# A Character-Based Model for Pragmatic Caption Generation

Poorvi Bhargava, Reuben Cohn-Gordon, Hiroto Udagawa
Stanford University
poorvib@stanford.edu,reubencg@stanford.edu,hiroto@stanford.edu

## Abstract

*Following the work of ([14]) and ([11]), we implement a pragmatic image captioning system, which generates informative captions in the context of distractor regions or images. We extend previous work by making use of a character LSTM in our neural model, which allows for improved pragmatic captioning. We use Visual Genome for data, which allows us to obtain captions for different regions in a single image, for regions which are similar. Qualitatively, we found that, in addition to describing a target image with less ambiguity in context with another image, the pragmatic captions often offered a more accurate, longer description of the target image than just the literal caption.*

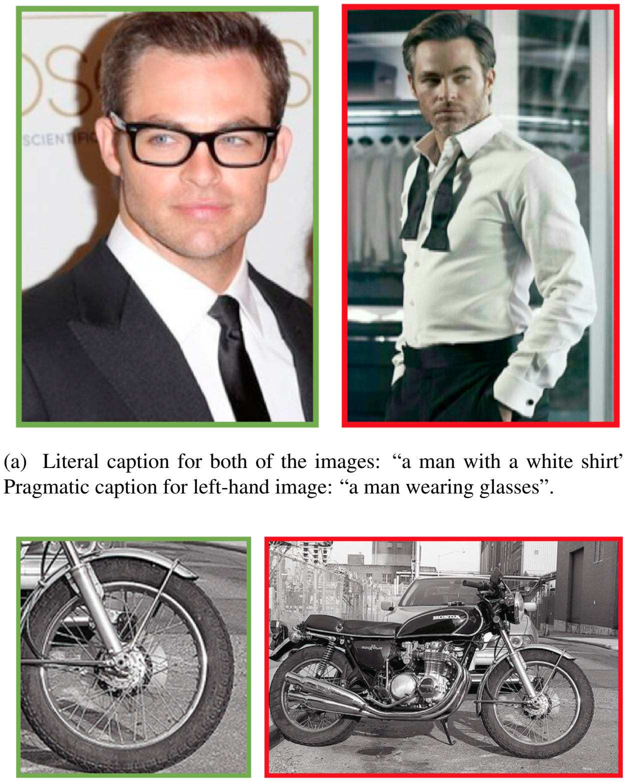
## 1. Introduction

Automatic image captioning is a complex task that has greatly benefited from end-to-end neural models. A neural image captioning model can be construed as a distribution of possible utterances conditioned on images. The task is of interest for AI, since it requires the model to learn a semantics for both images and language jointly, by mapping the input image to a hidden vectorial representation and then to a linguistic expression.

While semantics is key to artificial intelligence, human language use involves another component: pragmatics ([10]). This is the ability of language users to reason about other social agents.

Recent computational research in cognitive science ([1],[2]) has shown the success of Bayesian models of pragmatics, where a speaker, modeled as a distribution over utterances given world states, takes into account a listener, modeled as a distribution over world states given utterances. This paradigm is termed the *Rational Speech Acts* model.

A typical example of a task which requires pragmatic reasoning is *referring expression generation* ([8]), in which a speaker is presented with a pair (or more generally a set) of objects, and must generate a linguistic expression which singles out only one of the objects.



(a) Literal caption for both of the images: "a man with a white shirt". Pragmatic caption for left-hand image: "a man wearing glasses".

(b) Literal caption for left-hand image: "the wheel of a bike". Literal caption for right-hand image: "a motorcycle". Pragmatic caption for target: "front wheel of a motorcycle".

Figure 1: To distinguish between a target image (left, green) and a distractor image (right, red), our model generated the above example captions, which describe the target image well without referring to the distractor image as well. In the first case, the distractor is a separate image from the target, while in the second, the target is a region of the distractor.

