Baby Talk: Understanding and Generating Image Descriptions

Girish Kulkarni Visruth Premraj Sagnik Dhar Siming Li
Yejin Choi Alexander C Berg Tamara L Berg

Stony Brook University
Stony Brook University, NY 11794, USA
{tlberg}@cs.stonybrook.edu

Abstract

We posit that visually descriptive language offers computer vision researchers both information about the world, and information about how people describe the world. The potential benefit from this source is made more significant due to the enormous amount of language data easily available today. We present a system to automatically generate natural language descriptions from images that exploits both statistics gleaned from parsing large quantities of text data and recognition algorithms from computer vision. The system is very effective at producing relevant sentences for images. It also generates descriptions that are notably more true to the specific image content than previous work.

1. Introduction

People communicate using language, whether spoken, written, or typed. A significant amount of this language describes the world around us, especially the visual world in an environment or depicted in images or video. Such visually descriptive language is potentially a rich source of 1) information about the world, especially the visual world, and 2) training data for how people construct natural language to describe imagery. This paper exploits both of these lines of attack to build an effective system for automatically generating natural language – sentences – from images.

It is subtle, but several factors distinguish the task of taking images as input and generating sentences from tasks in many current computer vision efforts on object and scene recognition. As examples, when forming descriptive language, people go beyond specifying what objects are present in an image – this is true even for very low resolution images [23] and for very brief exposure to images [11]. In both these settings, and in language in general, people include specific information describing not only scenes, but specific objects, their relative locations, and modifiers adding additional information about objects.

Figure 1. Our system automatically generates the following descriptive text for this example image: *"This picture shows one person, one grass, one chair, and one potted plant. The person is near the green grass, and in the chair. The green grass is by the chair, and near the potted plant."*

Mining the absolutely enormous amounts of visually descriptive text available in special library collections and on the web in general, make it possible to discover statistical models for what modifiers people use to describe objects, and what prepositional phrases are used to describe relationships between objects. These can be used to select and train computer vision algorithms to recognize constructs in images. The output of the computer vision processing can be "smoothed" using language statistics and then combined with language models in a natural language generation process.

Natural language generation constitutes one of the fundamental research problems in natural language processing (NLP) and is core to a wide range of NLP applications such as machine translation, summarization, dialogue systems, and machine-assisted revision. Despite substantial advancement within the last decade, natural language generation still remains an open research problem. Most previous work in NLP on automatically generating captions or descriptions for images is based on retrieval and summarization. For instance, [1] relies on GPS meta data to access relevant text documents and [13] assume relevant

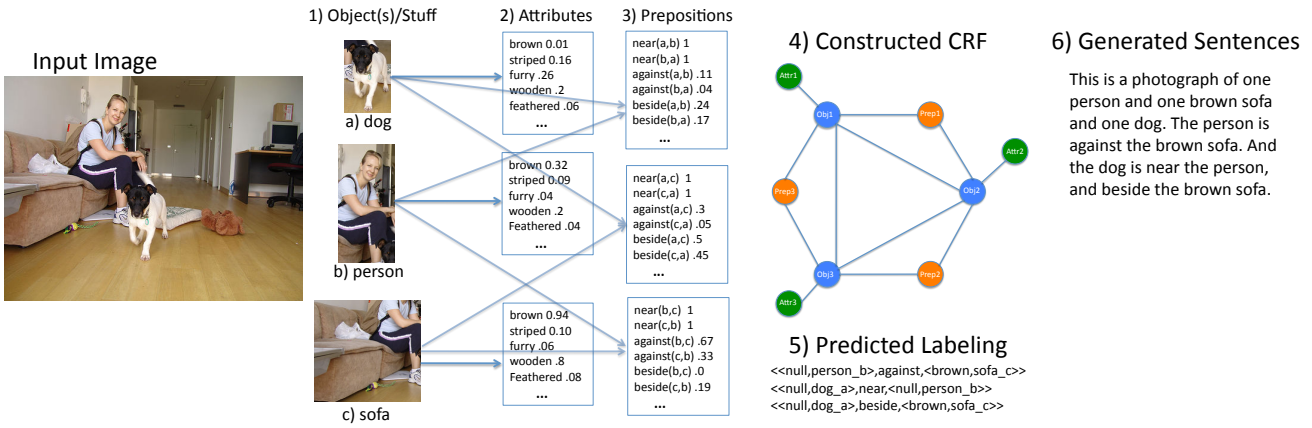

Figure 2. System flow for an example image: 1) object and stuff detectors find candidate objects, 2) each candidate region is processed by a set of attribute classifiers, 3) each pair of candidate regions is processed by prepositional relationship functions, 4) A CRF is constructed that incorporates the unary image potentials computed by 1-3, and higher order text based potentials computed from large document corpora, 5) A labeling of the graph is predicted, 6) Sentences are generated based on the labeling.

documents are provided. The process of generation then becomes one of combining or summarizing relevant documents, in some cases driven by keywords estimated from the image content [13]. From the computer vision perspective these techniques might be analogous to first recognizing the scene shown in an image, and then *retrieving* a sentence based on the scene type. It is very unlikely that a retrieved sentence would be as descriptive of a particular image as the *generated* sentence in Fig. 1.

