Every Picture Tells a Story: Generating Sentences from Images

Ali Farhadi[1], Mohsen Hejrati[2] , Mohammad Amin Sadeghi[2], Peter Young[1],
Cyrus Rashtchian[1], Julia Hockenmaier[1], David Forsyth[1]

[1] Computer Science Department
University of Illinois at Urbana-Champaign
{afarhad2,pyoung2,crashtc2,juliahmr,daf}@illinois.edu

[2] Computer Vision Group, School of Mathematics
Institute for studies in theoretical Physics and Mathematics(IPM)
{m.a.sadeghi,mhejrati}@gmail.com

Abstract. Humans can prepare concise descriptions of pictures, focusing on what they find important. We demonstrate that automatic methods can do so too. We describe a system that can compute a score linking an image to a sentence. This score can be used to attach a descriptive sentence to a given image, or to obtain images that illustrate a given sentence. The score is obtained by comparing an estimate of meaning obtained from the image to one obtained from the sentence. Each estimate of meaning comes from a discriminative procedure that is learned using data. We evaluate on a novel dataset consisting of human-annotated images. While our underlying estimate of meaning is impoverished, it is sufficient to produce very good quantitative results, evaluated with a novel score that can account for synecdoche.

1 Introduction

For most pictures, humans can prepare a concise description in the form of a sentence relatively easily. Such descriptions might identify the most interesting objects, what they are doing, and where this is happening. These descriptions are rich, because they are in sentence form. They are accurate, with good agreement between annotators. They are concise: much is omitted, because humans tend not to mention objects or events that they judge to be less significant. Finally, they are consistent: in our data, annotators tend to agree on what is mentioned. Barnard *et al.* name two applications for methods that link text and images: **Illustration**, where one finds pictures suggested by text (perhaps to suggest illustrations from a collection); and **annotation**, where one finds text annotations for images (perhaps to allow keyword search to find more images) [1].

This paper investigates methods to generate short descriptive sentences from images. Our contributions include: We introduce a dataset to study this problem (section 3.1). We introduce a novel representation intermediate between images and sentences (section 2.1). We describe a novel, discriminative approach that produces very good results at sentence annotation (section 2.4). For illustration, out of vocabulary words pose serious difficulties, and we show methods to use distributional semantics to cope with these issues (section 3.4). Evaluating sentence generation is very difficult, because sentences are fluid, and quite different sentences can describe the same phenomena. Worse, synecdoche (for example, substituting “animal” for “cat” or “bicycle” for “vehicle”) and the general richness of vocabulary means that many different words can quite legitimately be used to describe the same picture. In section 3, we describe a quantitative evaluation of sentence generation at a useful scale.

Linking individual words to images has a rich history and space allows only a mention of the most relevant papers. A natural strategy is to try and predict words from image regions. The first image annotation system is due to Mori *et al.* [2]; Duygulu *et al.* continued this tradition using models from machine translation [3]. Since then, a wide range of models has been deployed (reviews in [4, 5]); the current best performer is a form of nearest neighbours matching [6]. The most recent methods perform fairly well, but still find difficulty **placing** annotations on the correct regions.

Sentences are richer than lists of words, because they describe activities, properties of objects, and relations between entities (among other things). Such relations are revealing: Gupta and Davis show that respecting likely spatial relations between objects markedly improves the accuracy of both annotation and placing [7]. Li and Fei-Fei show that event recognition is improved by explicit inference on a generative model representing the scene in which the event occurs and also the objects in the image [8]. Using a different generative model, Li and Fei-Fei demonstrate that relations improve object labels, scene labels and segmentation [9]. Gupta and Davis show that respecting relations between objects and actions improve recognition of each [10, 11]. Yao and Fei-Fei use the fact that objects and human poses are coupled and show that recognizing one helps the recognition of the other [12]. Relations between words in annotating sentences can reveal image structure. Berg *et al.* show that word features suggest which names in a caption are depicted in the attached picture, and that this improves the accuracy of links between names and faces [13]. Mensink and Verbeek show that complex co-occurrence relations between people improve face labelling, too [14]. Luo, Caputo and Ferrari [15] show benefits of associating faces and poses to names and verbs in predicting “who’s doing what” in news articles. Coyne and Sproat describe an auto-illustration system that gives naive users a method to produce rendered images from free text descriptions (Wordseye; [16];`http://www.wordseye.com`).

There are few attempts to generate sentences from visual data. Gupta *et al.* generate sentences narrating a sports event in video using a compositional model based around AND-OR graphs [17]. The relatively stylised structure of the events helps both in sentence generation and in evaluation, because it is straightforward to tell which sentence is right. Yao *et al.* show some examples of both temporal narrative sentences (i.e. this happened, then that) and scene description sentences generated from visual data, but there is no evaluation [18]. These methods generate a direct representation of what is happening in a scene, and then decode it into a sentence.