A natural extension of RSA is to use a neural speaker model, such as an image captioning model, on top of which pragmatic calculations are made. A natural reframing of reference expression generation in the paradigm of image captioning is *pragmatic caption generation*, where a model

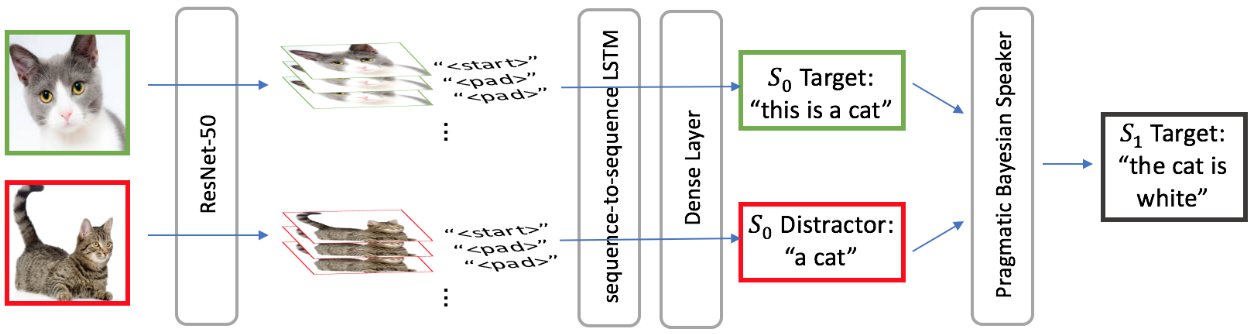

ResNet-50
"<start>"
"<pad>"
"<pad>"
⋮
"<start>"
"<pad>"
"<pad>"
⋮
sequence-to-sequence LSTM
Dense Layer
$S_0$ Target: "this is a cat"
$S_0$ Distractor: "a cat"
Pragmatic Bayesian Speaker
$S_1$ Target: "the cat is white"

Figure 2: Model Architecture (end-to-end).

must produce a caption which describes one but not the other of a pair of images (or a pair of regions in a single image). We will refer to the image which the caption is intended to refer to as the target and the other image as the distractor.

([14]) and ([11]) both put forward pragmatic image captioning models. Our goal is to build on their approaches. The inputs to our algorithm are two images, which are first passed through a pretrained ResNet-50 CNN model to obtain two feature vectors. Next, the feature vectors are passed through sequence-to-sequence LSTMs at every time step, along with the input word or character. The Bayesian layer is applied after the LSTM to obtain a pragmatic caption, which refers to the target image but not the distractor.

We explore the use of character based LSTMs for this task, which offer more flexible language modeling and often improve performance (see [7]).

## 2. Related Work

Neural image captioning has proved very successful, with work such as ([5]) employing a CNN-RNN architecture to produce captions for images. Recent approaches have employed more complex architectures (e.g. [15], which makes use of an attention mechanism over the input image.).

In order to make our captioning system pragmatic, as described above, we need to model language use in context. For present purposes, the context we are interested in is a set of candidate images (we consider the simple case of a target image and a single distractor). The task of a pragmatic speaker is to generate captions which are informative about the target in the presence of the distractor. Qualitatively, this amounts to choosing aspects of the target which do not also pertain to the distractor.

Work in formalization pragmatics in language more generally began informally with ([3]) and other philosophers of language. The core intuition is that a speaker designs utterances with a model of a listener in mind, and aims to produce utterances which are informative about their own world knowledge. Recently, pragmatics has been formalized in the Rational Speech Acts models ([1]), in a Bayesian setting. The aim is to model speakers and listeners as distributions P(utterance|world) and P(world|utterance) respectively, in a setting where both reason about the behavior of the other.

Fusing Bayesian pragmatics and NLP offers the opportunity of obtaining the realistic human behavior of the former and the scalability to real world data of the latter. One instance of this is pragmatic image captioning, a task recently attempted by ([11]) and ([14]).

