See No Evil, Say No Evil: Description Generation from Densely Labeled Images

Mark Yatskar [1*] my89@cs.washington.edu
Michel Galley[2] mgalley@microsoft.com
Lucy Vanderwende[2] lucyv@microsoft.com
Luke Zettlemoyer[1] lsz@cs.washington.edu

[1]Computer Science & Engineering
University of Washington
Seattle, WA, 98195, USA

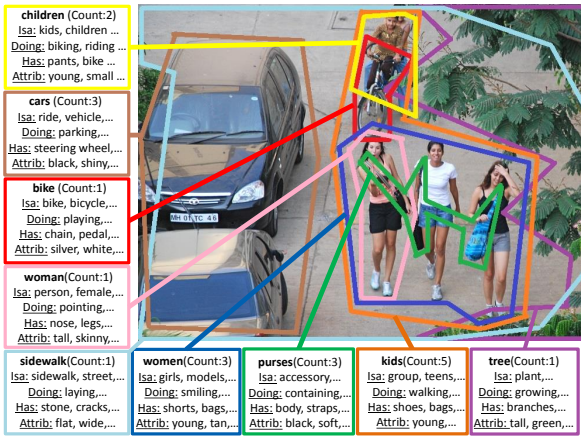
[2]Microsoft Research
One Microsoft Way
Redmond, WA, 98052, USA

Abstract

This paper studies generation of descriptive sentences from densely annotated images. Previous work studied generation from automatically detected visual information but produced a limited class of sentences, hindered by currently unreliable recognition of activities and attributes. Instead, we collect human annotations of objects, parts, attributes and activities in images. These annotations allow us to build a significantly more comprehensive model of language generation and allow us to study what visual information is required to generate human-like descriptions. Experiments demonstrate high quality output and that activity annotations and relative spatial location of objects contribute most to producing high quality sentences.

1 Introduction

Image descriptions compactly summarize complex visual scenes. For example, consider the descriptions of the image in Figure 1, which vary in content but focus on the women and what they are doing. Automatically generating such descriptions is challenging: a full system must understand the image, select the relevant visual content to present, and construct complete sentences. Existing systems aim to address all of these challenges but use visual detectors for only a small vocabulary of words, typically nouns, associated with objects that can be reliably found.[1] Such systems are blind to much of the visual content needed to generate complete, human-like sentences.



Five young people on the street, two sharing a bicycle.
Several young people are walking near parked vehicles.
Three girls with large handbags walking down the sidewalk.
Three women walk down a city street, as seen from above.
Three young woman walking down a sidewalk looking up.

Figure 1: An annotated image with human generated sentence descriptions. Each bounding polygon encompasses one or more objects and is associated with a count and text labels.This image has 9 high level objects annotated with over 250 textual labels.

In this paper, we instead study generation with more complete visual support, as provided by human annotations, allowing us to develop more comprehensive models than previously considered. Such models have the dual benefit of (1) providing new insights into how to construct more human-like sentences and (2) allowing us to perform experiments that systematically study the contribution of different visual cues in generation, suggesting which automatic detectors would be most beneficial for generation.

In an effort to approximate relatively complete visual recognition, we collected manually labeled representations of objects, parts, attributes and activities for a benchmark caption generation dataset that includes images paired with human authored

*This work was conducted at Microsoft Research.

[1]While object recognition is improving (ImageNet accuracy is over 90% for 1000 classes) progress in activity recognition has been slower; the state of the art is below 50% mean average precision for 40 activity classes (Yao et al., 2011).

descriptions (Rashtchian et al., 2010).[2] As seen in Figure 1, the labels include object boundaries and descriptive text, here including the facts that the children are "riding" and "walking" and that they are "young." Our goal is to be as exhaustive as possible, giving equal treatment to all objects. For example, the annotations in Figure 1 contain enough information to generate the first three sentences and most of the content in the remaining two. Labels gathered in this way are a type of feature norms (McRae et al., 2005), which have been used in the cognitive science literature to approximate human perception and were recently used as a visual proxy in distributional semantics (Silberer and Lapata, 2012). We present the first effort, that we are aware of, for using feature norms to study image description generation.

Such rich data allows us to develop significantly more comprehensive generation models. We divide generation into choices about which visual content to select and how to realize a sentence that describes that content. Our approach is grammar-based, feature-rich, and jointly models both decisions. The content selection model includes latent variables that align phrases to visual objects and features that, for example, measure how visual salience and spatial relationships influence which objects are mentioned. The realization approach considers a number of cues, including language model scores, word specificity, and relative spatial information (e.g. to produce the best spatial prepositions), when producing the final sentence. When used with a reranking model, including global cues such as sentence length, this approach provides a full generation system.

Our experiments demonstrate high quality visual content selection, within 90% of human performance on unigram BLEU, and improved complete sentence generation, nearly halving the difference from human performance to two baselines on 4-gram BLEU. In ablations, we measure the importance of different annotations and visual cues, showing that annotation of activities and relative bounding box information between objects are crucial to generating human-like description.

2 Related Work

A number of approaches have been proposed for constructing sentences from images, including copying captions from other images (Farhadi et al., 2010; Ordonez et al., 2011), using text surrounding an image in a news article (Feng and Lapata, 2010), filling visual sentence templates (Kulkarni et al., 2011; Yang et al., 2011; Elliott and Keller, 2013), and stitching together existing sentence descriptions (Gupta and Mannem, 2012; Kuznetsova et al., 2012). However, due to the lack of reliable detectors, especially for activities, many previous systems have a small vocabulary and must generate many words, including verbs, with no direct visual support. These problems also extend to video caption systems (Yu and Siskind, 2013; Krishnamoorthy et al., 2013).