This paper pushes to make a tight connection between the particular image content and the sentence generation process. This is accomplished by detecting objects, modifiers (adjectives), and spatial relationships (prepositions), in an image, smoothing these detections with respect to a statistical prior obtained from descriptive text, and then using the smoothed results as constraints for sentence generation. Sentence generation is performed either using a n-gram language model [3, 22] or a simple template based approach [27, 4]. Overall, our approach can handle the potentially huge number of scenes that can be constructed by composing even a relatively small number of instances of several classes of objects in a variety of spatial relationships. Even for quite small numbers for each factor, the total number of such layouts is not possible to sample completely, and any set of images would have some particular bias. In order to avoid evaluating such a bias, we purposefully avoid whole image features or scene/context recognition in our evaluation – although noting explicitly that it would be straightforward to include a scene node and appropriate potential functions in the model presented.

2. Related Work

Early work on connecting words and pictures for the purpose of automatic annotation and auto illustration focused on associating individual words with image regions [2, 8]. In continuations of that work, and other work on image parsing and object detection, the spatial relationships between labeled parts – either detections or regions – of images was used to improve labeling accuracy, but the spatial relationships themselves were not considered outputs in their own right [24, 7, 16, 21, 15]. Estimates of spatial relationships between objects form an important part of the output of the computer vision aspect of our approach and are used to drive sentence generation.

There is a great deal of ongoing research on estimating attributes for use in computer vision [18, 9, 19, 14] that maps well to our process of estimating modifiers for objects in images. We use low level features from Farhadi *et al*. [9] for modifier estimation. Our work combines priors for visually descriptive language with estimates of the modifiers based on image regions around object detections.

There is some recent work *very close in spirit* to our own. Yao *et al*. [26] look at the problem of generating text with a comprehensive system built on various hierarchical knowledge ontologies and using a human in the loop for hierarchical image parsing (except in specialized circumstances). In contrast, our work automatically mines knowledge about textual representation, and parses images fully automatically – without a human operator – and with a much simpler approach overall. Despite the simplicity of our framework it is still a step toward more complex description generation compared to Farhadi *et al*.'s (also fully automatic) method based on parsing images into a meaning representation "triple" describing 1 object, 1 action, and 1 scene [10]. In their work, they use a single triple estimated for an image to retrieve sentences from a collection written to describe similar images. In contrast our work detects multiple

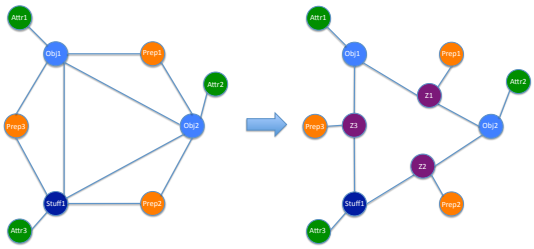

Figure 3. CRF for an example image with 2 object detections and 1 stuff detection. Left shows original CRF with trinary potentials. Right shows CRF reduced to pairwise potentials by introducing z variables with domains covering all possible triples of the original 3-clique.

objects, modifiers, and their spatial relationships, and *generates* sentences to fit these constituent parts, as opposed to *retrieving* sentences whole.

3. Method Overview

An overview of our system is presented in figure 2. For an input image: **1)** Detectors are used to detect things (*e.g.* bird, bus, car, person, etc.) and stuff (*e.g.* grass, trees, water, road, etc.). We will refer to these as *objects*, and *stuff*, or collectively as objects. **2)** each candidate object (either thing or stuff) region is processed by a set of attribute classifiers, **3)** each pair of candidate regions is processed by prepositional relationship functions, **4)** A CRF is constructed that incorporates the unary image potentials computed by 1-3, with higher order text based potentials computed from large text corpora, **5)** A labeling of the graph is predicted, and **6)** Sentences are generated based on the labeling.

The rest of the paper describes the Conditional Random Field used to predict a labeling for an input image (Sec. 4), then the image based potentials (Sec. 5.1), and higher order text based potentials (Sec. 5.2). Sentence generation is covered in (Sec. 6) and evaluation in (Sec. 7).

