dimension we choose is 300.

### 3.2. Captioning Phase

**Projection layer.** With the extracted fixed image feature, we use a projection layer to change its dimension. The projection layer is a single fully-connected layer with ReLU activation. It transforms the dimension of the image feature from 2048 to 512, which is the same as the hidden state dimension of the caption decoder and the caption encoder.

**Caption decoder.** We use a single RNN with LSTM units as the caption decoder to generate the image caption based on the projected image feature. The hidden state dimension is 512. We use the projected image feature vector as the initial hidden state of the caption decoder. We apply the Gumble-Softmax approximation on the output of the LSTM unit. We treat the output at each time step as a word in the generated caption. The input at the first time step is the  $\langle \text{SOS} \rangle$  token. If the image does not have a

annotated caption, the input to the following time step is the output (after Gumble-Softmax approximation) from the previous time step. Otherwise, the input to the following time step is the following word in the ground-truth caption. The inputs are all applied with the Word2Vec [10] embedding. The caption is finished generating until the  $\langle \text{EOS} \rangle$  token is generated.

### 3.3. Reconstruction Phase

**Caption encoder.** For the generated caption (after Gumble-Softmax approximation), we use another RNN with LSTM units to encode it. We initialize the hidden state of the caption encoder using a zero vector. The hidden state dimension is also 512. At each time step, the input to the caption encoder is the corresponding output from the caption decoder applied with the Word2Vec [10] embedding. We use the hidden state vector at the last time step of the caption encoder to represent the generated caption.

**Transforming layer.** After the caption feature vector is obtained, we employ a transforming layer to translate the caption feature to something like an image feature. We implement the transforming layer using a fully-connected layer with ReLU activation. The output dimension of the transforming layer is 2048, which is the same as the fixed extracted image feature dimension.

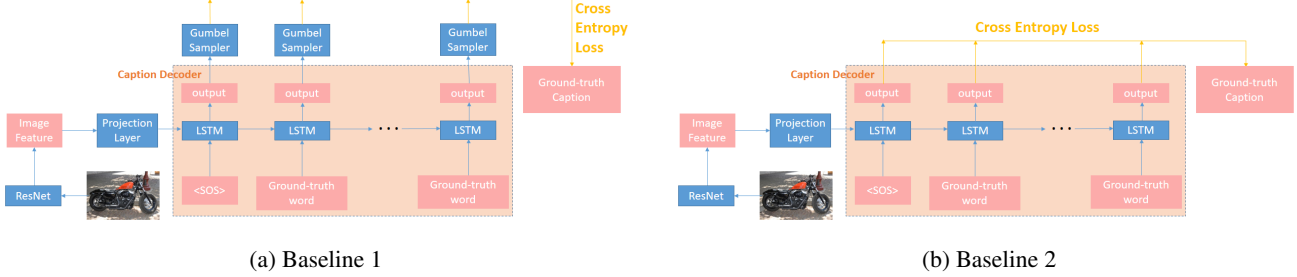


Figure 3: Baseline Models. (a) Baseline 1 is our model without the caption encoder and the MSE loss (the reconstruction phase). (b) Baseline 2 is Baseline 1 without the Gumbel-Softmax approximation.

### 3.4. Discreteness Problem

When training our model, we need to backpropagate gradients from the caption encoder to the caption decoder. However, since words are discrete samples, this makes the standard backpropagation a problem. We adopt the Gumbel-Softmax approximation [5] [9] to tackle this issue. Gumbel-Softmax approximation is a continuous relaxation of discrete random variables, combined with a reparametrization trick. It consists of two steps.

First, the Gumbel-Max trick [3] is applied to reparametrize sampling from a categorical distribution. Given a random variable  $r$  drawn from a categorical distribution parametrized by  $[\theta_1, \theta_2, \dots, \theta_n]$  where  $n$  is the number of categories,  $r$  can be expressed as

$$r = \text{one\_hot} \left[ \underset{i}{\operatorname{argmax}} (g_i + \log \theta_i) \right], \quad (1)$$

where  $g_i$ 's are i.i.d. standard gumbel distributed random variables, which can be simply computed as

$$g = -\log(-\log(u)) \quad (2)$$

where  $u$  is drawn from a uniform distribution in the range  $(0, 1)$ .

Next, approximate the argmax operation in Equation (1) with softmax, which results in a continuous and differentiable variable

$$r' = \text{softmax} \left[ \frac{g_i + \log \theta_i}{\tau} \right] \quad (3)$$

where  $\tau$  is a temperature parameter which controls how close  $r'$  is to  $r$ .  $r' = r$  when  $\tau = 0$ .