The Midge algorithm (Mitchell et al., 2012) is most closely related to our approach, and will provide a baseline in our experiments. Midge is syntax-driven but again uses a small vocabulary without direct visual support for every word. It outputs a large set of sentences to describe all triplets of recognized objects in the scene, but does not include a content selection model to select the best sentence. We extend Midge with content and sentence selection rules to use it as a baseline.

The visual facts we annotate are motivated by research in machine vision. Attributes are a good intermediate representation for categorization (Farhadi et al., 2009). Activity recognition is an emerging area in images (Li and Fei-Fei, 2007; Yao et al., 2011; Sharma et al., 2013) and video (Weinland et al., 2011), although less studied than object recognition. Also, parts have been widely used in object recognition (Felzenszwalb et al., 2010). Yet, no work tests the contribution of these labels for sentence generation.

There is also a significant amount of work on other grounded language problems, where related models have been developed. Visual referring expression generation systems (Krahmer and Van Deemter, 2012; Mitchell et al., 2013; FitzGerald et al., 2013) aim to identify specific objects, a sub-problem we deal with when describing images more generally. Other research generates descriptions in simulated worlds and, like this work, uses feature rich models (Angeli et al., 2010), or syntactic structures like PCFGs (Chen et al., 2010; Konstas and Lapata, 2012) but does not combine the two. Finally, Zitnick and Parikh (2013) study sentences describing clipart scenes. They present a number of factors influencing overall descriptive quality, several of which we use in sentence generation for the first time.

[2] Available at : http://homes.cs.washington.edu/~my89/

3 Dataset

We collected a dataset of richly annotated images to approximate gold standard visual recognition. In collecting the data, we sought a visual annotation with sufficient coverage to support the generation of as many of the words in the original image descriptions as possible. We also aimed to make it as visually exhaustive as possible—giving equal treatment to all visible objects. This ensures less bias from annotators' perception about which objects are important, since one of the problems we would like to solve is content selection. This dataset will be available for future experiments.

We built on the dataset from (Rashtchian et al., 2010) which contained 8,000 Flickr images and associated descriptions gathered using Amazon Mechanical Turk (MTurk). Restricting ourselves to Creative Commons images, we sampled 500 images for annotation.

We collected annotations of images in three stages using MTurk, and assigned each annotation task to 3-5 workers to improve quality through redundancy (Callison-Burch, 2009). Below we describe the process for annotating a single image.

Stage 1: We prompted five turkers to list *all* objects in an image, ignoring objects that are parts of larger objects (e.g., the arms of a person), which we collected later in Stage 3. This list also included groups, such as crowds of people.

Stage 2: For each unique object label from Stage 1, we asked two turkers to draw a polygon around the object identified.[3] In cases where the object is a group, we also asked for the number of objects present (1-6 or many). Finally, we created a list of all references to the object from the first stage, which we call the *Object* facet.

Stage 3: For each object or group, we prompted three turkers to provide descriptive phrases of:

- *Doing* – actions the object participates in, e.g. "jumping."
- *Parts* – physical parts e.g. "legs", or other items in the possession of the object e.g. "shirt."
- *Attributes* – adjectives describing the object, e.g. "red."
- *Isa* – alternative names for a object e.g. "boy", "rider."

Figure 1 shows more examples for objects in a labeled image.[4] We refer to all of these annotations, including the merged *Object* labels, as facets. These labels provide feature norms (McRae et al., 2005), which have recently used as a visual proxy in distributional semantics (Silberer and Lapata, 2012; Silberer et al., 2013) but have not been previous studied for generation. This annotation of 500 images (2500 sentences) yielded over 4000 object instances and 100,000 textual labels.

[3] We modified LabelMe (Torralba et al., 2010).

4 Approach

Given such rich annotations, we can now develop significantly more comprehensive generation models. In this section, we present an approach that first uses a generative model and then a reranker. The generative model defines a distribution over content selection and content realization choices, using diverse cues from the image annotations. The reranker trades off our generative model score, language model score (to encourage fluency), and length to produce the final sentence.

Generative Model We want to generate a sentence $\vec{w} = \langle w_1 \dots w_n \rangle$ where each word $w_i \in V$ comes from a fixed vocabulary V. The vocabulary V includes all 2700 words used in descriptive sentences in the training set.[5]

The model conditions on an annotated image I that contains a set of objects O, where each object $o \in O$ has a bounding polygon and a number of facets containing string labels. To model the naming of specific objects, words w_i can be associated with alignment variables a_i that range over O. One such variable is introduced for each head noun in the sentence. Figure 2 shows alignment variable settings with colors that match objects in the image. Finally, as a byproduct of the hierarchical generative process, we construct an undirected dependency tree $\vec{d}$ over the words in $\vec{w}$.

The complete generative model defines the probability $p(\vec{w}, \vec{a}, \vec{d} \mid I)$ of a sentence $\vec{w}$, word alignments $\vec{a}$, and undirected dependency tree $\vec{d}$, given the annotated input image I. The overall process unfolds recursively, as seen in Figure 3.

[4] In the experiments, Parts and Isa facets do not improve performance, so we do not use them in the final model. Isa is redundant with the Object facet, as seen in Figure 1. Also parts like clothing, were often annotated as separate objects.