4. CRF Labeling

We use a conditional random field (CRF) to predict the best labeling for an image (*e.g.* fig 3). Nodes of the CRF correspond to several kinds of image content: a) Objects - things or stuff, b) attributes which modify the appearance of an object, and c) prepositions which refer to spatial relationships between pairs of objects.

For a query image, we run a large set of (thing) object detectors across the image and collect the set of high scoring detections. We merge detections that are highly overlapping (greater than 0.3 intersection/union) into groups and create an object node for each group. In this way we avoid predicting two different object labels for the same region of an image which can occur when two different object detectors fire on the same object. We also run our stuff detectors across the image and create nodes for stuff categories with high scoring detections. Note that this means that the number of nodes in a graph constructed for an image depends on the number of object and stuff detections that fired in that image (something we have to correct for during parameter learning). For each object and stuff node we classify the appearance using a set of trained attribute classifiers and create a modifier node. Finally, we create a preposition node for each pair of object and stuff detections. This node predicts the probability of a set of prepositional relationships based on the spatial relationship between two object regions.

The domain (of possible labels) for each node is node dependent. For an object (or stuff) node the domain corresponds to the set of object (or stuff) detectors that fired at that region in the image. For the attribute nodes the domain corresponds to a set of appearance attributes that can modify the visual characteristics of an object (e.g. green or furry). For the preposition nodes the domain corresponds to a set of prepositional relations (e.g. on, under, near) that can occur between two objects.

We will minimize an energy function over labelings, L, of an image, I,

$$E(L;I,T) = -\sum_{i \in objs} F_i - \frac{2}{N-1} \sum_{ij \in objPairs} G_{ij}, \quad (1)$$

where T is a text prior, and N is the number of objects so $2/(N-1)$ normalizes – for variable number of node graphs – the contribution from object pair terms so that they contribute equally with the single object terms to the energy function. Here:

$$F_i = \alpha_0\beta_0\psi(obj_i; objDet) + \alpha_0\beta_1\psi(attr_i; attrCl) \quad (2)$$
$$+\alpha_1\gamma_0\psi(attr_i, obj_i; textPr) \quad (3)$$
$$G_{ij} = \alpha_0\beta_2\psi(prep_{ij}; prepFuns) \quad (4)$$
$$+\alpha_1\gamma_1\psi(obj_i, prep_{ij}, obj_j; textPr) \quad (5)$$

The three unary potential functions are computed from image based models and refer to: the detector scores for object(s) proposed by our trained object and stuff detectors ($\psi(obj_i; objDets)$), the attribute classification scores for an object (or stuff) region as predicted by our trained attribute classifiers ($\psi(attr_i; attrCl)$), and the prepositional relationship score computed between pairs of detection regions ($\psi(prep_{ij}; prepFuns)$). Descriptions of the particular detectors, classifiers and functions used are provided in Sec. 5.1.

The pairwise ($\psi(mod_i, obj_i; textPr)$) and trinary ($\psi(obj_i, prep_{ij}, obj_j; textPr)$) potential functions model the pairwise scores between object and attribute node labels, and the trinary scores for an object-preposition-object triple labeling respectively. These higher order potentials could be learned from a large pool of labeled image data. However, for a reasonable number of objects, and prepositions

the amount of labeled image data that would be required is daunting. Instead we learn these from large text collections. By observing in text how people describe objects, attributes and prepositions between objects we can model the relationships between node labels. Descriptions of the text based potentials are provided in Sec. 5.2.

4.1. Converting to Pairwise potentials

Since preposition nodes describe the relationship between a preposition label and two object labels, they are most naturally modeled through trinary potential functions:

$$\psi(obj_i, prep_{ij}, obj_j; textPr) \tag{6}$$

However, most CRF inference code accepts only unary and pairwise potentials. Therefore we convert this trinary potential into a set of unary and pairwise potentials through the introduction of an additional z node for each 3-clique of obj-prep-obj nodes (see fig 3). Each z node connecting two object nodes has domain O1×P×O2 where O1 is the domain of object node1, P is our set of prepositional relations, and O2 is the domain of object node2. In this way the trinary potential is converted to a unary potential on z, $\psi(z_{ij}; textPr)$, along with 3 pairwise potentials, one for each of object node1, preposition node, and object node2 that enforce that the labels selected for each node are the same as the label selected for Z:

$$\psi(z_{ij}, obj_i) = \begin{cases} 0 & \text{if } Z_{ij}(1) = O_i \\ -inf & \text{o.w.} \end{cases} \tag{7}$$

$$\psi(z_{ij}, prep_{ij}) = \begin{cases} 0 & \text{if } Z_{ij}(2) = P_{ij} \\ -inf & \text{o.w.} \end{cases} \tag{8}$$

$$\psi(z_{ij}, obj_j) = \begin{cases} 0 & \text{if } Z_{ij}(3) = O_j \\ -inf & \text{o.w.} \end{cases} \tag{9}$$