An alternative, which we espouse, is to build a scoring procedure that evaluates the similarity between a sentence and an image. This approach is attractive,

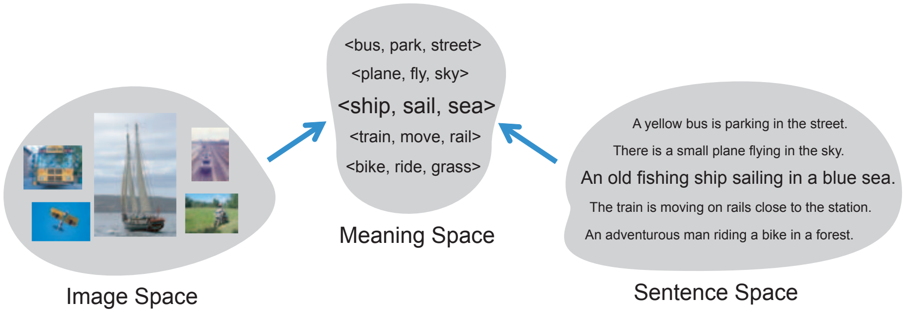

<bus, park, street>
<plane, fly, sky>
<ship, sail, sea>
<train, move, rail>
<bike, ride, grass>
Meaning Space
A yellow bus is parking in the street.
There is a small plane flying in the sky.
An old fishing ship sailing in a blue sea.
The train is moving on rails close to the station.
An adventurous man riding a bike in a forest.
Image Space
Sentence Space

Fig. 1. There is an intermediate space of meaning which has different projections to the space of images and sentences. Once we learn the projections we can generate sentences for images and find images best described by a given sentence.

because it is symmetric: given an image (resp. sentence), one can search for the best sentence (resp. image) in a large set. This means that one can do both illustration and annotation with one method. Another attraction is the method does not need a strong syntactic model, which is represented by the prior on sentences. Our scoring procedure is built around an intermediate representation, which we call the **meaning** of the image (resp. sentence). In effect, image and sentence are each mapped to this intermediate space, and the results are compared; similar meanings result in a high score. The advantage of doing so is that each of these maps can be adjusted discriminatively. While the meaning space could be abstract, in our implementation we use a direct representation of simple sentences as a meaning space. This allows us to exploit distributional semantics ideas to deal with out of vocabulary words. For example, we have no detector for "cattle"; but we can link sentences containing this word to images, because distributional semantics tells us that a "cattle" is similar to "sheep" and "cow", etc. (Figure 6)

2 Approach

Our model assumes that there is a space of *Meanings* that comes between the space of *Sentences* and the space of *Images*. We evaluate the similarity between a sentence and an image by (a) mapping each to the meaning space then (b) comparing the results. Figure 1 depicts the intermediate space of meanings. We will learn the mapping from images (resp. sentences) to meaning discriminatively from pairs of images (resp. sentences) and assigned meaning representations.

2.1 Mapping Image to Meaning

Our current representation of meaning is a triplet of $\langle object, action, scene\rangle$. This triplet provides a holistic idea about what the image (resp. sentence) is about and what is most important. For the image, this is the part that people would talk about first; for the sentence, this is the structure that should be preserved in the tightest summary. For each slot in the triplet, there is a discrete set of possible

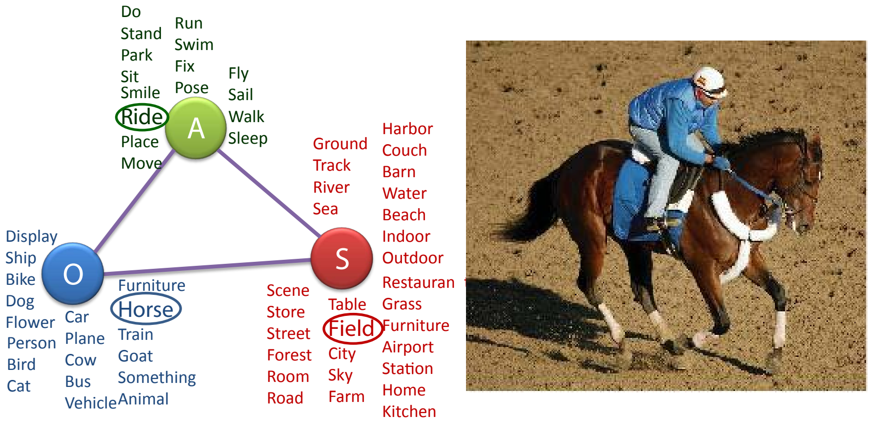


Fig. 2. We represent the space of the meanings by triplets of $\langle object, action, scene\rangle$. This is an MRF. Node potentials are computed by linear combination of scores from several detectors and classifiers. Edge potentials are estimated by frequencies. We have a reasonably sized state space for each of the nodes. The possible values for each nodes are written on the image. "O" stands for the node for the object, "A" for the action, and "S" for scene. Learning involves setting the weights on the node and edge potentials and inference is finding the best triplets given the potentials.