As in [14], we use the straight-through variation of the Gumbel-Softmax approximation [5] at the output of the caption decoder. That is, we use  $r$  in the forward pass and  $r'$  is the backward pass to allow backpropagation during training. In all our experiments, we set  $\tau = 1$ .

### 3.5. Loss Functions

**MSE loss.** A mean square error (MSE) loss is applied between the generated image feature and the fixed extracted image feature. For a  $d$ -dimension input vector  $x$  and a  $d$ -dimension target vector  $y$ , the MSE loss is calculated as

$$\text{MSE\_loss} = \frac{1}{d} \|x - y\|_2^2. \quad (4)$$

**Cross entropy loss.** During training, if an image comes with a ground-truth caption, a cross entropy loss is also applied between the generated caption (after Gumbel-Softmax approximation) and the ground-truth one. The cross entropy loss is a combination of the log-softmax operation and a negative log likelihood loss.

### 3.6. Model Weights Initialization

We train a purely language auto-encoder using the training sentence set. The structure of the encoder is the same as the caption encoder, and its hidden state is initialized as a zero vector. The structure of the decoder is the same as the caption decoder, but its hidden state is initialized as the last hidden state of the encoder. We train the auto-encoder on the training sentence set for 4 epochs. Then use the weights of the decoder/encoder of the auto-encoder as the initial weights of the caption decoder/caption encoder. For the projection layer and transforming layer (fully-connected layers), the weights are initialized as random numbers sampled from a standard normal distribution and biases are initialized as zeros.

## 4. Experiments

### 4.1. Baselines

We build two baselines. Baseline 1 is our model without the caption encoder and the MSE loss, that is, without the reconstruction phase. This model is illustrated in Fig. 3a. The cross entropy loss is still applied between the generated caption (after Gumbel-Softmax approximation) and the ground-truth ones. This aims to show the effectiveness of the reconstruction phase in our model. To

Anno. Rate	Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE.L	CIDEr
100%	Our Model	0.664	0.481	0.335	0.232	0.219	0.484	0.747
	Baseline 1	0.666	0.483	0.336	0.232	0.220	0.486	0.746
	Baseline 2	0.669	0.483	0.336	0.233	0.221	0.486	0.750
50%	Our Model	<b>0.655</b>	<b>0.466</b>	<b>0.319</b>	0.216	<b>0.215</b>	<b>0.477</b>	<b>0.704</b>
	Baseline 1	0.651	0.464	0.319	<b>0.218</b>	0.212	0.475	0.689
	Baseline 2	0.643	0.453	0.307	0.208	0.210	0.468	0.666
25%	Our Model	<b>0.638</b>	<b>0.449</b>	<b>0.301</b>	<b>0.200</b>	<b>0.208</b>	<b>0.468</b>	<b>0.651</b>
	Baseline 1	0.627	0.437	0.290	0.193	0.200	0.459	0.610
	Baseline 2	0.627	0.431	0.284	0.189	0.197	0.455	0.588
12.5%	Our Model	<b>0.617</b>	<b>0.431</b>	<b>0.288</b>	<b>0.192</b>	<b>0.199</b>	<b>0.453</b>	<b>0.596</b>
	Baseline 1	0.495	0.287	0.159	0.094	0.140	0.367	0.211
	Baseline 2	0.531	0.337	0.205	0.127	0.160	0.394	0.336

Table 1: Quantitative results for the pre-trained models.

Anno. Rate	Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE.L	CIDEr
100%	Our Model	0.676	0.493	0.346	0.241	0.224	0.493	0.773
	Baseline 1	0.677	0.497	0.349	0.243	0.226	0.495	0.779
	Baseline 2	0.673	0.492	0.345	0.240	0.222	0.491	0.766
50%	Our Model	0.650	0.464	0.317	0.216	0.212	0.475	0.691
	Baseline 1	<b>0.665</b>	<b>0.481</b>	<b>0.333</b>	<b>0.228</b>	<b>0.217</b>	<b>0.484</b>	<b>0.726</b>
	Baseline 2	0.650	0.464	0.317	0.214	0.214	0.475	0.692
25%	Our Model	0.577	0.366	0.221	0.137	0.167	0.413	0.392
	Baseline 1	0.617	0.425	0.277	0.182	0.192	0.448	0.557
	Baseline 2	<b>0.623</b>	<b>0.434</b>	<b>0.290</b>	<b>0.194</b>	<b>0.202</b>	<b>0.459</b>	<b>0.610</b>
12.5%	Our Model	0.454	0.261	<b>0.138</b>	<b>0.062</b>	<b>0.131</b>	0.347	<b>0.095</b>
	Baseline 1	<b>0.519</b>	<b>0.273</b>	0.122	0.057	0.121	<b>0.349</b>	0.069
	Baseline 2	0.407	0.168	0.064	0.032	0.108	0.304	0.047

Table 2: Quantitative results for the models without pre-training.

show that whether the Gumbel-Softmax approximation will largely change the model performance, we build the second baseline, Baseline 2, which is shown in Fig. 3b. Baseline 2 is Baseline 1 without the Gumbel-Softmax approximation. In this model, the cross entropy loss is applied between the output directly from the caption decoder and the ground-truth captions.