4.2. CRF Learning

We take a factored learning approach to estimate the parameters of our CRF from 100 hand-labeled images. In our energy function (Eqns (1)-(5)), the α parameters represent the trade-off between image and text based potentials. The β parameters represent the weighting between image based potentials. And, the γ parameters represent the weighting between text based potentials. In the first stage of learning we estimate the image parameters β while ignoring the text based terms (by setting α_1 to 0). To learn image potential weights we fix β_0 to 1 and use grid search to find optimal values for β_1 and β_2. Next we fix the β parameters to their estimated value and learn the remaining parameters – the trade-off between image and text based potentials (α parameters) and the weights for the text based potentials (γ parameters). Here we set α_0 and γ_0 to 1 and use grid search over values of α_1 and γ_1 to find appropriate values.

It is important to carefully score output labelings fairly for graphs with variable numbers of nodes (dependent on the number of object detections for an image). We use a scoring function that is graph size independent:

$$\frac{obj_{t-f}}{N} + \frac{(mod, obj)_{t-f}}{N} + \frac{2}{N-1}\frac{(obj, prep, obj)_{t-f}}{N}$$

measuring the score of a predicted labeling as: a) the number of true obj labels minus the number of false obj labels normalized by the number of objects, plus b) the number of true mod-obj label pairs minus the number of false mod-obj pairs, plus c) the number of true obj-prep-obj triples minus the number of false obj-prep-obj triples normalized by the number of nodes and the number of pairs of objects (N choose 2).

4.3. CRF Inference

To predict the best labeling for an input image graph (both at test time or during parameter training) we utilize the sequential tree re-weighted message passing (TRW-S) algorithm introduced by Kolmogorov [17] which improves upon the original TRW algorithm from Wainwright et al [25]. These algorithms are inspired by the problem of maximizing a lower bound on the energy. TRW-S modifies the TRW algorithm so that the value of the bound is guaranteed not to decrease. For our image graphs, the CRF constructed is relatively small (on the order of 10s of nodes). Thus, the inference process is quite fast, taking on average less than a second to run per image.

5. Potential Functions

In this section, we present our image based and descriptive language based potential functions. At a high level the image potentials come from hand designed detection strategies optimized on external training sets . In contrast the text potentials are based on text statistics collected automatically from various corpora.

5.1. Image Based Potentials

$\psi(obj_i; objDet)$ - *Object and Stuff Potential*

Object Detectors: We use an object detection system based on Felzenszwalb *et al.*'s mixtures of multi-scale deformable part models [12] to detect "thing objects". We use the provided detectors for the 20 PASCAL 2010 object categories and train 4 additional non-PASCAL object categories for *flower, laptop, tiger*, and *window*. For the non-PASCAL categories, we train new object detectors using images and bounding box data from Imagenet [6]. The output score of the detectors are used as potentials.

Stuff Detectors: Classifiers are trained to detect regions corresponding to non-part based object categories. We train linear SVMs on the low level region features of [9] to recognize: sky, road, building, tree, water, and grass stuff categories. SVM outputs are mapped to probabilities. Training images and bounding box regions are taken from ImageNet. At test time, classifiers are evaluated on a coarsely