values. Choosing among them will result in a triplet. The mapping from images to meaning is reduced to learning to predict triplet for images. The problem of predicting a triplet from an image involves solving a (small) multi-label Markov random field. Each slot in the meaning representation can take a value from a set of discrete values. Figure 2 depicts the representation of the meaning space and the corresponding MRF. There is a node for objects which can take a value from a possible set of 23 nouns, a node for actions with 16 different values, and a node to scenes that can select each of 29 different values. The edges correspond to the binary relationships between nodes. Having provided the potentials of the MRF, we use a greedy method to do inference. Inference involves finding the best selection of the discrete sets of values given the unary and binary potentials.

We learn to predict triplets for images discriminatively. This requires having a dataset of images labeled with their meaning triplets. The potentials are computed as linear combinations of feature functions. This casts the problem of learning as searching for the best set of weights on the linear combination of feature functions so that the ground truth triplets score higher than any other triplet. Inference involves finding $argmax_y w^T \Phi(x, y)$ where Φ is the potential function, y is the triplet label, and w are the learned weights.

2.2 Image Potentials

We need informative features to drive the mapping from the image space to the meaning space.

Node Potentials: To provide information about the nodes on the MRF we first need to construct image features. Our *image features* consist of:

Felzenszwalb *et al.* detector responses: We use Felzenszwalb detectors [19] to predict confidence scores on all the images. We set the threshold such that all of the classes get predicted, at least once in each image. We then consider the max confidence of the detections for each category, the location of the center of the detected bounding box, the aspect ratio of the bounding box, and it's scale.

Hoiem *et al.* classification responses: We use the classification scores of Hoiem et. al [20] for the PASCAL classification tasks. These classifiers are based on geometry, HOG features, and detection responses.

Gist-based scene classification responses: We encode global information of images using gist [21]. Our features for scenes are the confidences of our Adaboost style classifier for scenes.

First we build node features by fitting a discriminative classifier (a linear SVM) to predict each of the nodes independently on the image features. Although the classifiers are being learned independently, they are well aware of other objects and scene information. We call these estimates *node features*. This is a number-of-nodes-dimensional vector and each element in this vector provides a score for a node given the image. This can be a node potential for object, action, and scene nodes. We expect similar images to have similar meanings, and so we obtain a set of features by matching our test image to training images. We combine these features into various other node potentials as below:

- by matching image features, we obtain the k-nearest neighbours in the training set to the test image, then compute the average of the node features over those neighbours, *computed from the image side*. By doing so, we have a representation of what the node features are for similar images.
- by matching image features, we obtain the k-nearest neighbours in the training set to the test image, then compute the average of the node features over those neighbours, *computed from the sentence side*. By doing so, we have a representation of what the sentence representation does for images that look like our image.
- by matching those node features derived from classifiers and detectors (above), we obtain the k-nearest neighbours in the training set to the test image, then compute the average of the node features over those neighbours, *computed from the image side*. By doing so, we have a representation of what the node features are for images that produce similar classifier and detector outputs.
- by matching those node features derived from classifiers and detectors (above), we obtain the k-nearest neighbours in the training set to the test image, then compute the average of the node features over those neighbours, *computed from the sentence side*. By doing so, we have a representation of what the sentence representation does for images that produce similar classifier and detector outputs.

Edge Potentials: Introducing a parameter for each edge results in unmanageable number of parameters. In addition, estimates of the parameters for the majority of edges would be noisy. There are serious smoothing issues. We adopt an approach similar to Good Turing smoothing methods to a) control the number of parameters b) do smoothing. We have multiple estimates for the edges potentials which can provide more accurate estimates if used together. We form the linear combinations of these potentials. Therefore, in learning we are interested in finding weights of the linear combination of the initial estimates so that the final linearly combined potentials provide values on the MRF so that the ground truth triplet is the highest scored triplet for all examples. This way we limit the number of parameters to the number of initial estimates.

We have four different estimates for edges. Our final score on the edges take the form of a linear combination of these estimates. Our four estimates for edges from node A to node B are:

- The normalized frequency of the word A in our corpus, $f(A)$.
- The normalized frequency of the word B in our corpus, $f(B)$.
- The normalized frequency of (A and B) at the same time, $f(A, b)$.
- $\frac{f(A,B)}{f(A)f(B)}$.

2.3 Sentence Potentials

We need a representation of the sentences. We represent a sentence by computing the similarity between the sentence and our triplets. For that we need to have a notion of similarity for objects, scenes and actions in text.