[5] We do not generate from image facets directly, because only 20% of the sentences in our data can be produced like this. Instead, we develop features which consider the similarity between labels in the image and words in the vocabulary.

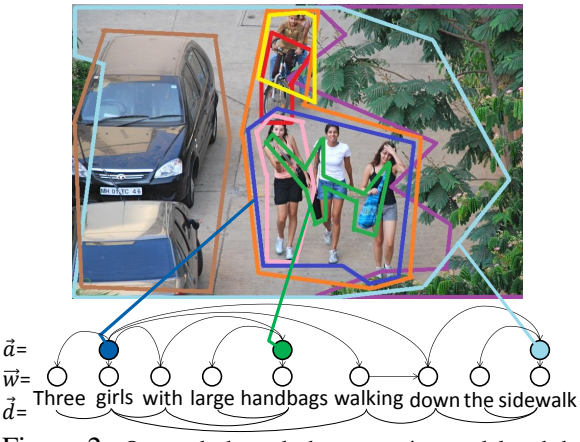


Figure 2: One path through the generative model and the Bayesian network it induces. The first row of colored circles are alignment variables to objects in the image. The second row is words, generated conditioned on alignments.

The main clause is produced by first selecting the subject alignment a_s followed by the subject word w_s. It then chooses the verb and optionally the object alignment a_o and word w_o. The process then continues recursively, modifying the subject, verb, and object of the sentence with noun and prepositional modifiers. The recursion begins at Step 2 in Figure 3. Given a parent word w and that word's relevant alignment variable a, the model creates attachments where w is the grammatical head of subsequently produced words. Choices about whether to create noun modifiers or prepositional modifiers are made in steps (a) and (b). The process chooses values for the alignment variables and then chooses content words, adding connective prepositions in the case of prepositional modifiers. It then chooses to end or submits new word-alignment pairs to be recursively modified.

Each line defines a decision that must be made according to a local probability distribution. For example, Step 1.a defines the probability of aligning a subject word to various objects in the image. The distributions are maximum entropy models, similar to previous work (Angeli et al., 2010), using features described in the next section. The induced undirected dependency tree $\vec{d}$ has an edge between each word and the previously generated word (or the input word w in Steps 2.a.i and 2.a.ii, when no previous word is available). Figure 2 shows a possible output from the process, along with the Bayesian network that encodes what each decision was conditioned on during generation.

Learning We learn the model from data $\{(\vec{w}_i, \vec{d}_i, I_i) \mid i = 1 \ldots m\}$ containing sentences $\vec{w}_i$, dependency trees $\vec{d}_i$, computed with the Stanford parser (de Marneffe et al., 2006), and images

1. for a main clause (d,e are optional), select:
   (a) subject a_s alignment from $p_a(a)$.
   (b) subject word w_s from $p_n(w \mid a_s, \vec{d}_c)$
   (c) verb word w_v from $p_v(w \mid a_s, \vec{d}_c)$
   (d) object alignment a_o from $p_a(a' \mid a_s, w_v, \vec{d}_c)$
   (e) object word w_o from $p_n(w \mid a_o, \vec{d}_c)$
   (f) end with p_{stop} or go to (2) with (w_s, a_s)
   (g) end with p_{stop} or go to (2) with (w_v, a_s)
   (h) end with p_{stop} or go to (2) with (w_o, a_o)
2. for a (word, alignment) (w', a) (a,b are optional):
   (a) if w' not verb: modify w' with noun, select:
      i. modifier word w_n from $p_n(w \mid a, \vec{d}_c)$.
      ii. end with p_{stop} or go to (2) with (a_m, w_n)
   (b) modify w' with preposition, select:
      i. preposition word w_p
      if w' not a verb: from $p_p(w \mid a, \vec{d}_c)$
      else: from $p_p(w \mid a, w_v, \vec{d}_c)$
      ii. object alignment a_p from $p_a(a' \mid a, w_p, \vec{d}_c)$
      iii. object word w_n from $p_n(w \mid a_p, \vec{d}_c)$.
      iv. end with p_{stop} or go to (2) with (a_p, w_n)

Figure 3: Generative process for producing words $\vec{w}$, alignments $\vec{a}$ and dependencies $\vec{d}$. Each distribution is conditioned on the partially complete path through generative process $\vec{d}_c$ to establish sentence context. The notation p_{stop} is short hand for $p_{stop}(STOP|\vec{w}, \vec{d}_c)$ the stopping distribution.

I_i. The dependency trees define the path that was taken through the generative process in Figure 3 and are used to create a Bayesian network for every sentence, like in Figure 2. However, object alignments $\vec{a}_i$ are latent during learning and we must marginalize over them.

The model is trained to maximize the conditional marginal log-likelihood of the data with regularization:

$$\mathcal{L}(\theta) = \sum_i \log \sum_{\vec{a}} p(\vec{a}, \vec{w}_i, \vec{d}_i \mid I_i; \theta) - r|\theta|^2$$

where θ is the set of parameters and r is the regularization coefficient. In essence, we maximize the likelihood of every sentence's observed Bayesian network, while marginalizing over content selection variables we did not observe.

Because the model only includes pairwise dependencies between the hidden alignment variables $\vec{a}$, the inference problem is quadratic in the number of objects and non-convex because $\vec{a}$ is unobserved. We optimize this objective directly with L-BFGS, using the junction-tree algorithm to compute the sum and the gradient.[6]

[6]To compute the gradient, we differentiate the recurrence in the junction-tree algorithm by applying the product rule.