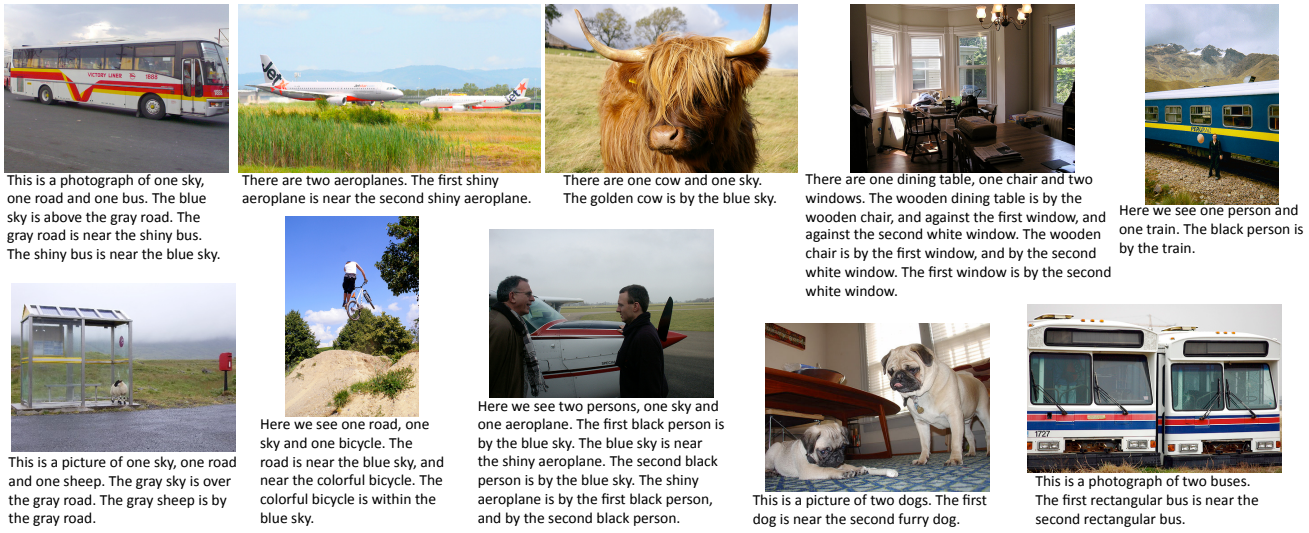

Figure 4. Results of sentence generation using our method with template based sentence generation. These are "*good*" results as judged by human annotators.

sampled grid of overlapping square regions covering the images. Pixels in any region with a classification probability above a fixed threshold are treated as detections, and the max probability for a region is used as the potential value.

$\psi(attr_i; attrCl)$ - *Attribute Potential*

Attribute Classifiers: We train visual attribute classifiers that are relevant for our object (and stuff) categories. Therefore, we mine our large text corpus of Flickr descriptions (described in Sec. 5.2) to find attribute terms commonly used with each object (and stuff) category removing obviously non-visual terms. The resulting list consists of 21 visual attribute terms describing color (e.g. blue, gray), texture (e.g. striped, furry), material (e.g. wooden, feathered), general appearance (e.g. rusty, dirty, shiny), and shape (e.g. rectangular) characteristics. Training images for the attribute classifiers come from Flickr, Google, the attribute dataset provided by Farhadi et al [9], and ImageNet [6]. An RBF kernel SVM is used to learn a classifier for each visual attribute term (up to 150 positive peer class with all other training examples as negatives). The outputs of the classifiers are used as potential values.

$\psi(prep_{ij}; prepFuns)$ - *Preposition Potential*

Preposition Functions: We design simple prepositional functions that evaluate the spatial relationships between pairs of regions in an image and provide a score for each of 16 preposition terms (e.g. above, under, against, beneath, in, on etc). For example, the score for $above(a, b)$ is computed as the percentage of $region_a$ that lies in the image rectangle above the bounding box around $region_b$. The potential for $near(a, b)$ is computed as the minimum distance between $region_a$ and $region_b$ divided by the diagonal size of a bounding box around $region_a$. Similar functions are used for the other preposition terms. We include synonymous prepositions to encourage variation in sentence generation but sets of synonymous prepositions share the same potential. Note for each preposition we compute both prep(a,b) and prep(b,a) as either labeling order can be predicted in the output result.

5.2. Text Based Potentials

We use two potential functions calculated from large text corpora. The first is a pairwise potential on attribute-object label pairs $\psi(attr_i, obj_i; textPr)$ and the second is a trinary potential on object-preposition-object triples $\psi(obj_i, prep_{ij}, obj_j; textPr)$. These potentials are the probability of various attributes for each object (given the object) and the probabilities of particular prepositional relationships between object pairs (given the pair of objects). The conditional probabilities are computed from counts of word co-occurrence as described below.

Parsing Potentials: To generate counts for the attribute-object potential $\psi_p(attr_i, obj_i; textPr)$ we collect a large set of Flickr image descriptions (similar to but less regulated than captions). For each object (or stuff) category we collect up to the min of 50000 or all image descriptions by querying the Flickr API[1] with each object category term. Each sentence from this descriptions set is parsed by the Stanford dependency parser [5] to generate the parse tree and dependency list for the sentence. We then collect statistics about the occurence of each attribute and object pair using the adjectival modifier dependency $amod(attribute, object)$. Counts for synonyms of object and attribute terms are merged together.

For generating the object-preposition-object potential

[1] http://www.flickr.com/services/api/

Templated Generation: This is a photograph of one furry sheep.
Simple Decoding: the furry sheep it.

Templated Generation: Here we see three persons, one sky, one grass and one train. The first colorful person is underneath the clear sky, and beside the second colorful person, and within the shiny train. The second colorful person is underneath the clear sky, and by the shiny train. The green grass is near the clear sky. The third black person is underneath the clear sky, and by the green grass, and within the shiny train. The shiny train is by the clear sky, and beside the green grass.
Simple Decoding: the colorful person is underneath the clear sky. the colorful person who beside the colorful person. the colorful person is underneath the clear sky. the green grass and near the clear sky. the colorful person is within the shiny train. the black person is underneath the clear sky. the black person and by the green grass. the shiny train and by the clear sky. the colorful person and by the shiny train. the shiny train and beside the green grass. the black person is within the shiny train.