We used the Curran & Clark parser [22] to generate a dependency parse for each sentence. We extracted the subject, direct object, and any nmod dependencis involving a noun and a verb. These dependencies were used to generate the (object, action) pairs for the sentences. In order to extract the scene information from the sentences, we extracted the head nouns of the prepositional phrases (except for the prepositions “of” and “with”), and the head nouns of the phrase “X in the background”.

Lin Similarity Measure for Objects and Scenes We use the Lin similarity measure [23] to determine the semantic distance between two words. The Lin similarity measure uses WordNet synsets as the possible meanings of each words. The noun synsets are arranged in a heirarchy based on hypernym (is-a) and hyponym (instance-of) relations. Each synset is defined as having an information content based on how frequently the synset or a hyponym of the synset occurs in a corpus (in the case, SemCor). The similarity of two synsets is defined as twice the information content of the least common ancestor of the synsets divided by the sum of the information content of the two synsets. Similar synsets will have a LCA that covers the two synsets, and very little else. When we compared two nouns, we considered all pairs of a filtered list of synsets for each noun, and used the most similar synsets. We filtered the list of synsets for each noun by limiting it to the first four synsets that were at least 10% as frequent as the most common synset of that noun. We also required the synsets to be physical entities.

Action Co-occurrence Score We generated a second image caption data set consisting of roughly 8,000 images pulled from six Flickr groups. For all pairs of verbs, we used the likelihood ratio to determine if the two verbs co-occurring in the different captions of the same image was significant. We then used the likelihood ratio as the similarity score for the positively correlated verb pairs, and the negative of the likelihood ratio as the similarity score for the negatively correlated verb pairs. Typically, we found that this procedure discovered verbs that were either describing the same action or describing two actions that commonly co-occurred.

Node Potentials: We now can provide a similarity measure between sentences and objects, actions, and scenes using scores explained above. Below we explain our estimates of sentence node potentials.

- First we compute the similarity of each object, scene, and action extracted from each sentence. This gives us the the first estimates for the potentials over the nodes. We call this the *sentence node feature*.
- For each sentence, we also compute the average of sentence node features for other four sentences describing the same images in the train set.
- We compute the average of k nearest neighbors in the sentence node features space for a given sentence. We consider this as our third estimate for nodes.
- We also compute the average of the image node features for images corresponding to the nearest neighbors in the item above.
- The average of the sentence node features of reference sentences for the nearest neighbors in the item 3 is considered as our fifth estimate for nodes.
- We also include the sentence node feature for the reference sentence.

Edge Potentials: The edge estimates for sentences are identical to to edge estimates for the images explained in previous section.

2.4 Learning

There are two mappings that need to be learned. The map from the image space to the meaning space uses the image potentials and the map from the sentence space to the meaning space uses the sentence potentials. Learning the mapping from images to meaning involves finding the weights on the linear combinations of our image potentials on nodes and edges so that the ground truth triplets score highest among all other triplets for all examples. This is a structure learning problem [24] which takes the form of

$$\min_{w} \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i \in examples} \xi_i \tag{1}$$

subject to

$$w\Phi(x_i, y_i) + \xi_i \geq \max_{y \in \text{meaning space}} w\Phi(x_i, y) + L(y_i, y) \; \forall i \in \text{examples}$$

$$\xi_i \geq 0 \; \forall i \in \text{examples}$$

where λ is the tradeoff factor between the regularization and slack variables ξ, Φ is our feature functions, x_i corresponds to our i^{th} image, and y_i is our structured label for the i^{th} image. We use the stochastic subgradient descent method [25] to solve this minimization.

3 Evaluation

We emphasize quantitative evaluation in our work. Our vocabulary of meaning is significantly larger than the equivalent in [8, 9]. Evaluation requires innovation both in datasets and in measurement, described below.

3.1 Dataset

We need a dataset with images and corresponding sentences and also labels for our representations of the meaning space. No such dataset exists. We build our own dataset of images and sentences around the PASCAL 2008 images. This means we can use and compare to state of the art models and image annotations in PASCAL dataset.

PASCAL Sentence data set To generate the sentences, we started with the 2008 PASCAL development kit. We randomly selected 50 images belonging to each of the 20 categories. Once we had a set of 1000 images, we used Amazon's Mechanical Turk to generate five captions for each image. We required the annotators to be based in the US, and that they pass a qualification exam testing their ability to identify spelling errors, grammatical errors, and descriptive captions. More details about the methods of collection can be found in [26]. Our dataset has 5 sentences for each image of the thousand images resulting in 5000 sentences. We also manually add labels for triplets of $\langle objects, actions, scenes \rangle$ for each images. These triplets label the main object in the image, the main action, and the main place. There are 173 different triplets in our train set and 123 in test set. There are 80 triplets in the test set that appeared in the train set. The dataset is available at `http://vision.cs.uiuc.edu/pascal-sentences/`.

