Image Caption Validation

Ryan McCaffrey Ioannis Christos Karakozis
Princeton University
{rm24,ick}@princeton.edu

Abstract

Our paper introduces the novel task of image caption validation and a first attempt towards achieving high performance on this task using deep learning models trained on MS COCO and ReferIt. Image caption validation is the task of classifying whether an image caption correctly describes the contents of the image. Image captions capture information about objects, their attributes and their relationships with other objects in the image. Our system parses this textual information using a GloVe word embedding and an LSTM network and compares it against the visual features extracted from the image using a convolutional neural network to realize whether the textual features match the contents of the image or of a portion of the image. Our best model achieves 88.62% accuracy on MS COCO. However, our qualitative analysis shows that the development of a dataset tailored for image caption validation is needed for the training of a model that is able to handle the more nuanced caption validation cases. Our codebase developed in Python using Tensorflow can be found by clicking here.

1. Introduction

Recent tasks in computer vision, such as Visual Question Answering (VQA) [2, 13], Caption Grounding [7], and Binary Image Segmentation guided by Image Captions [6], use text in the form or captions or questions in addition to images as input. The models developed for these tasks attempt to leverage the extra information provided by these natural language expressions, but in doing so, they make one implicit assumption: the textual input correctly represents the image. In a VQA model that takes an input question "Are these people family?", the model assumes that the question is associated with an image that depicts more than one person [13]. However, if there are no people in the image, the expected behavior of the model is undefined, as it is trained to produce an answer to the question, even when the question has no relevance to the paired image.

As computer vision models that work at this boundary of language and image understanding get more accurate, we want to make sure that is because the models are truly understanding the relationship between the input text and image. As a motivating example, consider the question "Is the kid riding the bicycle?" We would want a VQA model to differentiate between the two types of "no" responses that can be given for this question: is it because the kid in the image is not riding the bicycle or because there is no kid and no bike?

- **2** (score: 12.81 = 3.13 [image] + 9.68 [word])
- **1** (score: 11.29 = 1.88 [image] + 9.42 [word])
- **4** (score: 11.04 = 2.72 [image] + 8.32 [word])

Based on image only: chair (4.74), coffee table (4.60), ceiling (4.48),
Based on word only: 2 (9.68), 1 (9.42), 4 (8.32),

Figure 1: Example of how the iBOWIMG VQA model gives back meaningless output in questions that do not correctly describe the contents of the target image, such as "How many elephants are next to the giraffe?" [13].

The tendency of VQA models to assume some truthiness in their textual inputs is encapsulated in the work performed by Ganju *et al*. [3], whose model leverages secondary questions about an image to infer information about what the image is depicting and better answer the main question at hand. Their findings indicate that the questions used to train modern VQA models are almost always relevant to the im-

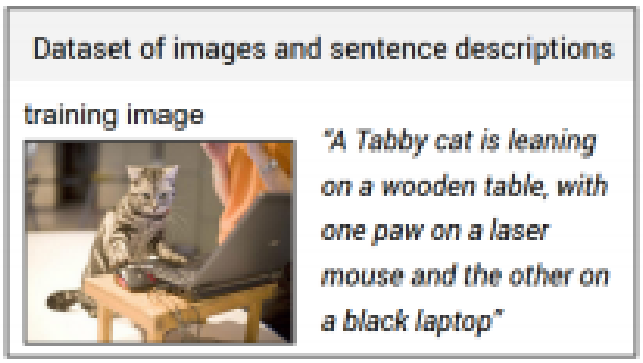

Figure 2: Example of an image and a caption that is highly related to the image. Figure borrowed from [7].

age, which leads to models expecting such questions as input. When a VQA model like iBOWIMG [13] is fed the image in Figure 1 and the question "How many elephants are next to the giraffe?", the output returned by the model is not only nonsensical, but the model might also try to draw relationships between the image and irrelevant question. Ideally, VQA models will understand the nonsensical relationship of the question in the context of the target image and will generate a more appropriate response as a result.