The aim of these papers, which we describe in detail in the following section, is to produce a system which generates captions which take pragmatics into account.

## 3. Methods

A non-pragmatic, literal image model can be framed as a probability distribution $S_0 = P(C|I)$, where $C$ refers to a caption and $I$ refers to a given image. To generate vanilla image captions, images were passed through a ResNet-50 CNN and feature vectors for each were obtained from the last fully connected layer. Next, each feature vector was concatenated to a start token character and was passed through a sequence-to-sequence LSTM. Both word and character-based models were employed. The output at each time step was then concatenated again with the corresponding feature vector and passed into the next time step. This generated the literal caption.

A *pragmatic* captioning model, on the other hand, attempts to generate a caption which describes a target image well while not describing a distractor image (or set of images[1]).

[1] Though our model is theoretically capable of modeling image sets of

Formally, we can first define a listener model $L$ using Bayes rule, modified from ([14]). For target image $I_t$ and distractor image $I_d$:

$$L = P_{posterior}(I_t|C) = \frac{P(C|I_t) * P_{prior}(I_t)}{\sum_{j\in\{t,d\}} P(C|I_j) * P_{prior}(I_j)}$$

$S_1$ then maximizes the likelihood of $I_t$ under L, assuming uniform prior probabilities. Intuitively, this forces the model to generate a caption which is good for the target but not for the distractor.

To implement $S_1$, two general approaches are possible, which we shall refer to as the modular and end-to-end $S_1$ respectively. A modular system involves the training of $S_0$ on normal pairs of images and captions, and calculating L in terms of $S_0$. $S_1$ is then defined as the weighted sum of $S_0$ and L.

By contrast, for the end-to-end $S_1$, the training items consist of pairs of images, and a ground truth caption which has been produced in a context. Our neural model then back-propagates not only through an RNN and CNN, but also the Bayesian layer. In terms of model architecture, this can be understood as a Siamese network, where the target and distractor image are both fed through a standard image captioning model $S_0$ and then merged at the end with the function:

$$\lambda(x, y) \rightarrow \frac{x}{x+y}$$

We consider variations on this end-to-end model, based around different modes of combining the $S_0$ prediction for each image. For instance, we experimented with having a further LSTM fold over the timestep-wise concatenated outputs of the $S_0$. We also tried initializing the weights for the $S_0$ to those pretrained on the non-end-to-end task.

Thus, our model can be summarized as being composed of the following three components, derived from ([14]):

1. $S_0 = f_{S_0}(I) = P(C|I)$
   which generates literal speaker semantic captions, using a pretrained ResNet-50 CNN for forward pass, feature vectors, and a sequence-to-sequence LSTM to literal generate captions from feature vectors.

2. $L = f_L(C, I_t, I_d) = P_{posterior}(I_t|C)$
   a Bayesian method listener which tries to differentiate between the target image, $I_t$, and the distractor image, $I_d$.

3. $S_1 = f_{S_1}(I_t, I_d) = argmax_C\{\lambda S_0 + (1-\lambda)L\}$
   a Bayesian method to act as a "pragmatic speaker" and generate pragmatic captions in the context of a pair of images (or a pair of regions in a single image).