Templated Generation: Here we see two cows and one tree. The first cow is by the tree. The second cow is by the tree.
Simple Decoding: the cow and by the tree. the cow and by the tree.

Figure 5. Comparison of our two generation methods.

$\psi_p(obj_i, prep_{ij}, obj_i; textPr)$ we collect ∼1.4 million Flickr image descriptions by querying for pairs of object terms. Sentences containing at least 2 object (or stuff) categories and a prepositional (∼140k) are parsed using the Stanford dependency parser. We then collect statistics for the occurence of each prepositional dependency between object categories. For a prepositional dependency occurence, object1 is automatically picked as either the subject or object part of the prepositional dependency based on the voice (active or passive) of the sentence, while object2 is selected as the other. Counts include synonyms.

Google Potentials: Though we parse thousands of descriptions, the counts for some objects can be too sparse. Therefore, we also collect additional Google Search based potentials: $\psi_g(attr_i, obj_i; textPr)$ and $\psi_g(obj_i, prep_{ij}, obj_j; textPr)$. These potentials are computed from the number of search results approximated by Google for an exact string match query on each of our attribute-object pairs (e.g. "brown dog") and preposition-object-preposition triples (e.g. "dog on grass").

Smoothed Potentials: Our final potentials are computed as a smoothed combination of the parsing based potentials with the Google potentials: $\alpha\psi_p + (1-\alpha)\psi_g$.

6. Generation

The output of our CRF is a predicted labeling of the image. This labeling encodes three kinds of information: objects present in the image (nouns), visual attributes of those objects (modifiers), and spatial relationships between objects (prepositions). Therefore, it is natural to extract this meaning into a triple (or set of triples), *e.g.*:

$$<< white, cloud >, in, < blue, sky >>$$

Based on this triple, we want to generate a complete sentence such as *"There is a white cloud in the blue sky"*. We restrictions generation so that: the set of words in the meaning representation is fixed and generation must make use of all given content words; and, generation may insert only gluing words (i.e., *function words* such as "there", "is", "the", etc). These restrictions could be lifted in future work.

6.1. Decoding using Language Models

A N-gram language model is a conditional probability distribution $P(x_i|x_{i-N+1}, ..., x_{i-1})$ of N-word sequences $(x_{i-N+1}, ..., x_i)$, such that the prediction of the next word depends only on the previous N-1 words. That is, with N-1'th order Markov assumption, $P(x_i|x_1, ..., x_{i-1}) = P(x_i|x_{i-N+1}, ..., x_{i-1})$. Language models are shown to be simple but effective for improving machine translation and automatic grammar corrections.

In this work, we make use of language models to predict gluing words (i.e. *function words*) that put together words in the meaning representation. As a simple example, suppose we want to determine whether to insert a function word x between a pair of words α and β in the meaning representation. Then, we need to compare the length-normalized probability $\hat{p}(\alpha x \beta)$ with $\hat{p}(\alpha\beta)$, where $\hat{p}$ takes the n'th root of the probability p for n-word sequences, and $p(\alpha x \beta) = p(\alpha)p(x|\alpha)p(\beta|x)$ using bigram (2-gram) language models. If considering more than two function words between α and β, dynamic programming can be used to find the optimal sequence of function words efficiently. Because the ordering of words in each triple of the meaning representation coincides with the typical ordering of words in English, we retain the original ordering for simplicity. Note that this approach composes a separate sentence for each triple, independently from all other triples.

6.2. Templates with Linguistic Constraints

Decoding based on language models is a statistically principled approach, however, two main limitations are: (1) it is difficult to enforce grammatically correct sentences using language models alone (2) it is ignorant of discourse structure (coherency among sentences), as each sentence is generated independently. We address these limitations by constructing templates with linguistically motivated constraints. This approach is based on the assumption that there are a handful of salient syntactic patterns in descriptive language that we can encode as templates.

7. Experimental Results & Conclusion

To construct the training corpus for language models, we crawled Wikipedia pages that describe objects our system can recognize. For evaluation, we use the UIUC PASCAL sentence dataset[2], which contains up to five human-generated sentences that describe 1000 images. From this set we evaluate results on 847 images[3].

[2] http://vision.cs.uiuc.edu/pascal-sentences/
[3] 153 were used to learn CRF and detection parameters.