3.2 Inference

Our model is learned to maximize the sum of the scores along the path identified by a triplet. In inference we search for the triplet which gives us the best additive score, $argmax_y w^T \Phi(x_i, y)$. These models prefer triplets with combination of strong and poor responses over all mediocre responses. We conjecture that a multiplicative inference model would result in better predictions as the multiplicative model prefers all the responses to be reasonably good. Our multiplicative inference has the form of $argmax_y \prod w^T \Phi(x_i, y)$. We select the best triplet given the potentials on the nodes and edges greedily by relaxing an edge and solving for the best path and re-scoring the results using the relaxed edge.

3.3 Matching

Once we predict triplets for images and sentences we can score a match between an image and a sentence. If an image and a sentence predict very similar triplets, they should be projections of nearby points in the meaning space, and so they should have a high matching score. A natural score of the similarity of sentence triplets and image triples is the sum of ranks of sentence meaning and image meaning; the pair with smallest value of this sum is both strongly predicted by the image and strongly predicted by the sentence. However, this score is likely to be noisy, and is difficult to compute, because we must touch all pairs of meanings. We use a good, noise resistant approximation. To obtain the score, we:

- obtain the top k ranking triplets derived from sentences and compute the rank of each as an image triplet
- obtain the top k ranking triplets derived from images and compute the rank of each as a sentence triplet
- sum the sum of ranks for each of these sets, weighted by in the inverse rank of the triplet, so as to emphasize triplets that score strongly.

3.4 Out of Vocabulary Extension

We generate sentences by searching a pool of sentences for one that has a good match score to the image. We cannot learn a detector/classifier for each object/action/scene that exists. This means we need to score the similarity between the image and sentences that contain unfamiliar words. We propose using text information to attack this problem. For each unknown object we can produce a score of the similarity of that object with all of the objects in our vocabulary using distributional semantics methods explained in section 2.3 . We do the same thing for verbs and scenes as well. These similarity measures work as a crude guide to our model. For example, in Figure 6, we don't have a detector for "Volkswagen", "herd", "woman", and "cattle" but we can recognize them. our similarity measures provides a similarity distributions over things we know. This similarity distribution helps us to recognize objects, actions, and scenes for which we have no detector/classifier using objects/actions/scenes we know.

3.5 Experimental settings

We divide our 1000 images to 600 training images and 400 testing images. We use 15 nearest neighbors in building potentials for images and sentences. For matching we use 50 closest triplets.

3.6 Mapping to the Meaning Space

Table 1 compares the results of mapping the images to the meaning space, predicting triplets for images. To do that, we need a measure of comparisons between pairs of triplets, the one that we predict and the ground truth triplets. One way of doing this is by simple comparisons of triplets. A prediction is correct if all three elements agree and wrong otherwise. We could also measure if any of the elements in the triplet match. Each score is insensitive to important aspects of loss. For example, predicting $\langle cat, sit, mat\rangle$ when ground truth is $\langle dog, sit, ground\rangle$ is not as bad as predicting $\langle bike, ride, street\rangle$. This implies that the penalty for confusing cats with dogs should be smaller than that for confusing cats with bikes. The same argument holds for actions and scenes as well. We also need our measure to take into account the amount of information a prediction conveys. For example, predicting $\langle object, do, scene\rangle$ is less favorable than $\langle cat, sit, mat\rangle$.

Tree-F1 measure: Tree-F1 measure: We need a measure that reflects two important interacting components, accuracy and specificity. We believe the right way to score error is to use taxonomy trees. We have taxonomy trees for objects, actions, and scenes and we can use them to measure the accuracy, relevance, and specificity of predictions. We introduce a novel measure, Tree-F1, which reflects how accurate and specific the prediction is. Given a taxonomy tree for, say, objects objects, we represent each prediction by the path from the root of the taxonomy tree to the predicted node. For example, if the prediction is cat we represent it as $Objects \Rightarrow animal \Rightarrow cat$. We can then report the standard F1 measure using the precision and recall. Precision is defined as the total number of edges on the path that matches the edges on the ground truth path divided by the total number of edges on the ground truth path and recall as the total number of edges on the predicted path which is in the ground truth path divided by the total number of edges in the path. For example, the measure for predicting dog when the ground truth is cat is 0.5 where the precision is 0.5 and recall is 0.5, the measure for predicting animal when the ground truth is cat is 0.66, and it is 0 for predicting bike when the ground truth is cat. The same procedure is applied to actions and scenes. The Tree-F1 measure for a triple is the mean of the three measures for objects, actions, and scenes. Table 1 shows Tree-F1 measures for several different experimental settings.

BLUE Measure: Similar to Machine translation approaches where reports of accuracy involves scores for the correctness of the translation and the correctness of the generated translation in terms of language and logic, we also consider another measure to check if the triplet we generate is logically valid or not. Analogous to the BLEU score in machine translation literature we introduce the "BLUE" score which measures this. For example, $\langle bottle, walk, street\rangle$ is not valid. For that, we check if the triplet ever appeared in our corpus or not. Table 1 shows these scores for the triplets predicted by several different experimental settings.