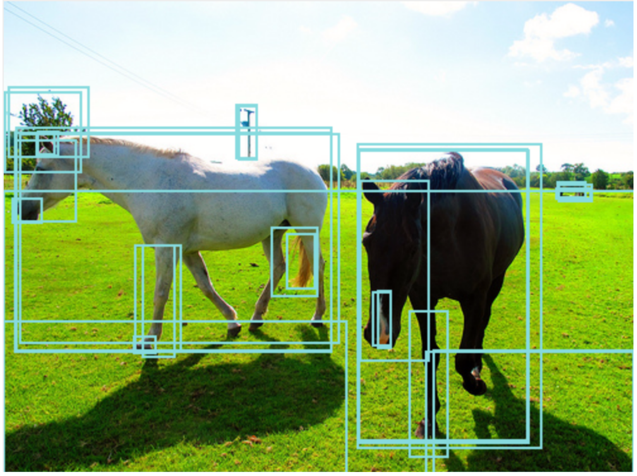


Figure 3: The figure above is an example of an image from the Visual Genome dataset. Each of the blue regions were selected and captioned by workers on Mechanical Turk, under the assumption that they are described more specifically in context to the entire image. Examples of pragmatic region captions include "The white horse" or "The horse on the left", when trying to describe the horse on the left but not the horse on the right. ([9])

In order to generate captions from our model at predict time, we use a beam search rather than greedy unrolling. In the character model, the beam width was 39, one for each character in the training set. In the word model, the beam width was limited to the 10,000 most common words in the training dataset. Thus, the character-based model offers a computational advantage specifically for beam search.

For our implementations of both end-to-end and modular $S_1$, we used both a character and word based language model for our caption generation. Since the Visual Genome captions are quite short, and contain a reasonably high frequency of irregular spelling, we hope this will improve captioning quality, as well offering a new type of captioning model in general.

A key difference between our models and those tried in previous work is the use of a character based language model. In other words, captions are generated character by character, rather than word by word. To our knowledge, this is the first such instance of this approach for captioning.

## 4. Dataset and Features

The chosen dataset was Visual Genome ([9]), which contains 108,077 images, with over 4 million region descriptions (averaging about 40-50 region descriptions per image). Region descriptions were generated by language users who were presented with the entire image and sectioned off areas to caption. The average length of each region de-

arbitrary size, we restrict ourselves to two images.

scription was 5.18 words. There was an average of 1.01 objects per region and about 21.24 objects per image. The most common objects included people and buildings and the most common attributes included colors and size. An example image can be seen in Figure 3

From each image, one region was chosen such that it was at least 100 pixels and had a nearly 1:1 pixel height to width ratio. The average size of each image was 500.14 pixels. The region was cropped and scaled to 224 by 224 pixels and the associated caption formed the ground truth. This formed the training set for $S_0$ and was composed of roughly 55,000 images, 5000 validation images, and 1000 test images.

For testing the $S_1$ model, we selected pairs of visually similar regions, on the assumption that the labels for these regions will have been created by users who are actively trying to disambiguate it from the similar regions in the same image. For each image, two non-overlapping regions were selected and fed into the model.

We also used these pairs of images and distractors as input to the end-to-end word and character models. One of the flaws of our design was insufficient constraints on the quality of the distractors. Because the distractors were often unrelated to the target images, the process of learning pragmatics and semantics was very noisy.

For this reason, we considered an alternative choice of distractor dataset for both training and testing. Here, the distractors consisted of the full uncropped image from which the region was selected. This sort of distractor contains the target region, and as such, presents a slightly different sort of task to pragmatic captioning with a non-overlapping distractor region.

## 5. Experiments, Results, and Discussion

Though our focus was to produce a character-based LSTM for captioning, we also implemented an $\mathrm{S}_0$ with a word-level LSTM, using Glove vectors for our pretrained word embeddings ([13]). We began with a word based approach as this seemed easier to train and prior literature had greater success with this type of model. However, we hypothesized that a character based approach would be more versatile in pragmatic predictions, be better at coping with learned typos from the training data, absolve the need for a pre-trained word embedding, and allow for smaller beam width to be used.

For both word and character models, we trained a modular and an end-to-end $\mathrm{S}_1$. Our most successful results, qualitatively, by far came from the modular system - the end-to-end $\mathrm{S}_1$ learned a poor latent $\mathrm{S}_0$, despite a number of variations of the architecture.