This desired behavior of understanding the relationship between text and images further extends to caption-related tasks. Figure 2 depicts an image and its corresponding caption that were used as inputs to a Caption Grounding model developed by Karpathy and Li in [7]. The caption describes very precisely the objects present in the image, their attributes, and some of the core relationships between them. However, given a caption that is a bad description of the image, how should models like [6, 7], that aim at drawing connections between the textual and visual inputs, behave? What should the expected output of the system be when a user inputs a completely unrelated caption of the form "A human riding a horse"? Even more importantly, what should the grounding be if the caption has more nuanced differences from the ground truth, like "A black cat leaning on an iron table with both paws on a laser mouse"? A human would clearly realize that this is a bad caption of the image, but would a Caption Grounding or Caption Segmentation model realize that and return no groundings and no foreground segmentation, respectively?

Our research introduces the task of Image Caption Validation: given an image and a textual input (in the context of this paper, an image caption that may or may not describe the image well), validate whether the textual input is a good description of the image with a binary output. A good description in the context of this research is one that is correctly paired with its respective image in the datasets that are used. Whereas the caption provided in Figure 2 would be seen as a valid caption with its paired image, we would want a model to understand that variants such as "a dog", "a black cat" or "a cat under a wooden table" are not valid due to object, object attributes, or object relationships inconsistencies.

Inspired by [6], we introduce a model with two legs: a convolutional neural network for visual feature extraction and a word embedding layer followed by a Long-Short-Term-Memory (LSTM) network [5] for textual input encoding. Our model combines the signals from the two legs and then produces a binary output for whether or not the given expression correctly describes the image.

2. Related work

Visual feature extraction: Our work, like much research that intertwines text and images, premises itself on having a strong model for image feature extraction. Recent work has shown a trend towards pre-trained Convolutional Neural Nets (CNNs) on large image repositories, such as ImageNet [3, 13]. Earlier layers of the CNN weights are frozen to accelerate training, while the final layers are fine-tuned during training with the rest of the model to improve accuracy of image feature extraction on a per-application basis. Deep CNN architectures, such as the one proposed by the ImageNet Challenge 2014 winning model VGG-16 [11], has shown that architectures with many small convolutional layers can boost accuracy without losing generalisability across datasets. Our model utilizes VGG-16 for image feature extraction for this reason.

Segmentation using natural language: Research performed by Hu *et al*. [6] explores the intersection of image and text by using a natural language expression to inform its model on which objects to segment within the provided image. The model developed by this group uses the two-legged approach that inspired our architecture: a CNN for visual feature extraction paired with a word embedding layer and Long-Term-Short-Term-Memory (LSTM) network for textual encoding, which are then concatenated and classified to produce outputs that take signals from each of the two legs.

To the best of our knowledge, models that perform segmentation over natural language are designed to optimize segmentation behavior for natural language expressions that accurately describe a portion of the image, but they do not account for when an expression weakly describes the image or is not relevant to the image at all. In the situation of an irrelevant caption, these models will produce an erroneous segmentation that can be corrected if the model were to check for caption relevancy.

Visual question answering: Recent work by Ganju *et al*. has looked to use information from visual questions to better inform a model in the VQA task and, as a result, has

shown that providing multiple questions for a single image can improve performance in the VQA task [3]. While we conjecture a similar behavior of improved performance in using multiple natural language expressions for a single image, we note the danger of assuming the truthfulness of a question paired with an image. For a question such as "What breed of dog is that?", the model may erroneously learn an image feature to be a dog when there is no dog present in the image.

This sort of image caption validation has been partially implemented by Goyal *et al*. in their dataset that attempts to elevate the role of image understanding in VQA [4]. The research performed by this group sought to counter language biases in VQA by collecting complementary images such that each question in the dataset is associated with pairs of images that result in different answers to the question. For questions that ask about the presence of a set of objects in an image (e.g. "Is there a cat playing with a red ball?"), this approaches image caption validation in that the model learns to detect whether or not the set of objects is a valid caption for the image (e.g. "A cat plays with a red ball."). However, this model still fails for when a question assumes the presence of an object, such as "Is the TV turned on?", when there is, in fact, no television present in the image.