Inference To describe an image, we need to maximize over word, alignment, and the dependency parse variables:

$$\arg\max_{\vec{w},\vec{a},\vec{d}} p(\vec{w},\vec{a},\vec{d} \mid I)$$

This computation is intractable because we need to consider all possible sentences, so we use beam search for strings up to a fixed length.

Reranking Generating directly from the process in Figure 3 results in sentences that may be short and repetitive because the model score is a product of locally normalized distributions. The reranker takes as input a candidate list c, for an image I, as decoded from the generative model. The candidate list includes the top-k scoring hypotheses for each sentence length up to a fixed maximum. A linear scoring function is used for reranking optimized with MERT (Och, 2003) to maximize BLEU-2.

5 Features

We construct indicator features to capture variation in usage in different parts of the sentence, types of objects that are mentioned, visual salience, and semantic and visual coordination between objects. The features are included in the maximum entropy models used to parameterize the distributions described in Figure 3. Whenever possible, we use WordNet Synsets (Miller, 1995) instead of lexical features to limit over-fitting.

Features in the generative model use tests for local properties, such as the identity of a synset of a word in WordNet, conjoined with an identifier that indicates context in the generative process.[7] Generative model features indicate (1) visual and semantic information about objects in distributions over alignments (content selection) and (2) preferences for referring to objects in distributions over words (content realization). Features in the reranking model indicate global properties of candidate sentences. Exact formulas for computing the features are in the appendix.

Visual features, such as an object's position in the image, are used for content selection. Pairwise visual information between two objects, for example the bounding box overlap between objects or the relative position of the two objects, is included in distributions where selection of an alignment variable conditions on previously generated alignments. For verbs (Step 1.d in Figure 3) and prepositions (Step 2.b.ii), these features are conjoined with the stem of the connective.

Semantic types of objects are also used in content selection. We define semantic types by finding synsets of labels in objects that correspond to high level types, a list motivated by the animacy hierarchy (Zaenen et al., 2004).[8] Type features indicate the type of the object referred to by an alignment variable as well as the cross product of types when an alignment variable is on conditioning side of a distribution (e.g. Step 1.d). Like above, in the presence of a connective word, these features are conjoined with the stem of the connective.

Content realization features help select words when conditioning on chosen alignments (e.g. Step 1.b). These features include the identity of the WordNet synset corresponding to a word, the word's depth in the synset hierarchy, the language model score for adding that word[9] and whether the word matches labels in facets corresponding to the object referenced by an alignment variable.

Reranking features are primarily used to overcome issues of repetition and length in the generative distributions, more commonly used for alignment, than to create sentences. We use only four features: length, the number of repetitions, generative model score, and language model score.

6 Experimental Setup

Data We used 70% of the data for training (1750 sentences, 350 images), 15% for development, and 15% for testing (375 sentences, 75 images).

Parameters The regularization parameter was set on the held out data to $r = 8$. The reranker candidate list included the top 500 sentences for each sentence length up to 15 and weights were optimized with Z-MERT (Zaidan, 2009).

Metrics Our evaluation is based on BLEU-n (Papineni et al., 2001), which considers all n-grams up to length n. To assess human performance using BLEU, we score each of the five references against the four other ones and finally average the five BLEU scores. In order to make these results comparable to BLEU scores for our model

[7] For example, in Figure 2 the context for the word "sidewalk" would be "word,syntactic-object,verb,preposition" indicating it is a word, in the syntactic object of a preposition, which was attached to a verb modifying prepositional phrase.

[8] For example, human, animal, artifact (a human created object), natural body (trees, water, ect.), or natural artifact (stick, leaf, rock).

[9] We use tri-grams with Kneser-Ney smoothing over the 1 million caption data set (Ordonez et al., 2011).

and baselines, we perform the same five-fold averaging when computing BLEU for each system.

We also compute accuracy for different syntactic positions in the sentence. We look at a number of categories: the main clause's components (S,V,O), prepositional phrase components, the preposition (Pp) and their objects (Po) and noun modifying words (N), including determiners. Phrases match if they have an exact string match and share context identifiers as defined in the features sections.

Human Evaluation Annotators rated sentences output by our full model against either human or a baseline system generated descriptions. Three criteria were evaluated: grammaticality, which sentence is more complete and well formed; truthfulness, which sentence is more accurately capturing something true in the image; and salience, which sentence is capturing important things in the image while still being concise. Two annotators annotated all test pairs for all criteria for a given pair of systems. Six annotators were used (none authors) and agreement was high (Cohen's kappa = 0.963, 0.823 and 0.703 for grammar, truth and salience).

Machine Translation Baseline The first baseline is designed to see if it is possible to generate good sentences from the facet string labels alone, with no visual information. We use an extension of phrase-based machine translation techniques (Och et al., 1999). We created a virtual bitext by pairing each image description (the target sentence) with a sequence[10] of visual identifiers (the source "sentence") listing strings from the facet labels. Since phrases produced by turkers lack many of the functions words needed to create fluent sentences, we added one of 47 function words either at the start or the end of each output phrase.

The translation model included standard features such as language model score (using our caption language model described previously), word count, phrase count, linear distortion, and the count of deleted source words. We also define three features that count the number of *Object*, *Isa*, and *Doing* phrases, to learn a preference for types of phrases. The feature weights are tuned with MERT (Och, 2003) to maximize BLEU-4.