For our character model, we used a 256 dimensional LSTM, with 30 timesteps. We trained only a final dense layer of output dimension 64 on top of the pretrained ResNet, which we concatenated with our character at each timestep. A final dense layer of output dimension 39 (the number of characters we used), with a softmax activation, produced the output distribution. Our loss was categorical cross entropy. For the modular $\mathrm{S}_1$, we found that a weighting of 0.2 on L and 0.8 on $\mathrm{S}_0$ produced the best results. One disadvantage of the modular $\mathrm{S}_1$ is that the ideal weighting seems to vary from example to example, depending on the relevance of the distractor - ideally, the weighting parameter should therefore be incorporated into the model.

We use no dropout or other regularization, and trained on 55000 image-caption pairs. We used the RMSProp optimizer with a learning rate of 0.001, and aborted training once the loss on the validation set began to increase. The model was implemented in Keras, an library built on top of Tensorflow. In addition, beam search was incorporated and produced qualitatively better captions than the model without beam search.

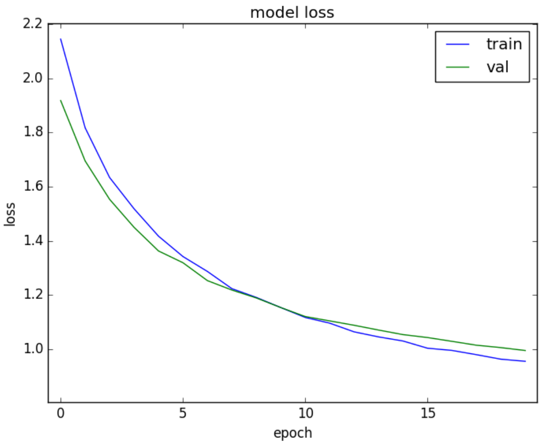
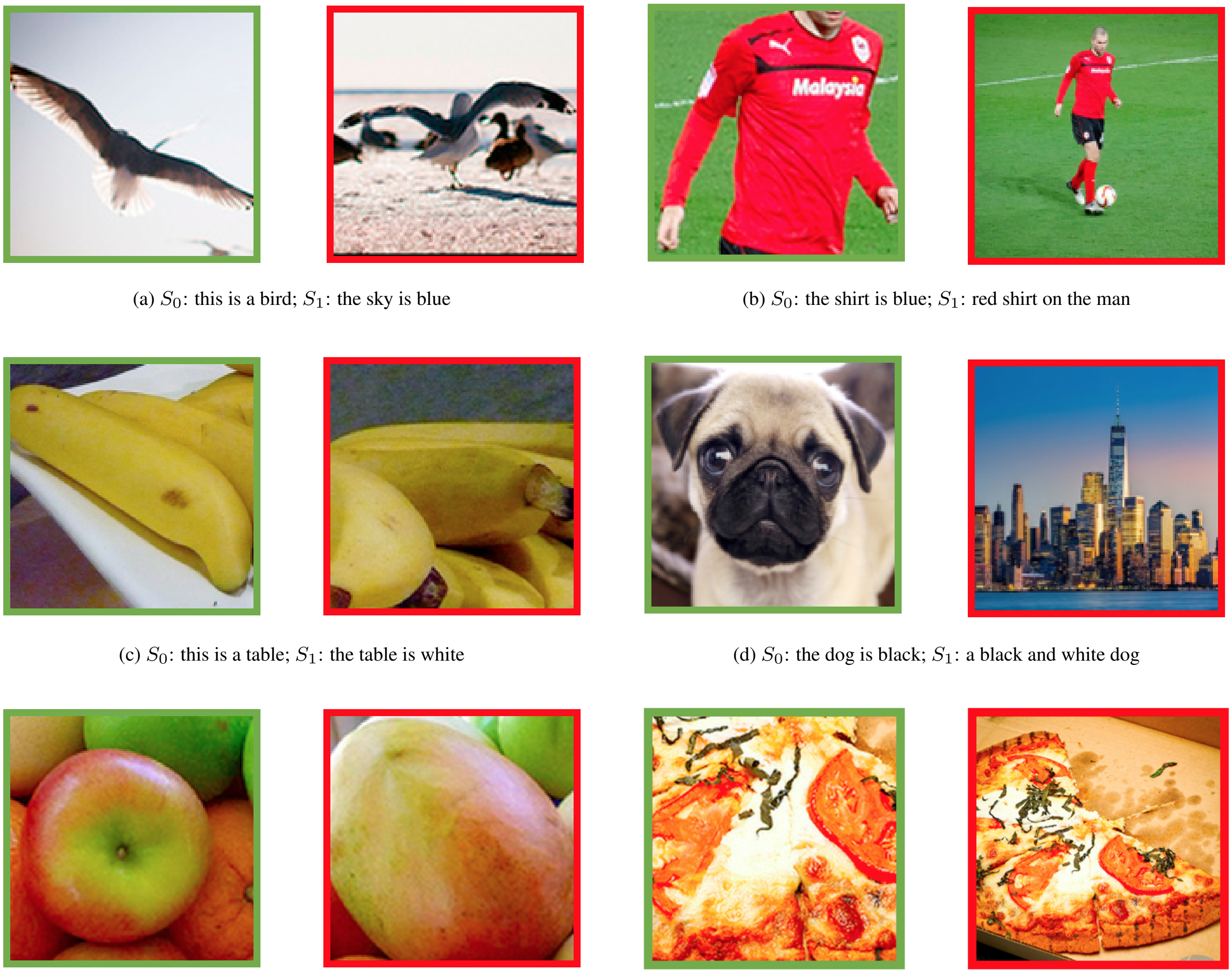


Figure 4: Loss on training and validation datasets while training the character-based modular $S_0$ model. The validation loss roughly followed the training loss, suggesting little overfitting.

Figure 5 shows a few examples derived from our model. The first example shows two images, each producing the same literal caption at the $S_0$ level. However, the $S_1$ model tries balance describing the target image well while not also describing the distractor image, and does so by choosing a feature of the target image, the blue sky, that is not as salient in the distractor.

Generally, we also observed that $S_1$ generated longer, more informative captions so as to better describe the target image. An example of this is part (d) of Figure 5, where despite the relative unrelatedness of the distractor image, the $\mathrm{S}_1$ produces a more informative caption than the $\mathrm{S}_0$.

Often, $S_1$ improved the caption quality. An example of this is the second example in which the $S_1$ output for the



(a) $S_0$: this is a bird; $S_1$: the sky is blue

(b) $S_0$: the shirt is blue; $S_1$: red shirt on the man

(c) $S_0$: this is a table; $S_1$: the table is white

(d) $S_0$: the dog is black; $S_1$: a black and white dog

(e) $S_0$: the fruit is green; $S_1$: a yellow apple

(f) $S_0$: this is a pizza; $S_1$: the meat on the pizza

Figure 5: Examples of image pairs and resulting captions describing the target image. Target images are on the left (outlined in green) and distractor images are on the right (outlined in red). The $S_0$ caption refers to the literal caption and $S_1$ refers to the pragmatic caption in which the model tries to minimize the ambiguity of which image the model is describing.

target image correctly describes the shirt as red, while the $\mathrm{S}_0$ describes it as blue.

Traditionally, quantitatively evaluating image captions is a difficult task due to the large number of ways an image could be described. Human-generated evaluations are often collected using online crowd-sourcing, such as through Mechanical Turk. Some standardized metrics such as BLEU, CIDEr, or METEOR scores exist. We chose to evaluate $S_0$ with BLEU scores ([12]).

Pragmatic image captioning is even more challenging to evaluate quantitatively. We considered a form of evaluation in which a separate listener model $\mathrm{L}_0$ would first be trained to take two images and a caption, and adduce which caption was being referred to. Our plan was to then compare the accuracy scores for $\mathrm{L}_0$ given $\mathrm{S}_0$ and $\mathrm{S}_1$ captions, with the hope of showing the latter to produce more accurate results. Unfortunately, we were unable to train an $\mathrm{L}_0$ with sufficient accuracy to perform this task at all.