3. Model Architecture

Our model borrows heavily from the architecture described by Hu *et al*. in [6], with several important changes. Given an image and an image caption, the goal is to produce a binary output that validates whether or not the given caption accurately describes the image. We seek to achieve this goal with two of the same components and a modified third component used in [6]: 1) a natural language expression encoder based on a pre-trained GloVe word embedding [10] and a recurrent LSTM network [5], a convolutional network to extract local image descriptors and generate a spatial feature map, and a fully-connected network to classify the concatenated encoded expression and image descriptors. Figure 3 is a graphical representation of our model.

3.1. Spatial feature map extraction

Given an input image, we want to produce a feature map that accurately and uniquely represents the image, such that the important objects are easily identified. In our model, we adopt VGG-16, which uses a series of convolutional and pooling layers to take an image of dimension $W \times H$ and produce a spatial feature map with dimensions $w \times h$ [11]. In this representation, each position of the feature map has D_{im} channels, which are the D_{im} local descriptors for each position in the map. For the VGG-16 implementation used in our model, we set $D_{im} = 1000$, $W = H = 224$, and $w = W/s$ and $h = H/s$, where $s = 32$ is the pixel stride on the output of fc8 layer. Like Hu *et al*., we perform L2-normalization on the D_{im} dimensional vector at each position in the extracted feature map [6]. This produces a feature map of dimension $w \times h \times D_{im}$.

3.2. Encoding captions with LSTM network

For each caption associated with an input image, we want to represent the expression as a fixed-size vector. To achieve this, we follow the model described in [6]: we embed each word into a vector through a word embedding matrix, and then use a recurrent Long-Short Term Memory (LSTM) [5] network to scan through the embedded word sequence. For a given natural language expression with T words, at each time step t, the LSTM takes vectorized word w_t from the word embedding matrix as input and updates its hidden state, which is D_{text}-dimensional. When $t = T$, we use the hidden state h_T of the LSTM as the encoded vector representation of the caption input. Following the procedure described in [6], the D_{text} dimensional vector h_T is L2-normalized, with $D_{text} = 1000$.

For the word embedding matrix, we initially adopted the method used by Hu *et al*. in [6], where the weights of the embedding matrix are randomly initialized and then trained with the rest of the model. The alternative we considered was to use a pre-trained word embedding matrix, the Global Vectors for Word Representation (GloVe) matrix [10]. Given the latter approach yielded significantly better performance during our initial experiments, we decided to use the pre-trained word embedding for the textual input encoding.

3.3. Feature concatenation and caption validation

To perform a classification over both the encoded natural language expression and extracted feature map, we concatenate the encoded expression h_T to the local descriptor at every location in the spatial grid. The spatial map is now $w \times h \times (D_{im} + D_{text})$ dimensional and contains information from both the image and the language expression. This will be the input to a fully-connected classifier to produce a binary image caption validation output.

The classification network is a 2-layer fully-connected network separated by a ReLU activation, similar to the network used in [6], though ours is fully-connected rather than convolutional. The classification network is applied over the $w \times h$ feature map, with a ReLU nonlinearity between them and a hidden layer with dimensionality $D_{cls} = 500$. Our 2-layer fully-connected network outputs a single value, representing whether or not the encoded language expression accurately captions the given feature map.

During training, each training instance exists as a tuple (I, S, B), where I is the image, S is the natural language expression that may or may not describe the image well, and B is a binary value 0 or 1, where $B = 0$ means the expression does not describe the image, and $B = 1$ implies