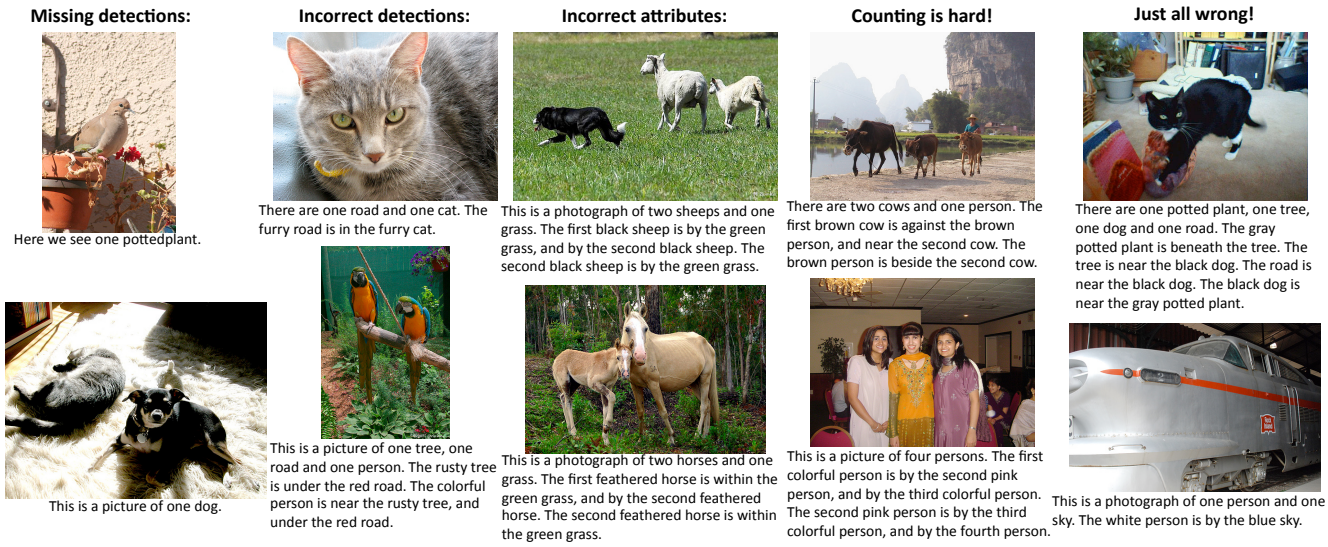

Figure 6. Results of sentence generation using our method with template based sentence generation. These are "*bad*" results as judged by human annotators.

Method	w/o	w/ synonym
Human	0.50	0.51
Language model-based generation	0.25	0.30
Template-based generation	0.15	0.18
Meaning representation (triples)	0.20	0.30

Table 1. Automatic Evaluation: BLEU score measured at 1

Method	Score
Quality of image parsing	2.85
Language model-based generation	2.77
Template-based generation	3.49

Table 2. Human Evaluation: possible scores are 4 (perfect without error), 3 (good with some errors), 2 (many errors), 1 (failure)

Method	k=1	k=2	k=3	k=4+
Quality of image parsing	2.90	2.78	2.82	3.33
Language model-based	2.27	3.00	2.76	2.95
Template-based generation	3.83	3.50	3.43	3.61

Table 3. Human Evaluation: k refers to the number of objects detected by CRF. Possible scores are 4 (perfect without error), 3 (good with some errors), 2 (many errors), 1 (failure)

Automatic Evaluation: BLEU [20] is a widely used metric for automatic evaluation of machine translation that measures the n-gram precision of machine generated sentences with respect to human generated sentences. Because our task can be viewed as machine translation from images to text, BLEU may seem like a reasonable choice at first glance. Upon a close look however, one can see that there is inherently larger variability in generating sentences from images than translating a sentence from one language to another. For instance, from the image shown in Figure 1, our system correctly recognizes objects such as "chair", "green grass", "potted plant", none of which is mentioned in the human generated description available in the UIUC PASCAL sentence dataset. As a result, BLEU will inevitably penalize many correctly generated sentences. Nevertheless, we report BLEU score as a standard evaluation method, and quantify its shortcomings for future research.

The first column in Table 1 shows BLEU score when measured with exact match for each word, and the second shows BLEU when we give full credits for synonyms. For context, we also compute the BLEU score between human-generated sentences; we average the BLEU score between each human-generated sentence to the set of others over all images. Finally, we compute BLEU score of the CRF outputs with respect to the human-generated sentences.

Human Evaluation: Evaluation by BLEU score facilitates efficient comparisons among different approaches, but does not measure vision output quality directly, and is oblivious to correctness of grammar or discourse quality (coherency across sentences). To directly quantify these aspects, we perform human judgment on the entire test set. The results are shown in Table 2 and 3, where the image parsing score evaluates how well we describe image content (the triples output by the CRF), and the other two scores evaluate the overall sentence quality. Overall our template generation method demonstrates a very high average human evaluation score of 3.49 (max 4) for the quality of generated sentences. We also do well at predicting image content (ave 2.85).