4 Results

To evaluate our method we provide qualitative and quantitative results. There are two stages in our model. First we show the ability of our method to map

	Obj	No Edge	FW(A)	SL(A)	FW(M)	SL(M)
Mean Tree-F1 for first 5	0.44	**0.52**	0.38	0.45	0.47	0.51
Mean BLUE for first 5	0.24	0.27	0.16	0.58	**0.76**	0.74
Mean Tree-F1 for first 5 objects	**0.59**	0.58	0.36	0.53	0.55	0.57
Mean Tree-F1 for first 5 actions	0.27	**0.52**	0.50	0.37	0.42	0.47
Mean Tree-F1 for first 5 scenes	0.28	0.48	0.28	0.44	0.46	**0.48**

Table 1. Evaluation of mapping from the image space to the meaning space. “Obj” means when we only consider the potentials on the object node and use uniform potentials for other nodes and edges. “No Edge” means assuming a uniform potential over edges. “FW(A)” stands for fixed weights with additive inference model. This is the case where we use all the potentials but we don’t learn any weights for them. “SL(A)” means using structure learning with additive inference model. “FW(M)“ is similar to “FW(A)” with the exception that the inference model is multiplicative instead of additive. “SL(M)” is the structure learning with multiplicative inference.

from the image space to the meaning space. We then evaluate our results on predicting sentences for images, annotation. We also show qualitative results for finding images for sentences, illustration.

4.1 Mapping Images to Meanings

Table 1 compares several different experimental settings in terms of two measures explained above, Tree-F1 and BLUE. Each column in Table 1 corresponds to an experimental setting. We report average Tree-F1 and average BLUE measures for five top triplets for all images. We also breakdown the Tree-F1 to objects, actions, and scenes in bottom three rows of the table.

4.2 Annotation: Generating Sentences from Images

Figure 3 shows top 5 predicted triplets and top 5 generated sentences for example images in our test set. Quantitative evaluation of generated sentence is very challenging. We trained 2 individuals to annotate generated sentences. We ask them to annotate each generated sentence by either 1, 2, or 3. 1 means that the sentence is quite accurate with possible little mistakes about details in the sentence. 2 implies that the sentence have a rough idea about the image but it’s not very accurate and 3 means that the sentence is not even remotely close to the image. We generate 10 sentences for each image. The total average of the scores given by these individuals is 2.33. The average number of sentences with score one per image is 1.48. The average number of sentences with score 2 per image is 3.8. 208 of 400 images have at least one sentence with score 1. 354 sentences out of 400 images have at least one sentence with score 2.

4.3 Illustration: Finding images best described by sentences

Not only our model can provide sentences that describe an image, but it also can find images which are best described by a given sentence. Once the connections to the meaning space is established, one could go in both directions, from images to sentences or the other way around. Figure 4 shows examples of finding images for sentences. For more qualitative results please see the supplementary material.

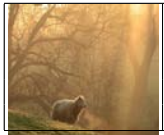
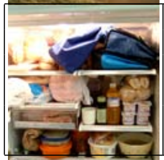
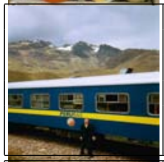
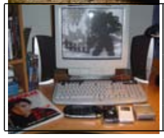
	(pet, sleep, ground) (dog, sleep, ground) (animal, sleep, ground) (animal, stand, ground) (goat, stand, ground)	see something unexpected. Cow in the grassfield. Beautiful scenery surrounds a fluffly sheep. Dog hearding sheep in open terrain. Cattle feeding at a trough.
	(furniture, place, furniture) (furniture, place, room) (furniture, place, home) (bottle, place, table) (display, place, table)	Refrigerator almost empty. Foods and utensils. Eatables in the refrigerator. The inside of a refrigerator apples, cottage cheese, tupperwares and lunch bags. Squash apenny white store with a hand statue, picnic tables in front of the building.
	(transportation, move, track) (bike, ride, track) (transportation, move, road) (pet, sleep, ground) (bike, ride, road)	A man stands next to a train on a cloudy day A backpacker stands beside a green train This is a picture of a man standing next to a green train There are two men standing on a rocky beach, smiling at the camera. This is a person laying down in the grass next to their bike in front of a strange white building.
	(display, place, table) (furniture, place, furniture) (furniture, place, furniture) (bottle, place, table) (furniture, place, home)	This is a lot of technology. Somebody's screensaver of a pumpkin A black laptop is connected to a black Dell monitor This is a dual monitor setup Old school Computer monitor with way to many stickers on it

Fig. 3. Generating sentences for images: We show top five predicted triplets in the middle column and top five predicted sentences in the right column.