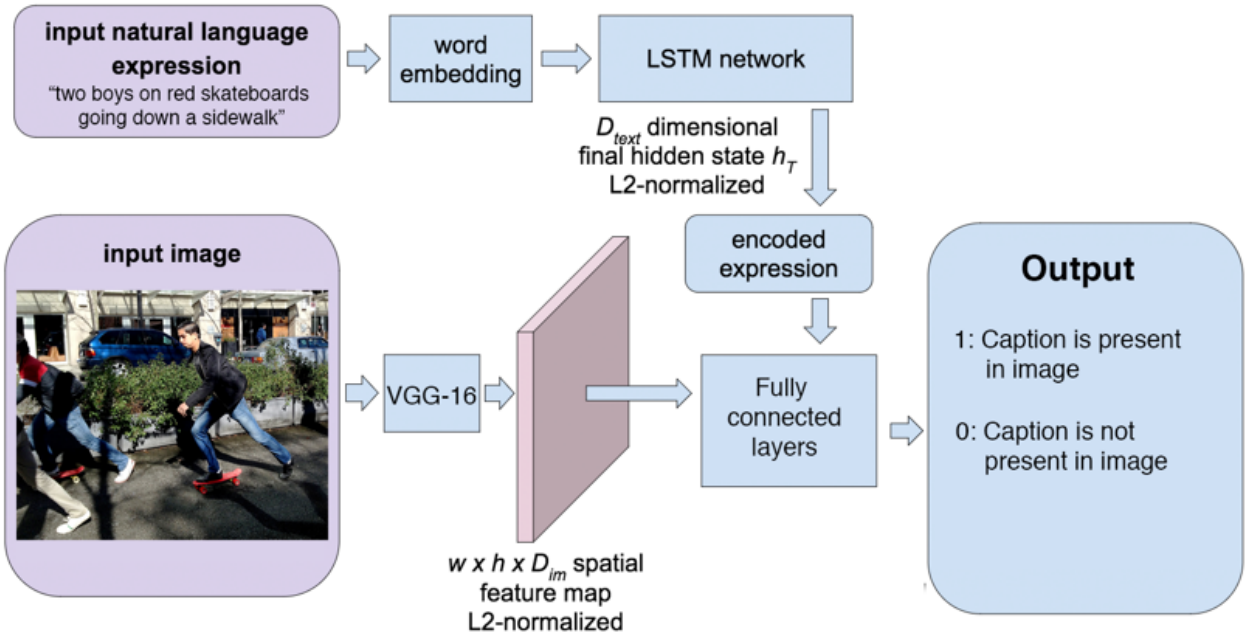

Figure 3: Our proposed model for image caption validation. The model has three main coponents: a caption encoder based on a recurrent LSTM, a convolutional network to generate a spatial feature map, and a fully connected classification network to produce our binary validation output.

that it does. The loss function is the binary logistic loss:

$$L = -z \log(\sigma(x)) - (1 - z) \log(1 - \sigma(x))$$

where z is an image/caption label (1 if the caption accurately describes the image and 0 if it does not), x is the model prediction, and $\sigma(x)$ is the logistic function on x:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

The word embedding matrix is initialized using the GloVe pre-trained weights and remains frozen throughout training. The parameters in the image feature map extraction network are initialized from a VGG-16 model pre-trained on ImageNet [11]. We freeze the convolutional layers of the VGG architecture and fine-tune only the fully connected layers of the architecture during training. The LSTM weights and fully-connected classification layer weights are randomly initialized and are also fine-tuned during training. We use L2-regularization to control the magnitude of the weight vectors. The network is trained end-to-end with standard backpropagation using SGD with momentum.

4. Experiments and Evaluation

4.1. Dataset Selection

Since we introduce a novel task, there is no established dataset for Image Caption Validation. Thus, we resort to two datasets with a large number of image-caption annotations: ReferIt and MS COCO [8, 9].

ReferIt contains nearly 20k images of natural scenes with a total of approximately 130k expressions that refer to different objects in the scenes [8]. We use ReferIt because its image captions are expressions referring to part of the image, which allows us to train models that can better validate captions that partially describe images. However, some ReferIt captions, such as "sky" and its variants, are repeated across a large number of images, making negative examples of non-related images and captions difficult to generate, as it will be analyzed in the next section.

Given the above limitation of Referit and the fact that many captions are not grammatically complete and lexicographically correct, we need an alternative training set [8]. As a result, our model is also trained on MS COCO, which contains 328k images, each with five unique captions that are grammatically complete and lexicographically correct phrases [9].

					Test Accuracy			
Model Name	**Embedding**	**Training Set**	**Training Set Size**	**Training Accuracy**	**Simple Captions**	**Irrelevant Captions**	**Concatenated Captions**	**Object Class**
cls_coco_glove	GloVe (50d)	COCO	90000 (9000 images)	91.56%	86.59%	82.72%	67.04%	59.84%
cls_referit_glove	GloVe (50d)	ReferIt	90000 (9789 images)	80.82%	73.01%	66.32%	58.07%	**64.98%**
cls_coco_glove+	GloVe (300d)	COCO	90000 (9000 images)	95.12%	88.60%	86.10%	68.61%	62.66%
cls_coco_glove++	GloVe (300d)	COCO	90000 (45000 images)	95.15%	**88.62%**	**86.12%**	**69.34%**	63.37%

Figure 4: Performance summary of the four core models across all experiments.