Note that human judgment of the generation quality does not correlate with BLEU score. Per BLEU, it looks as though language-model generation performs better than template-based one, but human judgment reveals the opposite is true. The Pearson's correlation coefficient between BLEU and human evaluationare is -0.17 and 0.05 for language model and template-based methods respectively. We also measure human annotation agreement on 160 instances. The scores given by two evaluators were identical on 61% of the instances, and close (difference ≤ 1) on 92%.

7.1. Qualitative Results

The majority of our generated sentences look quite good. Example results on PASCAL images rated as "good" are shown in fig 4. In fact most of our results look quite good. Even "bad" results almost always look reasonable and are relevant to the image content (fig 6). Only for a small minority of the images are the generated descriptions completely unrelated to the image content (fig 6, 2 right most images). In cases where the generated sentence is not quite perfect this is usually due to one of three problems: a failed object detection that misses an object, a detection that proposes the wrong object category, or an incorrect attribute prediction. However, because of our use of powerful vision systems (state of the art detectors and attribute methodologies) the results produced are often astonishingly good.

7.2. Conclusion

We have demonstrated a surprisingly effective, fully automatic, system that generates natural language descriptions for images. The system works well and can produce results much more specific to the image content than previous automated methods. Human evaluation validates the quality of the generated sentences. One key to the success of our system was automatically mining and parsing large text collections to obtain statistical models for visually descriptive language. The other is taking advantage of state of the art vision systems and combining all of these in a CRF to produce input for language generation methods.

Acknowledgements

This work supported in part by NSF Faculty Early Career Development (CAREER) Award #1054133.

References

[1] A. Aker and R. Gaizauskas. Generating image descriptions using dependency relational patterns. In *Pr. ACL*, pages 1250–1258, 2010. 1601

[2] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. Jordan. Matching words and pictures. *JMLR*, 3:1107–1135, 2003. 1602

[3] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean. Large language models in machine translation. In *EMNLP-CoNLL*, 2007. 1602

[4] S. Channarukul, S. W. McRoy, and S. S. Ali. Doghed: a template-based generator for multimodal dialog systems targeting heterogeneous devices. In *NAACL*, 2003. 1602

[5] M.-C. de Marnee and C. D. Manning. Stanford typed dependencies manual. 1605

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 1604, 1605

[7] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. In *ICCV*, 2009. 1602

[8] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation. In *ECCV*, 2002. 1602

[9] A. Farhadi, I. Endres, D. Hoiem, and D. A. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 1602, 1604, 1605

[10] A. Farhadi, M. Hejrati, A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. A. Forsyth. Every picture tells a story: generating sentences for images. In *ECCV*, 2010. 1602

[11] L. Fei-Fei, C. Koch, A. Iyer, and P. Perona. What do we see when we glance at a scene. *Journal of Vision*, 4(8), 2004. 1601

[12] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. http://people.cs.uchicago.edu/ pff/latent-release4/. 1604

[13] Y. Feng and M. Lapata. How many words is a picture worth? automatic caption generation for news images. In *Pr. ACL*, ACL '10, pages 1239–1249, 2010. 1601, 1602

[14] V. Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, 2007. 1602

[15] C. Galleguillos, A. Rabinovich, and S. J. Belongie. Object categorization using co-occurrence, location and appearance. In *CVPR*, 2008. 1602

[16] A. Gupta and L. S. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *ECCV*, 2008. 1602

[17] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *TPAMI*, 28, Oct. 2006. 1604

[18] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, 2009. 1602

[19] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 1602

[20] K. Papineni, S. Roukos, T. Ward, and W. jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *IBM Research Report*, 2001. 1607

[21] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 81:2–23, January 2009. 1602

[22] H. Stehouwer and M. van Zaanen. Language models for contextual error detection and correction. In *CLAGI*, 2009. 1602

[23] A. Torralba, R. Fergus, and W. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *TPAMI*, 30, 2008. 1601

[24] A. Torralba, K. P. Murphy, and W. T. Freeman. Using the forest to see the trees: exploiting context for visual object detection and localization. *Commun. ACM*, 53, March 2010. 1602

[25] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Map estimation via agreement on (hyper)trees: Message-passing and linear-programming approaches. *IEEE Tr Information Theory*, 51:3697–3717, 2005. 1604

[26] B. Yao, X. Yang, L. Lin, M. W. Lee, and S.-C. Zhu. I2t: Image parsing to text description. *Proc. IEEE*, 98(8), 2010. 1602

[27] L. Zhou and E. Hovy. Template-filtered headline summarization. In *Text Summarization Branches Out: Pr ACL-04 Wkshp*, July 2004. 1602