Midge Baseline As described in related work, the Midge system creates a set of sentences to describe everything in an input image. These sentences must all be true, but do not have to select the same content that a person would. It can be adapted to our task by adding object selection and sentence ranking rules. For object selection, we choose the three most frequently named objects in the scene according to a background corpus of image descriptions. For sentence selection, we take all sentences within one word of the average length of a sentence in our corpus, 11, and select the one with best Midge generation score.

[10]We defined a consistent ordering of visual identifiers and set the distortion limit of the phrase-based decoder to infinity.

	BL-1	BL-2	BL-3	BL-4
Human	61.0	42.0	27.8	18.3
Full Model	**57.1**	**35.7**	**18.3**	**9.5**
MT Baseline	39.8	23.6	13.2	6.1
Midge Baseline	43.5	20.2	9.4	0.0

Table 1: Results for the test set for the BLEU1-4 metrics.

Grammar	Full	Other	Equal
Full vs **Human**	7.65	19.4	72.94
Full vs MT	6.47	5.29	88.23
Full vs Midge	40.59	15.88	43.53
Truth	Full	Other	Equal
Full vs **Human**	0.59	67.65	31.76
Full vs MT	30.0	10.59	59.41
Full vs Midge	51.76	27.71	23.53
Salience	Full	Other	Equal
Full vs **Human**	8.82	88.24	2.94
Full vs MT	51.76	16.47	31.77
Full vs Midge	71.18	14.71	14.12

Table 2: Human evaluation of our Full-Model in heads up tests against Human authored sentences and baseline systems, the machine translation baseline (MT) and the Midge inspired baseline. **Bold** indicates the better system. Other is not the Full system. Equal indicates neither sentence is better.

7 Results

We report experiments for our generation pipeline and ablations that remove data and features.

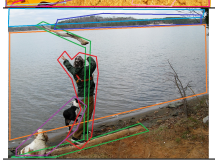
Overall Performance Table 1 shows the results on the test set. The full model consistently achieves the highest BLEU scores. Overall, these numbers suggest strong content selection by getting high recall for individual words (BLEU-1), but fall further behind human performance as the length of the n-gram grows (BLEU-2 through BLEU-4). These number match our perception that the model is learning to produce high quality sentences, but does not always describe all of the important aspects of the scene or use exactly the expected wording. Table 4 presents example output, which we will discuss in more detail shortly.

Model	BL-1	BL-2	BL-3	BL-4	S	V	O	Pp	Po	N
Human	64.7	46.0	31.5	20.1	-	-	-	-	-	-
Full-Model	**59.0**	36.9	**19.3**	**10.5**	**64.9**	**40.4**	36.8	50.0	20.7	69.1
– doing	51.1	32.6	16.9	9.2	63.2	15.8	10.5	45.5	**21.6**	69.7
– count	55.4	33.5	16.0	8.5	59.6	35.1	15.4	**53.7**	19.5	66.7
– properties	57.8	**37.2**	18.8	10.0	61.4	36.8	36.8	47.1	20.7	**73.5**
– visual	56.7	35.1	18.9	9.4	**64.9**	36.8	**50.0**	41.8	15.3	71.6
– pairwise	56.9	35.5	16.5	8.2	**64.9**	**40.4**	45.5	42.4	21.2	70.9

Table 3: Ablation results on development data using BLEU1-4 and reporting match accuracy for sentence structures.

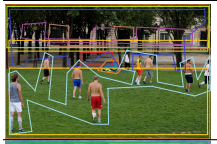
S: A girl playing a guitar in the grass

R: A woman with a nylon stringed guitar is playing in a field



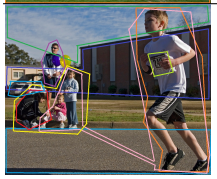
S: A man playing with two dogs in the water

R: A man is throwing a log into a waterway while two dogs watch



S: Two men playing with a bench in the grass

R: Nine men are playing a game in the park, shirts versus skins



S: Three kids sitting on a road

R: A boy runs in a race while onlookers watch

Table 4: Two good examples of output (top), and two examples of poor performance (bottom). Each image has two captions, the system output **S** and a human reference **R**.

Human Evaluation Table 2 presents the results of a human evaluation. The full model outperforms all baselines on every measure, but is not always competitive with human descriptions. It performs the best on grammaticality, where it is judged to be as grammatical as humans. However, surprisingly, in many cases it is also often judged equal to the other baselines. Examination of baseline output reveals that the MT baseline often generates short sentences, having little chance of being judged ungrammatical. Furthermore, the Midge baseline, like our system, is a syntax-based system and therefore often produces grammatical sentences. Although our system performs well with respect to the baselines on truthfulness, often the system constructs sentences with incorrect prepositions, an issue that could be improved with better estimates of 3-d position in the image. On truthfulness, the MT baseline is comparable to our system, often being judged equal, because its output is short. Our system's strength is salience, a factor the baselines do not model.

Data Ablation Table 3 shows annotation ablation experiments on the development set, where we remove different classes of data labels to measure the performance that can be achieved with less visual information. In all cases, the overall behavior of the system varies, as it tries to learn to compensate for the missing information.