## 4.2. Dataset and Evaluation Metrics

We use the MSCOCO dataset [8] with the 2014 train/validation split to train and evaluate the models. This dataset version has 82783 training images and 40504 validation images. Each training/validation image is annotated with at least 5 ground-truth captions. We randomly split half of the validation data as test set and the remaining as the validation set used during training. We use the whole training data in the training stage. The built vocabulary has size 8855.

We use the commonly used language evaluation metrics to evaluate the models. The language evaluation metrics

include BLEU-1,2,3,4 [11], METEOR [1], ROUGE-L [7], and CIDEr [15]. We use the code available online to calculate these scores <sup>1</sup>.

## 4.3. Results

We conduct experiments with image annotation rates as 100%, 50%, 25%, and 12.5%. The image annotation rate is how many percent of images in the training set annotated with ground-truth captions. We use these percentages of images with their ground-truth captions to train the baselines, since the baselines are standard fully-supervised image captioning models. For our model, besides using these annotated images, we also include the remaining images in the training set without annotations during training.

As mentioned in section 3.6, the weights of the caption decoder and the caption encoder in our model and baselines are initialized as the weights of a pre-trained auto-encoder. We refer these models as pre-trained models or models with pre-training. To validate the usefulness of using the pre-

<sup>1</sup><https://github.com/tylin/coco-caption>



	Anno. Rate	Model	Generated Caption
	100%	Our Model Baseline 1 Baseline 2	a man playing tennis on a tennis court. a man playing tennis on a tennis court. a man is playing tennis on a court.
	50%	Our Model Baseline 1 Baseline 2	a man is playing tennis on a tennis court. a woman is holding a tennis racket and ball. a woman in a short skirt holding a tennis racket.
	25%	Our Model Baseline 1 Baseline 2	a man is playing tennis on a court a man in a blue shirt is playing tennis a man in a red shirt is playing tennis
	12.5%	Our Model Baseline 1 Baseline 2	a tennis player is swinging his racket on the court. a man in a suit is standing in the grass. a man in a baseball uniform swinging his bat at a pitch.
	100%	Our Model Baseline 1 Baseline 2	a man holding a surfboard walking in the water. a man holding a surfboard on a beach. a man holding a surfboard on the beach.
	50%	Our Model Baseline 1 Baseline 2	a man holding a surfboard on top of a beach. a man holding a surfboard on top of a beach. a man holding a surfboard on a beach.
	25%	Our Model Baseline 1 Baseline 2	a man is holding a surfboard in the water. a man in a suit and a suit holding a kite. a man and a woman are on a beach.
	12.5%	Our Model Baseline 1 Baseline 2	a man in a wet suit is holding a surf board a man in a suit is standing in the snow. a man in a wet suit is surfing on a wave.

Table 3: Sample qualitative results for the pre-trained models.

trained weights, we also conduct the same experiments to all the three models without using the pre-trained weights. We refer these models without using the pre-trained weights as models without pre-training.

**Implementation details.** We use PyTorch<sup>2</sup> framework to build all the models. We use Adam optimizer to train all the network components. For the pre-trained models, the initial learning rates for the linear layers are 0.001, and are set to 0.0008 for the RNNs. For the models without pre-training, we set the initial learning rates for all the network components as 0.001.

#### 4.3.1 Quantitative Results

The quantitative results in terms of BLEU-1,2,3,4 [11], METEOR [1], ROUGE-L [7], and CIDEr [15] scores for the pre-trained models are shown in Table 1. The results for the models without pre-training are shown in Table 2.

According to Table 1, when the image annotation rate is 100%, our model is comparable to the standard image captioning models, Baseline 1 and Baseline 2. When the image annotation rate is 50% or 25%, the advantage of our model appears. Our model outperforms the two baselines in terms

<sup>2</sup><http://pytorch.org/>

of all the seven scores except the BLEU-4 score when the image annotation rate is 50%, and when the image annotation rate is 25%, our model outperforms the two baselines for all the seven scores. When the image annotation rate decreases to 12.5%, the advantage of our model becomes significant. In this setting, both baselines can not produce reasonable captions for images, while our model can generate relatively good captions. This is also illustrated by the qualitative results shown below.