Another limitation of such an approach is that both models may interpret parts of the image incorrectly in the same way and give the caption an exaggerated score, as discussed in ([11]).

Thus, we simply used BLEU scores to evaluate our $\mathrm{S}_1$ as well. The results for the $\mathrm{S}_0$ and $\mathrm{S}_1$ BLEU scores are summarized in Table 1. The character-based models output higher BLEU scores than the word models. Both the modular $S_0$ word and character approaches had higher or equal BLEU scores than their corresponding $S_1$ scores, which was surprising because, qualitatively, it seemed as if the $S_1$ model outputted more descriptive and accurate captions. In addition, the character end-to-end model had the highest BLEU score as compared to the rest of the variations. Again, this opposed the qualitative results in that the modular model seemed to work better than the end-to-end.

Table 1: Quantitative Analysis of Model

| Model Version | BLEU Score |
|---|---|
| Word $S_0$ Modular | 0.11727 |
| Word $S_1$ Modular | 0.11182 |
| Character $S_0$ Modular | 0.21509 |
| Character $S_1$ Modular | 0.18893 |
| Character End-to-End | 0.22616 |

.

## 6. Conclusion and Future Work

Our findings consist of two separate results. Firstly, we find that image captioning in general is possible with a character level LSTM. Secondly, we find that pragmatic image captioning, in the spirit of ([14]) is possible with a character level model, so long as beam search is used at the full width (i.e. the number of characters, here 39).

One extension of the systems of ([14]) and ([11]) that we plan to explore is the use of an attention mechanism in the style of ([15]). As well as improving the quality of the image captioning in question, this will also allow us to provide a visual form of model evaluation - whereas ([15]) visualizes the attention over a single image during the unrolling of a caption, we hope to visualize the attention over both the target image and distractor.

We further envision that rather than inputting two regions of a given image into our system as separate inputs, we could train a model which takes a whole image as input, but directs attention on a particular region. Non-end-to-end pragmatic caption generation over two regions in an image could then involve the same input with two different input attention vectors.

A further direction to pursue would be the use of randomly chosen distractors in the training of our system. Since humans tend to maximize informativity when captioning even without the presence of distractor images.

## 7. Citations

This project was based on the work started by Reuben Cohn-Gordon in CS224n (previous report attached).

The following were adapted from prior work and fit into our model:

- ResNet50 model for Keras from ([4]).
- BLEU Score Method from CS231n Homework ([6]).
- Glove vectors for pretrained word embeddings ([13]).

## References

[1] N. D. Goodman and M. C. Frank. Pragmatic language interpretation as probabilistic inference. *Trends in Cognitive Sciences*, 20(11):818–829, 2016.

[2] N. D. Goodman and A. Stuhlmüller. Knowledge and implicature: Modeling language understanding as social cognition. *Topics in cognitive science*, 5(1):173–184, 2013.

[3] H. P. Grice. Logic and conversation. *1975*, pages 41–58, 1975.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[5] 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.

[6] A. Karpathy, J. Johnson, and L. Fei-Fei. Cs231n. 2017.

[7] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush. Character-aware neural language models. *arXiv preprint arXiv:1508.06615*, 2015.

[8] E. Krahmer and K. van Deemter. Computational generation of referring expressions: A survey. *Comput. Linguist.*, 38(1):173–218, Mar. 2012.

[9] 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. 2016.

[10] G. Leech. *Principles of Pragmatics*. Longman linguistics library ; title no. 30. Longman, 1983.

[11] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. pages 11–20, 2016.

[12] 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*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[13] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[14] R. Vedantam, S. Bengio, K. Murphy, D. Parikh, and G. Chechik. Context-aware captions from context-agnostic supervision. *arXiv preprint arXiv:1701.02870*, 2017.

[15] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.