Ablating actions is by far the most detrimental. Overall BLEU score suffers and prediction accuracy of the verb (V) degrades significantly causing cascading errors that affect the object of the verb (O). Removing count information affects noun attachment (N) performance. Images where determiner use is important or where groups of objects are best identified by the number (for example, three dogs) are difficult to describe naturally. Finally, we see a tradeoff when removing properties. There is an increase in noun modifier accuracy (N) but a decrease in content selection quality (BL-1), showing recall has gone down. In essence, the approach learns to stop trying to generate adjectives and other modifiers that would rely on the missing properties. The difference in BLEU score with the Full-Model is small, even without these modifiers, because there often still exists a a short output with high accuracy.

Feature Ablation The bottom two rows in Table 3 show ablations of the visual and pairwise features, measuring the contribution of the visual information provided by the bounding box annotations. The ablated visual information includes bounding-box positions and relative pairwise visual information. The pairwise ablation removes the ability to model any interactions between objects, for example, relative bounding box or pairwise object type information.

Overall, prepositional phrase accuracy is most affected. Ablating visual features significantly impacts accuracy of prepositional phrases (Pp and Po), affecting the use of preposition words the most, and lowering fluency (BL-4). Precision in

the object of the verb (O) rises; the model makes $\sim 50\%$ fewer predictions in that position than the Full-Model because it lacks features to coordinate subject and object of the verb. Ablating pairwise features has similar results. While the model corrects errors in the object of the preposition (Po) with the addition of visual features, fluency is still worse than Full-Model, as reflected by BL-4.

Qualitative Results Table 4 has examples of good and bad system output. The first two images are good examples, including both system output (**S**) and a human reference (**R**). The second two contain lower quality outputs. Overall, the model captures common ways to refer to people and scenes. However, it does better for images with fewer sentient objects because content selection is less ambiguous.

Our system does well at finding important objects. For example, in the first good image, we mention the guitar instead of the house, both of which are prominent and have high overlap with the woman. In the second case, we identify that both dogs and humans tend to be important actors in scenes but poorly identify their relationship.

The bad examples show difficult scenes. In the first description the broad context is not identified, instead focusing on the bench (highlighted in red). The second example identifies a weakness in our annotation: it encodes contradictory groupings of the people. The groupings covers all of the children, including the boy running, and many subsets of the people near the grass. This causes ambiguity and our methods cannot differentiate them, incorrectly mentioning just the children and picking an inappropriate verb (one participant in the group is not sitting). Improved annotation of groups would enable the study of generation for more complex scenes, such as these.

8 Conclusion

In this work we used dense annotations of images to study description generation. The annotations allowed us to not only develop new models, better capable of generating human-like sentences, but also to explore what visual information is crucial for description generation. Experiments showed that activity and bounding-box information is important and demonstrated areas of future work. In images that are more complex, for example multiple sentient objects, object grouping and reference will be important to generating good descriptions. Issues of this type can be explored with annotations of increasing complexity.

Appendix A

This appendix describes the feature templates for the generative model in greater detail.

Features in the generative model conjoin indicators for local tests, such as $\mathbf{STEM}(w)$ which indicates the stem of a word w, with a global contextual identifier $\mathbf{CONTEXT}(v, d)$ that indicates properties of the generation history, as described in detail below. Table 5 provides a reference for which feature templates are used in the generative model distributions, as defined in Figure 3.

8.1 Feature Templates

$\mathbf{CONTEXT}(n, d)$ is an indicator for a contextual identifier for a variable n in the model depending on the dependency structure d. There is an indicator for all combinations of the type of n (alignment or word), the position of n (subject, syntactic object, verb, noun-modifier, or preposition), the position of the earliest variable along the path to generate n, and the type of attachment to that variable (noun or prepositional modifier). For example, in Figure 2 the context for the word "sidewalk" would be "word,syntactic-object,verb,preposition" indicating it is a word, the object of a preposition, whose path was along a verb modifying prepositional phrase.[11]

$\mathbf{TYPE}(a)$ indicates the high level type of an object referred to by alignment variable a. We use synsets to define high level types including human, animal, artifact, natural artifact and various synsets that capture scene information,[12] a list motivated by the animacy hierarchy (Zaenen et al., 2004). Each object is assigned a type by finding the synset for its name (object facet), and tracing the hypernym structure in Wordnet to find the appropriate class, if one exists. Additionally, the type indicates whether the object is a group or not. For example, in Figure 2, the blue polygon has type "person,group", or the red bike polygon has type "artifact,single."

[11] Similarly "large" is "word,noun,subject,preposition" while "girls" is special cased to "word,subject,root" because it has no initial attachment. The alignment variable above the word handbags is "alignment,syntactic-object,subject,preposition" because it an alignment variable, is in the syntactic object position of a preposition and can be located by following a subject attached pp.

[12] WordNet divides these into synsets expressing water, weather, nature and a few more.