4.2. Baseline Models

Since we found no previous work on Image Caption Validation, there is no benchmark against which we can compare our model. Thus, we train two baseline models using the Section 3 architecture to compare against one another.

The first model, **cls_coco_glove**, is trained using the 50-dimensional GloVe embedding [10] on a subset of the MS COCO 2017 training set [9] comprised of 90,000 image-caption pairs. Only 9,000 images are used from the MS COCO training set, each of which is paired with 10 captions. The positive examples are the 5 captions paired with the image in MS COCO and the remaining five are negative examples that are generated by randomly sampling captions from the rest of the MS COCO training set. We do not train on the full MS COCO training set because of computational resource limitations.

The second model, **cls_referit_glove**, is trained using the 50-dimensional GloVe embedding [10] on a subset of the training set of ReferIt [8] comprised of 90,000 image-caption pairs. The number of images in the training set are 9,789. This is not 9,000 because Referit images have a variable number of captions [8]. The negative examples are generated by randomly sampling captions from the rest of the ReferIt training set so they match the number of positive examples per image. We do not train on the full ReferIt training set because of computational resource limitations.

The method of generating negative captions is not ideal, as there is always the risk of sampling a negative caption that actually describes the target image well. This results in noise in the negative training samples, which hinders test-time performance. Given COCO captions provide a very detailed description of the image they are paired with, it is highly unlikely that a randomly sampled caption describes well the target image [9]. In the case of the second model though, the noise in the negative captions is more significant. This is because ReferIt captions refer to a portion of the image, which make certain captions appear across images, such as the most prevalent "sky" caption and its variants [8].

4.3. Test-time Evaluation

The results of our experiments are summarized in Figure 4. For this section, focus on the first two rows. All test-time experiments are performed on the MS COCO 2017 validation set, comprised of 5,000 images. We use MS COCO over ReferIt since the MS COCO captions are much better resemble spoken English and encode more complex relationships about the objects present in the image, than the much shorter and simpler referring expressions of ReferIt [8, 9]. This also allows for the generation of much less noisy negative examples. Here is a breakdown of each experiment we run:

1. **Simple Captions:** We use the 5 image captions of each image as positive examples and we generate 5 negative examples using the method described above. This is the key comparison metric, as it tests the models on instances similar to the ones they are trained on.

2. **Irrelevant Captions:** We use only 1/10 of the images in the test set and we assign 50 negative examples to each image, resulting in 25,000 pairs of images and negative captions. This a sanity-check of the extent to which our models can correctly identify captions that are obviously unrelated to the image as negative, which is one of the two fundamental goals of Image Caption Validation.

3. **Concatenated Captions:** We evaluate the ability of our models to validate more complex captions. We do so by forming for each image four types of complex captions: 1) two positive captions concatenated into one, 2) two negative captions concatenated into one, 3) one positive and one negative caption concatenated in this order, 4) same as (3) but reverse order. The captions are concatenated using the -and- conjunctive.

4. **Object Class:** MS COCO has 80 object classes [9]. In this experiment, we evaluate the ability of our models to identify the object classes present in the image, simulating the traditional image classification task [11]. For each image, we form positive captions by using the COCO object class definition of the object classes present in the image and we form negative captions by randomly sampling an equal number of the remaining object classes.

cls_coco_glove outperforms **cls_referit_glove** in almost all experiments, which is not surprising given the former has been trained on captions more similar in structure and vocabulary to the ones found in the test set. As shown in