4.4 Out of Vocabulary Extension

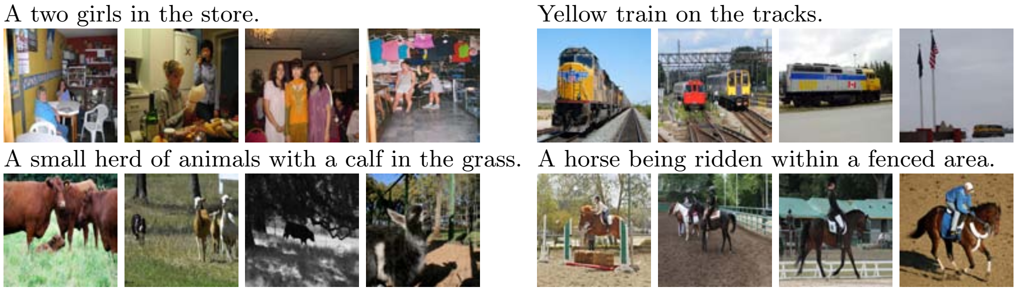
Figure 6 depicts examples of the cases where we could successfully recognize objects/actions for which we have no detector/classifier. This is very interesting as the intermediate meaning space allows us to benefit from distributional semantics. This means that we can learn to recognize unknown objects/actions/scenes by looking at the patterns of responses from other similar known detector/classifiers.

5 Discussion and Future Work

Sentences are rich, compact and subtle representations of information. Even so, we can predict good sentences for images that people like. The intermediate meaning representation is one key component in our model as it allows benefiting from distributional semantics. Our sentence model is oversimplified. We think an iterative procedure for going deeper in sentences and images would be the right direction. Once a sentence is generated for an image, it is much easier to check for adjectives and adverbs.

6 Aknowledgements

This work was supported in part by the National Science Foundation under IIS - 0803603 and in part by the Office of Naval Research under N00014-01-1-0890 as part of the MURI program, in part by a gift from Google. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do



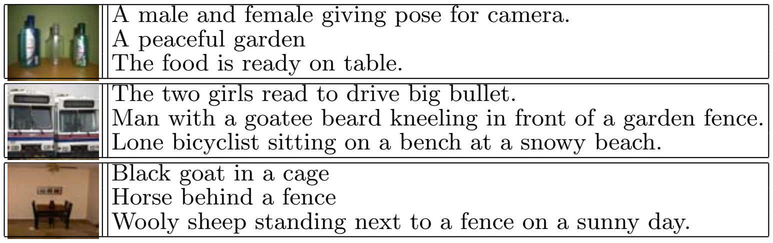
A two girls in the store.

Yellow train on the tracks.

A small herd of animals with a calf in the grass.

A horse being ridden within a fenced area.

Fig. 4. Finding images for sentences: Once the matching in the meaning space is established we can generate sentences for images (annotation) and also find images that can be best describe by a sentence. In this picture we show four sentences with four 144 highest ranked images. We provide a list of 10 highest score images for each sentence for the test set in the supplementary material.



A male and female giving pose for camera.
A peaceful garden
The food is ready on table.

The two girls read to drive big bullet.
Man with a goatee beard kneeling in front of a garden fence.
Lone bicyclist sitting on a bench at a snowy beach.

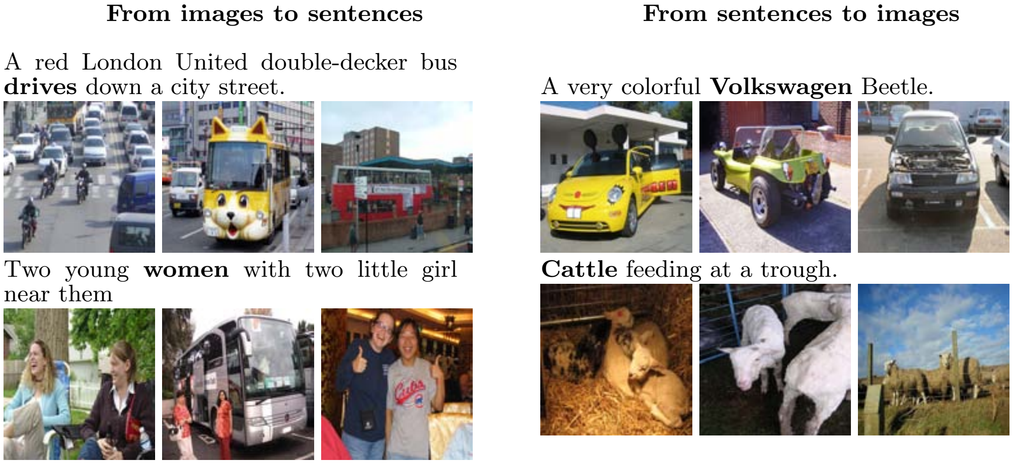
Black goat in a cage
Horse behind a fence
Wooly sheep standing next to a fence on a sunny day.