Feature Family	Included In	Steps
$\mathbf{CONTEXT}(a', \vec{d_c}) \otimes$ $\{\mathbf{TYPE}(a'), \mathbf{MENTION}(a', do), \mathbf{MENTION}(a', obj), \mathbf{VISUAL}(a')\}$	$p_a(a' \mid \vec{d_c})$ $p_a(a' \mid a, w, \vec{d_c})$	1.a, 1.d, 2.b.ii
$\mathbf{CONTEXT}(a', \vec{d_c}) \otimes \{\mathbf{TYPE}(a) \otimes \mathbf{TYPE}(a'), \mathbf{VISUAL2}(a, a')\}$	$p_a(a' \mid a, w, \vec{d_c})$	1.d, 2.b.i
$\mathbf{CONTEXT}(a', \vec{d_c}) \otimes$ $\{\mathbf{TYPE}(a) \otimes \mathbf{TYPE}(a') \otimes \mathbf{STEM}(w), \mathbf{VISUAL2}(a, a') \otimes \mathbf{STEM}(w)\}$	$p_a(a' \mid a, w, \vec{d_c})$	1.d, 2.b.i
$\mathbf{CONTEXT}(a, \vec{d_c}) \otimes$ $\{\mathbf{WORDNET}(w), \mathbf{MATCH}(w, a), \mathbf{SPECIFICITY}(w, a),$ $\mathbf{ADJECTIVE}(w, a), \mathbf{DETERMINER}(w, a)\}$	$p_n(w \mid a, \vec{d_c})$	1.b, 1.e, 2.a.i 2.b.ii
$\mathbf{CONTEXT}(a, \vec{d_c}) \otimes \{\mathbf{MATCH}(w, a), \mathbf{TYPE}(a) \otimes \mathbf{STEM}(w)\}$	$p_v(w \mid a, \vec{d_c})$	1.c
$\mathbf{CONTEXT}(a', \vec{d_c}) \otimes \mathbf{TYPE}(a) \otimes \mathbf{STEM}(w_p)$	$p_p(w \mid a, \vec{d_c})$ $p_p(w \mid a, w_v, \vec{d_c})$	2.b.i
$\mathbf{CONTEXT}(a', \vec{d_c}) \otimes \mathbf{STEM}(w_v) \otimes \mathbf{STEM}(w)$	$p_p(w \mid a, w_v, \vec{d_c})$	2.b.i

Table 5: Feature families and distributions that include them. $\otimes$ indicates the cross-product of the indicator features. Distributions are listed more than once to indicate they use multiple feature families.

$\mathbf{VISUAL}(a)$ returns indicators for visual facts about the object that a aligns to. There is an indicator for two quantities: (1) overlap of object's polygon with every horizontal third of the image, as a fraction of the object's area, and (2) the object's distance to the center of the image as fraction of the diagonal of the image. Each quantity, v, is put into three overlapping buckets: if $v > .1$, if $v > .5$, and if $v > .9$.

$\mathbf{VISUAL2}(a, a')$ indicates pairwise visual facts about two objects. There is an indicator for the following quantities bucketed: the amount of overlap between the polygons for a and a' as a fraction of the size of a's polygon, the distance between the center of the polygon for a and a' as a fraction of image's diagonal, and the slope between the center of a and a'. Each quantity, v, is put into three overlapping buckets: if $v > .1$, if $v > .5$, and if $v > .9$. There is an indicator for the relative position of extremities a and a': whether the rightmost point of a is further right than a''s rightmost or leftmost point, and the same for top, left, and bottom.

$\mathbf{WORDNET}(w)$ returns indicators for all hypernyms of a word w. The two most specific synsets are not used when there at least 8 options.

$\mathbf{MENTION}(a, facet)$ returns the union of the $\mathbf{WORDNET}(w)$ features for all words w in the facet *facet* for the object referred to alignment a.

$\mathbf{ADJECTIVE}(w, a)$ indicates four types of features specific to adjective usage. If $\mathbf{MENTION}(w, Attributes)$ is not empty, indicate : (1) the satellite adjective synset of w in Wordnet, (2) the head adjective synset of w in Wordnet, (3) the head adjective synset conjoined with $\mathbf{TYPE}(a)$, and (4) the number of times there exists a label in the Attributes facet of a that has the same head adjective synset as w.

$\mathbf{DETERMINER}(w, a)$ indicates four determiner specific features. If w is a determiner, then indicate : (1) the identity of w conjoined with the count (the label for numerosity) of a, (2) the identity of w conjoined with an indicator for if the count of a is greater than one, (3) the identity of w conjoined with $\mathbf{TYPE}(a)$ and (4) the frequency with which w appears before its head word in the Flikr corpus (Ordonez et al., 2011).

$\mathbf{MATCH}(w, a)$, indicates all *facets* of object a that contain words with the same stem as w.

$\mathbf{SPECIFICITY}(w, a)$ is an indicator of the specificity of the word w when referring to the object aligned to a. Indicates the relative depth of w in Wordnet, as compared to all words w' where $\mathbf{MATCH}(w', a)$ is not empty. The depth is bucketed into quintiles.

$\mathbf{STEM}(w)$ returns the Porter2 stem of w.[13]

The distribution for stopping, $p_{stop}(STOP \mid \vec{d_c}, \vec{w})$, contains two types of features. (1) Structural features indicating for the number of times a contextual identifier has appeared so far in the derivation and (2) mention features indicating the types of objects mentioned.[14] To compute mention features, we consider all possible types of objects, t, then there is an indicator for: (1) if $\exists o, \exists w \in \vec{w} : \mathbf{MATCH}(w, o) \neq \emptyset \wedge \mathbf{TYPE}(o) = t$, (2) whether $\exists o, \nexists w \in \vec{w} : \mathbf{MATCH}(w, o) \neq \emptyset \wedge \mathbf{TYPE}(o) = t$ and (3) if (1) does not hold but (2) does.

Acknowledgments This work is partially funded by DARPA CSSG (D11AP00277) and ARO (W911NF-12-1-0197). We thank L. Zitnick, B. Dolan, M. Mitchell, C. Quirk, A. Farhadi, B. Russell for helpful conversations. Also, L. Zilles, Y. Atrzi, N. FitzGerald, T. Kwiatkowski and reviewers for comments.