Comparing the results in Table 1 and Table 2, it is necessary to use the pre-trained weights to initialize the caption decoder and the caption encoder of our model to achieve good caption results when the image annotation rate is low (25% and 12.5%). For our model, there is too many degrees of freedom from the extracted image feature to the reconstructed image feature. It is essential to add some regularizer or language priors onto our model.

Comparing the performance of Baseline 1 and Baseline 2, we can see that using Gumbel-Softmax approximation will not much change the results.

#### 4.3.2 Qualitative Results

We show some sample qualitative results for the pre-trained models in Table 3 and for the models without pre-training in



	Anno. Rate	Model	Generated Caption
	100%	Our Model Baseline 1 Baseline 2	a woman is playing tennis on a court a man is playing tennis on a tennis court. a woman in a white shirt and black shorts playing a game of tennis.
	50%	Our Model Baseline 1 Baseline 2	a woman in a blue shirt is playing tennis a man is playing tennis on a tennis court. a man is playing tennis on the court
	25%	Our Model Baseline 1 Baseline 2	a man in a red shirt is playing tennis a man in a red shirt is playing tennis a man holding a tennis racket on a tennis court.
	12.5%	Our Model Baseline 1 Baseline 2	a man is sitting on a bench next to a woman. a man is sitting on a bench in the park. a man in a suit and tie is standing by a fence.
	100%	Our Model Baseline 1 Baseline 2	a man holding a surfboard walking in the water. a man holding a surfboard on the beach. a man holding a surf board on a beach.
	50%	Our Model Baseline 1 Baseline 2	a man in a wet suit is on a surfboard. a man holding a surfboard on a beach. a man standing on a beach holding a surf board.
	25%	Our Model Baseline 1 Baseline 2	a man is surfing on a wave in the ocean. a man in a wet suit is surfing on a wave a man holding a surfboard on top of a beach.
	12.5%	Our Model Baseline 1 Baseline 2	a man is sitting on a bench next to a woman. a man is sitting on a bench in the park. a man in a suit and tie is standing by a fence.

Table 4: Sample qualitative results for the models without pre-training.

Table 4. From these qualitative results, we can see that our model can, while the baselines can not, produce relatively good captions for images when the image annotation rate is low. It is necessary to use the pre-trained weights for our model to achieve these results.

## 5. Conclusion and Discussion

In this work, we propose an image captioning model using the idea of reconstruction that can generate image captions in a semi-supervised learning setting. We show that our model can generate relatively good image captions when only a small amount of training images are annotated with ground-truth captions.

Future work can be in the direction that adapting the current model to generate image captions unsupervisedly. Actually, we have also trained the current model with 6.25%, 3.125%, and even 0% of training images being annotated. However, the captioning results of our model are not good under these settings. To modify the current model to fit even lower image annotation rates or in an unsupervised learning setting, one way is to incorporate the idea of generative adversarial networks. We think that it will be beneficial to add a caption discriminator to discriminate the quality of the caption generated by the caption decoder in these set-

tings.

## Acknowledgements.

We want to thank the course instructor, Professor Leonid Sigal, and other peers for providing the countless valuable comments and feedback to this work.

## References

- [1] S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005. 5, 6
- [2] B. Dai, D. Lin, R. Urtasun, and S. Fidler. Towards diverse and natural image descriptions via a conditional gan. *arXiv preprint arXiv:1703.06029*, 2017. 2
- [3] E. Gumbel. Statistical theory of extreme values and some practical applications: A series of lectures. *US Government Printing Office, Washington*, 1954. 4
- [4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2
- [5] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 1, 2, 4

- [6] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015. 2
- [7] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004. 5, 6
- [8] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 2, 5
- [9] C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016. 1, 2, 4
- [10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013. 3
- [11] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002. 5, 6
- [12] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele. Grounding of textual phrases in images by reconstruction. In *European Conference on Computer Vision*, pages 817–834. Springer, 2016. 2
- [13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge, 2014. 2
- [14] R. Shetty, M. Rohrbach, L. A. Hendricks, M. Fritz, and B. Schiele. Speaking the same language: Matching machine to human captions by adversarial training. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 4
- [15] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015. 5, 6
- [16] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 3156–3164. IEEE, 2015. 2
- [17] Y. Zhang, Z. Gan, K. Fan, Z. Chen, R. Heno, D. Shen, and L. Carin. Adversarial feature matching for text generation. *arXiv preprint arXiv:1706.03850*, 2017. 2