Black Cat - cls_coco_glove	Semantic Focus:	Object Class		Color	Relative Position	Context	Counting
		Foreground	Background				
Positive Captions	Caption	cat	flowerpot	black cat	cat in front of flower pots	cat in the garden	one cat
	Label	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
	Score	2.1575	1.2415	2.3877	4.5698	3.1794	2.4131
		Different	Similar				
Negative Captions	Caption	bridge	dog	green cat	cat behind flower pots	cat on a highway	two cats
	Label	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE
	Score	-1.8931	2.0468	-1.5117	2.9832	-0.1182	1.449
Red Bridge - cls_coco_glove	Semantic Focus:	Object Class		Color	Relative Position	Context	Counting
		Foreground	Background				
Positive Captions	Caption	bridge	hill	red bridge	bridge above the sea	bridge during the day	one bridge
	Label	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
	Score	2.6962	0.7724	2.2783	3.1548	4.534	3.2834
		Different	Similar				
Negative Captions	Caption	cat	tunnel	green bridge	bridge under the sea	bridge during the night	five bridges
	Label	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
	Score	-3.6501	2.4202	1.6025	3.2134	4.5189	3.2449

Figure 5: Logit scores and caption validation labels produced by **cls_coco_glove** for various captions for the black cat and red bridge images presented in Figure 6.


Figure 6: Black Cat image from MS COCO 2017 [9] and Red Bridge image from ReferIt [8] that were used for the qualitative analysis.

Figure 4, the accuracy for both Simple Captions and Irrelevant Captions is high, indicating that we successfully differentiate between relevant and irrelevant captions at a very satisfactory level. However, once the queries become more complex, performance drops significantly, as seen in the Concatenated Captions experiment. This is an indication that our models struggle at invalidating more nuanced captions that only partly describe the image correctly, since we observe the main source of error are concatenated captions of type (3) and (4).

The even more surprising result is that **cls_referit_glove** outperforms **cls_coco_glove** in the Object Class experiment. We could not pinpoint the exact reason why this happens, so this is definitely an area of future work. However, we identify that the relatively low performance of the models in this experiment is a result of the models overwhelmingly classifying object class captions as positive, especially the more similar the negative captions are in nature to the ground-truth, as our qualitative analysis in the next section shows.

To further understand the lower performance of the models in the last two experiments, we perform some qualitative analysis on carefully crafted pairs of image-caption.

4.4. Qualitative Analysis

Despite MS COCO being a very comprehensive dataset, it is not one tailored for Image Caption Validation. This is because there are no carefully crafted negative captions for its images, which forces us to use the method described in Section 4.2 to generate them. The issue with this approach is that a caption will either be perfectly describing the image or be completely unrelated to it. No negative caption will have nuanced differences from what its image is depicting. For example, if we consider Figure 2, a nuanced negative caption would be "A black cat leaning on an iron table with both paws on a laser mouse". Such a caption correctly identifies the core objects int he image ("cat", "table", "laptop", "laser mouse"), but does not correctly identify their attributes (e.g. "tabby" vs "black") and the relationships between them (e.g. "both paws on" vs "one paw on"). Given such captions are not present in MS COCO and, thus, our test set, some manual qualitative analysis is required to check the extent to which our models can correctly validate such more nuanced captions.

The qualitative analysis is performed using the **cls_coco_glove** model. The summary of our qualitative analysis is presented in Figure 5. It shows that our model is very good at validating captions whose object classes are present in the image. It is also very good at invalidating captions whose object classes are not present in the image and are dissimilar from the object classes depicted, such as "cat"

Black Cat	**Caption**	a living room with white curtains	two people sitting on bikes beside each other.	this is a bad caption
	Label	FALSE	FALSE	FALSE
	Score	-2.5876	-3.5939	-1.4228
Red Bridge	**Caption**	a living room with white curtains	two people sitting on bikes beside each other.	this is a bad caption
	Label	FALSE	FALSE	FALSE
	Score	-6.8357	-1.8869	-6.5727

Figure 7: Logit scores and caption validation labels for captions irrelevant to the images depicted in Figure 6.

vs "bridge". However this is not the case for object classes that are similar to the ones depicted in the image, such as "tunnel" vs "bridge" and "cat" vs "dog". Thus, it seems our models are able to identify correctly the general nature of the main objects depicted (e.g. animal, structure) and not their specific object class. What is very interesting though is that our model is able to do that even for item categories that are not one of the 80 object classes of MS COCO, such as "bridge" and "hill" [9].

This qualitative analysis also validates our hypothesis. The models are not able to validate correctly nuanced negative captions that identify the objects depicted correctly but misidentify object attributes and relationships, such as color, relative position, context, and number of object instances. According to this, it seems our models are validating any caption that references the objects present in the image, regardless of whether the attributes and relationships of those are correctly identified. This is justified from the highly similar high confidence-logit scores that captions in the color, relative position and counting semantic category receive as long as they identify the main image object (i.e. "bridge" or "cat'), regardless of whether they are negative or positive captions. This hypothesis is further supported from Figure 7, which shows that our models succeed in invalidating negative captions with high confidence-logit scores when those captions make no references to the core object classes presented.

4.5. Higher Dimensionality Word Embedding:

So far our models have succeeded in two ways: 1) they are invalidating negative captions that are completely unrelated to the target image, which was one of our core goals and 2) they are correctly identifying the general nature of the objects depicted in the image, even if those are not one of the core object classes of the training set. However, (2) is not sufficient, as it leads to the limitations identified through our qualitative analysis. To tackle this limitation, we need to differentiate more effectively between concepts that lie very close to each other in the GloVe embedding parameter space due to being used in very similar sentence contexts, such as the words "dog" and "cat" (or any other domestic animal in this case) [10]. To do so, we introduce **cls_coco_glove+**, which is the same model as **cls_coco_glove**, but is trained on the 300-dimensional GloVe word embedding [10].

We re-run the experiments introduced in section 4.1 to evaluate our new model. **cls_coco_glove+** outperforms the other two models in all categories, but Object Class, according to the results summarized in Figure 4. Since it gains a significant improvement over **cls_coco_glove** across all experiments, the higher dimensionality embeddings indeed allow for a better encoding of the textual input. In fact, our new model better differentiates across object classes than the original COCO model due to the better performance of **cls_coco_glove+** over **cls_coco_glove** in the Object Class experiment. However, the fact our latest model still loses to **cls_referit_glove** in the Object Class experiment is an indication that there is still room for improvement in the way the model identifies the object classes present in the image.

4.6. Larger Effective Vocabulary:

COCO images and, by extension, their captions focus on 80 object classes, which leads to 80 words dominating the caption vocabulary [9]. By contrast, in ReferIt there are no explicit object classes. It is a dataset that aims for breadth over possible objects one can encounter in natural scenes and, thus, there are way more object concepts/classes that appear in its image captions [8]. This results in a much richer set of objects that the network learns by training on ReferIt rather than MS COCO.

Given the 5 captions of a COCO image are fairly similar in content to one another, we now use 1 positive and 1 negative caption per image. This allows us to train on a much larger set of images, which implies a wider set of captions in terms of their semantic content. We hope that this will increase the richness of the textual and visual elements of our training set in such a way, so that the trained model will be able to distinguish between objects of similar nature. The resulting model trained on 45,000 images from the MS COCO 2017 training set is **cls_coco_glove++**.

Our quantitative analysis shows that **cls_coco_glove++** does not achieve better performance than **cls_coco_glove+** in a statistically significant way, despite a small boost in experiments 3 and 4. Even if we increase the variance in semantic and visual content in the training set, there is minimal change in performance. Thus, given our architecture, it does not seem we can do much better at capturing the difference between semantically similar object classes by training on the MS COCO dataset.

This is further supported from the second stage of our qualitative analysis performed this time on **cls_coco_glove+** and **cls_coco_glove++**. The summary is presented in Figure 8. We observe that the labels are almost identical to the ones of the original qualitative analysis presented in Figure 5, which implies that both the higher and the lower dimensional model are able to validate correctly the same

Black Cat - cls_coco_glove++	Semantic Focus:	Object Class		Color	Relative Position	Context	Counting
		Foreground	Background				
Positive Captions	Caption	cat	flowerpot	black cat	cat in front of flower pots	cat in the garden	one cat
	Label	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
	Score	1.145	-3.4806	1.3988	3.7984	2.5408	1.3068
		Different	Similar				
Negative Captions	Caption	bridge	dog	green cat	cat behind flower pots	cat on a highway	two cats
	Label	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE
	Score	-5.7742	0.1694	0.0219	3.1177	0.5573	-0.0492
Red Bridge - cls_coco_glove++	Semantic Focus:	Object Class		Color	Relative Position	Context	Counting
		Foreground	Background				
Positive Captions	Caption	bridge	hill	red bridge	bridge above the sea	bridge during the day	one bridge
	Label	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
	Score	2.3838	1.0858	2.3365	2.9309	3.4711	2.7694
		Different	Similar				
Negative Captions	Caption	cat	tunnel	green bridge	bridge under the sea	bridge during the night	five bridges
	Label	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
	Score	-5.4305	1.0538	2.1594	2.9094	2.3732	2.6785

Figure 8: Logit scores and caption validation labels produced by **cls_coco_glove++** for various captions for the black cat and red bridge images presented in Figure 6. Results were similar for the **cls_coco_glove+** model, so they are omitted.

types of captions. In that sense, we have not overcome the difficulty of our models in invalidating nuanced negative captions by increasing the embedding dimensionality or increasing the effective vocabulary size.

However, the logit scores for the negative examples are now in most cases significantly smaller than the logit scores of the corresponding positive examples. Thus, since these results are similar across **cls_coco_glove+** and **cls_coco_glove++**, it is the higher dimensionality embeddings that allow the model to diffrentiate more effectively between semantically similar captions, such as ”bridge“ vs ”tunnel“ and ”one cat“ vs ”two cats“. However, the qualitative analysis still shows that our models suffer when having to validate captions that differ not in the object classes present but in the attributes of and relationships among those object classes.

5. Conclusions and Future Work

Although we have developed a model that can differentiate with 88.62% accuracy between highly relevant and highly irrelevant captions for an image, our model is far from perfect as it struggles at handling more nuanced cases. Our qualitative analysis has shown that our models do not invalidate more cleverly formed negative textual inputs, which correctly identify the objects present in the image but misidentify object attributes, such as color, or object relationships, such as relative position. Higher dimensionality word embeddings and a wider set of captions and images in the training set do improve performance, but they cannot solve this problem. Thus, we propose the following two areas of future work:

Alternative baselines - iBOWIMG: While there is no established research for image caption validation, there are existing models that can act as a baselines, given slight modifications. An example is iBOWIMG, a model widely used as a VQA baseline [1, 3, 12]. By editing each caption C to be a question ”Is there a C?“, we can reliably get reliable yes/no responses from iBOWIMG, which can be interpreted as image caption validation outputs. Given the ability of VQA models to answer yes/no questions with high success as seen in [3, 13], this could overcome the limitations of our current model.

Image Caption Validation Dataset: MS COCO and ReferIt are fairly rich datasets in image captions. However, neither of them has carefully crafted negative captions for each of its images. This implies that it is very hard for image caption validation models to learn to invalidate nuanced negative captions, as they are never presented such captions during training. Thus, the development of a dataset tailored for image caption validation models can lead to a significant boost in the ability of our models to invalidate nuanced negative captions. This could be done by expanding the MS COCO dataset to include negative captions for each image, such as those captions correctly identify the object classes present in the image, but misidentify object attributes and relationships, such as context, relative position, number of instances, color and texture.

6. Acknowledgements

We want to thank Professor Russakovsky for helping us better define our topic AND for listing [6] on the course schedule as we literally had no project topic with a good handle on up until we ran into this. We want to also thank the creators of that work that inspired the topic of this project. Most importantly, we want to thank our parents for raising us into good people that enjoy Computer Vision and are willing to pursue interesting Visual Recognition projects.

References

[1] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Learning to compose neural networks for question answering. *CoRR*, abs/1601.01705, 2016.

[2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: visual question answering. *CoRR*, abs/1505.00468, 2015.

[3] S. Ganju, O. Russakovsky, and A. Gupta. What's in a question: Using visual questions as a form of supervision. *CoRR*, abs/1704.03895, 2017.

[4] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. *CoRR*, abs/1612.00837, 2016.

[5] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[6] R. Hu, M. Rohrbach, and T. Darrell. Segmentation from natural language expressions. *CoRR*, abs/1603.06180, 2016.

[7] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. 39, 12 2014.

[8] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg. Referit game: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014.

[9] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

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

[11] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[12] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. *CoRR*, abs/1511.05234, 2015.

[13] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus. Simple baseline for visual question answering. *CoRR*, abs/1512.02167, 2015.