[13] http://snowball.tartarus.org/algorithms/english/stemmer.html

[14] Object mention features cannot contain $\vec{a}$ because that creates large dependencies in inference for learning.

References

Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Chris Callison-Burch. 2009. Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk. In *EMNLP*, pages 286–295, August.

David L. Chen, Joohyun Kim, and Raymond J. Mooney. 2010. Training a multilingual sportscaster: Using perceptual context to learn language. *JAIR*, 37:397–435.

Marie-Catherine de Marneffe, Bill MacCartney, Christopher D Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. In *LREC*, volume 6, pages 449–454.

Desmond Elliott and Frank Keller. 2013. Image Description using Visual Dependency Representations. In *EMNLP*.

Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. 2009. Describing objects by their attributes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1778–1785. IEEE.

Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. 2010. Every picture tells a story: Generating sentences from images. In *Proceedings of the 11th European conference on Computer Vision*, ECCV'10, pages 15–29.

Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. 2010. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645.

Yansong Feng and Mirella Lapata. 2010. How many words is a picture worth? Automatic caption generation for news images. In *ACL*, pages 1239–1249.

Nicholas FitzGerald, Yoav Artzi, and Luke Zettlemoyer. 2013. Learning distributions over logical forms for referring expression generation. In *EMNLP*.

Ankush Gupta and Prashanth Mannem. 2012. From image annotation to image description. In *NIPS*, volume 7667, pages 196–204.

Ioannis Konstas and Mirella Lapata. 2012. Concept-to-text generation via discriminative reranking. In *ACL*, pages 369–378.

Emiel Krahmer and Kees Van Deemter. 2012. Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1):173–218.

Niveda Krishnamoorthy, Girish Malkarnenkar, Raymond Mooney, Kate Saenko, and Sergio Guadarrama. 2013. Generating natural-language video descriptions using text-mined knowledge. *Procedings of AAAI*, 2013(2):3.

G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A.C. Berg, and T.L. Berg. 2011. Baby talk: Understanding and generating simple image descriptions. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1601–1608.

Polina Kuznetsova, Vicente Ordonez, Alexander C Berg, Tamara L Berg, and Yejin Choi. 2012. Collective generation of natural image descriptions. In *ACL*, pages 359–368.

Li-Jia Li and Li Fei-Fei. 2007. What, where and who? Classifying events by scene and object recognition. In *ICCV*, pages 1–8. IEEE.

Ken McRae, George S. Cree, Mark S. Seidenberg, and Chris Mcnorgan. 2005. Semantic feature production norms for a large set of living and nonliving things. *Behavior Research Methods*, 37(4):547–559.

George A Miller. 1995. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Margaret Mitchell, Xufeng Han, Jesse Dodge, Alyssa Mensch, Amit Goyal, Alex Berg, Kota Yamaguchi, Tamara Berg, Karl Stratos, and Hal Daumé, III. 2012. Midge: Generating image descriptions from computer vision detections. In *EACL*, pages 747–756.

Margaret Mitchell, Kees van Deemter, and Ehud Reiter. 2013. Generating expressions that refer to visible objects. In *Proceedings of NAACL-HLT*, pages 1174–1184.

F. Och, C. Tillmann, and H. Ney. 1999. Improved alignment models for statistical machine translation. In *Proc. of the Joint Conf. of Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *ACL*, pages 160–167.

Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. 2011. Im2Text: Describing images using 1 million captioned photographs. In *NIPS*, pages 1143–1151.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL*.

C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier. 2010. Collecting image annotations using Amazon's Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 139–147.

Gaurav Sharma, Frédéric Jurie, Cordelia Schmid, et al. 2013. Expanded parts model for human attribute and action recognition in still images. In *CVPR*.

Carina Silberer and Mirella Lapata. 2012. Grounded models of semantic representation. In *EMNLP*, July.

Carina Silberer, Vittorio Ferrari, and Mirella Lapata. 2013. Models of semantic representation with visual attributes. In *ACL*, pages 572–582.

Antonio Torralba, Bryan C Russell, and Jenny Yuen. 2010. LabelMe: Online image annotation and applications. *Proceedings of the IEEE*, 98(8):1467–1484.

Daniel Weinland, Remi Ronfard, and Edmond Boyer. 2011. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224–241.

Yezhou Yang, Ching Lik Teo, Hal Daumé III, and Yiannis Aloimonos. 2011. Corpus-guided sentence generation of natural images. In *Empirical Methods in Natural Language Processing*.

Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas J. Guibas, and Li Fei-Fei. 2011. Action recognition by learning bases of action attributes and parts. In *ICCV*, Barcelona, Spain, November.

Haonan Yu and Jeffrey Mark Siskind. 2013. Grounded language learning from video described with sentences. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 53–63.

Annie Zaenen, Jean Carletta, Gregory Garretson, Joan Bresnan, Andrew Koontz-Garboden, Tatiana Nikitina, M Catherine O'Connor, and Tom Wasow. 2004. Animacy encoding in English: why and how. In *ACL Workshop on Discourse Annotation*, pages 118–125.

Omar F. Zaidan. 2009. Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems. *The Prague Bulletin of Mathematical Linguistics*, 91:79–88.

C. Lawrence Zitnick and Devi Parikh. 2013. Bringing semantics into focus using visual abstraction. In *CVPR*.