Fig. 5. Examples of failures in generating sentences for images.

not necessarily reflect those of the National Science Foundation or the Office of Naval Research. Ali Farhadi was supported by the Google PhD fellowship. We also would like to thank Majid Ashtiani for his help on cluster computing, and Hadi Kiapour, Attiye Hosseini for their help on evaluation.

References

1. Barnard, K., Duygulu, P., Forsyth, D.: Clustering art. In: CVPR. (2001) II:434–441
2. Mori, Y., Takahashi, H., Oka, R.: Image-to-word transformation based on dividing and vector quantizing images with words. In: WMISR. (1999)
3. Duygulu, P., Barnard, K., de Freitas, N., Forsyth, D.: Object recognition as machine translation. In: ECCV. (2002) IV: 97–112
4. Datta, R., Li, J., Wang, J.Z.: Content-based image retrieval: approaches and trends of the new age. In: MIR '05. (2005) 253–262
5. Forsyth, D., Berg, T., Alm, C., Farhadi, A., Hockenmaier, J., Loeff, N., Wang, G.: Words and pictures: Categories, modifiers, depiction and iconography. In: Object Categorization: Computer and Human Vision Perspectives, CUP (2009)
6. Phillips, P.J., Newton, E.: Meta-analysis of face recognition algorithms. In: ICAFGR. (2002)
7. Gupta, A., Davis, L.: Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In: ECCV. (2008)
8. Li, L.J., Fei-Fei, L.: What, where and who? classifying event by scene and object recognition. In: ICCV. (2007)
9. Li, L.J., Socher, R., Fei-Fei, L.: Towards total scene understanding:classification, annotation and segmentation in an automatic framework. In: CVPR. (2009)
10. Gupta, A., Davis, L.: Objects in action: An approach for combining action understanding and object perception. In: CVPR. (2007)



From images to sentences

A red London United double-decker bus **drives** down a city street.

Two young **women** with two little girl near them

From sentences to images

A very colorful **Volkswagen** Beetle.

Cattle feeding at a trough.

Fig. 6. Out of vocabulary extension: We don't have detectors for "drives", "women", "Volkswagen", and "Cattle". Despite this fact, we could recognize these objects/actions. Distributional semantics provide us with the ability to model unknown objects/actions/categories with their similarities to known categories. Here we show examples of sentences and images when we could recognize these unknowns for both generating sentences from images and finding images for sentences.

11. A. Gupta, A.K., Davis, L.: Observing human-object interactions: Using spatial and functional compatibility for recognition. In: Trans on PAMI. (2009)
12. Yao, B., Fei-Fei, L.: Modeling mutual context of object and human pose in human-object interaction activities. In: CVPR. (2010)
13. Berg, T.L., Berg, A.C., Edwards, J., Forsyth, D.A.: Who's in the picture. In: Advances in Neural Information Processing. (2004)
14. Mensink, T., Verbeek, J.: Improving people search using query expansions: How friends help to find people. In: ECCV. (2008)
15. Luo, J., Caputo, B., Ferrari, V.: Who's doing what: Joint modeling of names and verbs for simultaneous face and pose annotation. In: NIPS. (2009)
16. Coyne, B., Sproat, R.: Wordseye: an automatic text-to-scene conversion system. In: SIGGRAPH '01. (2001)
17. Gupta, A., Srinivasan, P., Shi, J., Davis, L.: Understanding videos, constructing plots: Learning a visually grounded storyline model from annotated videos. In: CVPR. (2009)
18. Yao, B.Z., Yang, X., Lin, L., Lee, M.W., Zhu, S.C.: I2t: Image parsing to text description. Proc. IEEE (2010) In Press.
19. Felzenszwalb, P., Mcallester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. CVPR 2008 (2008)
20. Hoiem, D., Divvala, S., Hays, J.: Pascal voc 2009 challenge. In: PASCAL challenge workshop in ECCV. (2009)
21. Oliva, A., Torralba, A.: Building the gist of a scene: the role of global image features in recognition. In: Progress in Brain Research. (2006) 2006
22. Curran, J., Clark, S., Bos, J.: Linguistically motivated large-scale nlp with c&c and boxer. (In: ACL) 33–36
23. Lin, D.: An information-theoretic definition of similarity. In: ICML. (1998) 296–304
24. Taskar, B., Chatalbashev, V., Koller, D., Guestrin, C.: Learning structured prediction models: a large margin approach. In: ICML. (2005) 896–903
25. Ratliff, N., Bagnell, J.A., Zinkevich, M.: Subgradient methods for maximum margin structured learning. In: ICML. (2006)
26. Rashtchian, C., Young, P., Hodosh, M., Hockenmaier, J.: Collecting image annotations using amazon's mechanical turk. (In: NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk)