Toward a General Framework for Words and Pictures

Alexander C. Berg
Stony Brook University

Tamara L. Berg
Stony Brook University

Hal Daumé III
University of Maryland

Jesse Dodge
University of Washington

Amit Goyal
University of Maryland

Xufeng Han
Stony Brook University

Alyssa Mensch
M.I.T.

Margaret Mitchell
University of Aberdeen

Karl Stratos
Columbia University

Kota Yamaguchi
Stony Brook University

Abstract

This is a report on activities as part of the JHU-CLSP summer workshops 2011. The report is followed by three papers currently in submission based on work during the summer and continuing work over the following semester. Two future paper submissions and a grant proposal are expected in addition to other follow-on activities.

1 Introduction

"It was an arresting face, pointed of chin, square of jaw. Her eyes were pale green without a touch of hazel, starred with bristly black lashes and slightly tilted at the ends. Above them, her thick black brows slanted upward, cutting a startling oblique line in her magnolia-white skin–that skin so prized by Southern women and so carefully guarded with bonnets, veils and mittens against hot Georgia suns"
– description of Scarlett O'Hara, Gone with the Wind.

Pictures convey a visual description of the world directly to their viewer. Computer vision strives to design algorithms to extract the underlying world state captured in the camera's eye, with an overarching goal of general computational image understanding. To date, much vision research has approached the goal of image understanding by focusing on object detection and localization. There has been amazing progress in this effort, both driven by, and resulting in larger and larger labeled image collections [27, 38, 22, 23, 78, 71, 1, 68], and more effective and efficient classification techniques [17, 29, 90, 33, 34, 82, 56, 55, 50]. These collections are becoming broader and more thorough with each iteration, but together represent only one perspective on the image understanding problem – where object presence is the main labeled information.

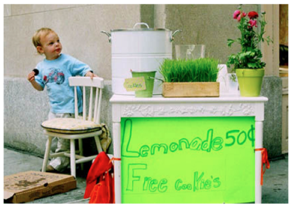
We propose that there is an additional, complimentary way to collect and describe information about the visual world – by directly analyzing the enormous amount of visually descriptive text available on the web. Textual descriptions, like the one above describing Scarlett O'Hara, can produce vivid images in the mind of the reader similar to the meaning conveyed by a picture. This visually descriptive text is often associated with images, *e.g.*, captions, tags, etc. Vivid visual narratives often appear in literature (with or without corresponding images) or in general text on web pages (often with related images). Instead of producing image collections and labeling the information to extract, we propose using existing descriptive text to guide data set construction by informing ideas – in particular the vocabulary – for image understanding. This will enable us to use the combined human knowledge of the visual world to reveal what information is useful to attach to, and extract from pictures.

Studying visual descriptions created by people can provide rich information about the visual world and will lead to more informed systems for understanding images, going far beyond estimating the object categories present in a picture. This will include additional useful information about *attributes* – visual characteristics of objects or scene regions – and *object interactions* – the arrangements or relationships between objects. These large quantities of textual data and associated image content are a natural source for examining how people capture and communicate annotations of the visual world around them. This approach is doubly effective because search is currently most often driven by textual queries as well. Learning the natural vocabulary and semantics of visual description is a critical step in improving the effectiveness of search for visual content.

Our six week summer workshop consisted of:

- refining computational models to identify visually descriptive text
- studying empirical statistics of such text
- linking these to underlying visual representations and to current computer vision techniques
- building statistical models of visually descriptive language and of outputs of computer vision algorithms – with a particular focus on modifiers (e.g., adjectives), prepositional phrases, actions, and scene context
- applying these models in tasks: auto-annotation, auto-illustration, and search. High level results include progress toward a natural vocabulary and structure for visual description that is useful for both the computer vision and language communities.

This was an ambitious workshop, but we had a strong team including broad based support at Johns Hopkins from the NLP side, good infrastructure in terms of existing software from our groups based on related prior work and open source routines, and most importantly ***prototypes*** for all stages of the proposed work, based on work in progress (both recently published and in submission). These were developed over the year before the workshop began. A main thrust of the workshop was *refining* how the structure of visually descriptive text can be effectively identified in large text corpora, and how to parse and extract useful structure and statistics from this text. Results will hopefully help reinforce interest in studying visually descriptive text in NLP, and suggest research directions for computer vision.



BabyTalk prototype automatic generation:
"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."

While there has been significant work on the connection between words and pictures in the past, the goal of this workshop is to both broaden and deepen the connections – especially identifying directions in which to encourage research on recognition in computer vision. Recent work in computer vision has begun to demonstrate the efficacy of broadening the scope of what to recognize – *e.g.*, considering attributes instead of object categories. This direction includes work from the organizers on the first example of an attribute based recognition system improving state of the art performance on widely attacked benchmark [46], and on automatically discovering text referring to visual attributes by mining paired text and images [9]. One of the goals of the workshop is to study language to motivate and inform future work in computer vision. This pursuit relies on multiple connections between language and vision: identifying visually descriptive text, defining what to recognize, defining priors on the visual world, providing regularization for the output of recognition, and determining useful output structure for applications like natural language generation and search using language queries.

In addition to the preliminary work described above, the workshop exploited synergies with other activities of the participants and work of several visitors. Prof. Tamara Berg from Stony Brook University had collected and begun to process the main dataset of 700,000 flickr images with descriptions used for the workshop. Prof. Alex Berg from Stony Brook University collaborates on the

ImageNet large scale visual recognition project with Fei-Fei Li's lab at Stanford University – this project was a source of recognition models. Professor Hal Daume brought datasets and expertise on efficiently computing and exploiting context statistics for large text corpora. Graduate student Margaret Mitchell brought her previous work on studying factors in referring expression generation and background in linguistics.

From visitors we obtained insight into the datasets of images with descriptions collected by Prof. Julia Hockenmaier at UIUC, clever analysis of when attributes and scene words are used from Prof. Erik Learned-Miller at UMass Amherst, and insight into practical optimization for generating natural language sentences from Prof. Yejin Choi at Stony Brook University.

A main potential outcome of the workshop is impact on both the computer vision and NLP communities. This is made likely by the proposers' active roles in the computer vision and NLP communities.

This workshop brought together language researchers and computer vision researchers. The participants include computer vision experts who work on large scale recognition (Alexander Berg), and problems related to combining information from words and pictures (Tamara Berg), and a natural language processing researcher (Hal Daume III).

Proposers: Tamara Berg studies the connection between words and pictures, particularly in her influential work on, "Names and Faces" [6, 8] and others *e.g.,* [4]. She is co-organizer (with Trevor Darrell, ICSI & UCB) of a NSF sponsored workshop to bring together linguists and computer vision researchers interested in exploiting deep language and visual structures (also Summer 2011). Alex Berg is active in the recognition and machine learning communities in computer vision (e.g. [56, 55, 90, 66, 46]), and is part of the ImageNet team [19], a major collection and labeling effort inspired by WordNet and hopefully useful here! He is co-organizing both the ImageNet Large Scale Visual Recognition Challenge (ongoing), and the Large Scale Learning in Computer Vision workshop (CVPR 2011).

2 Main Datasets

To study how people form natural language descriptions for images, and build systems to do the same, we begin by considering a number of different types of data from which to collect statistics and analyze performance. The primary data we use consists of examples of images with captions. These come from four sources. Starting with the UIUC Pascal Datasets of 1000/8000 images [69] from the Pascal VOC challenge [23], each with multiple captions collected from Mechanical Turk workers. The third dataset was collected by Prof. Tamara Berg and her students at Stony Brook University and consists of 700K photos from flickr with natural language descriptions written by the people posting the images (part of the 1 million captioned image dataset SBUCPD [65]). The fourth dataset is ImageCLEF [39, 21] consisting of 20K images with text descriptions, and critically with full segmentation and labeling of the image contents with 275 labels. Labels for image contents of the other datasets were collected by hand or using mechanical turk as needed. Auxiliary data from the MIT SUNS scene understanding dataset [85], the LabelMe dataset [71], the GigaWord corpus[35, 36], and WordNet [61] were used as part of specific experiments.

3 Characterizing descriptions of images: How and What.

We broke down studying descriptions of images into two main thrusts, considering how people describe images, and what image contents they describe.

3.1 Studying How People Describe Images

we look at statistics of patterns of language in the descriptive text to find quantifiable aspects of *how* people construct natural descriptions of images.

3.1.1 Statics of Descriptions – Margaret (Meg) Mitchell University of Aberdeen

We sought to understand how description works; what it means to 'describe' an image. We collected statistics from several corpora to explore factors at play in description, predominantly using Flickr as a large database of descriptive text. We were interested in discovering the kind of constituents that tend to make up descriptions, how many words tend to be mentioned, and and what kinds of words they are. Beyond raw statistics, we also leveraged WordNet [61] to understand the positions of different kinds of object classes, and semantic clustering described starting in Sec. 3.2.7 to understand the kinds of adjectives that tend to be used to describe objects.

Extracting this data from the Flickr corpus was only possible after a good deal of text normalization, and constituent parsing using the Berkeley parser [67]. The problem of text normalization is itself a well-recognized issue within NLP, especially when working with free text collected online from social media sites, and indeed, was the sole topic of a previous workshop [74].

However, our project was not focused on text normalization. While we spent some time cleaning up the data – removing HTML characters and emoticons, elaborating short-hand, handling a problematic artifact of the caption extraction process which concatenated phrases across newlines, building off of Norvig's spelling corrector [64] to train a model for this data – we have only begun to scratch the surface for this task. The statistics presented below are based off of the data we cleaned and normalized, however, before publication of final results on this dataset, we hope to more thoroughly address the text normalization issue.

To begin comparing the Flickr descriptions to writing in other domains, we use the first million sentences of the New York Times section of the Gigaword corpus [36], also parsed using the Berkeley parser. This is a relatively non-visual corpus of newswire text, and serves to highlight features of the Flickr corpus that may be specific to visual description.

In general, we find that the Flickr corpus is very different from NYT corpus, but does have similarities with other visual corpora, such as ImageClef [80]. This suggests that the patterns we find reflect the kind of language that is used to describe images, and is different from the kind of language used to describe other kinds of actions and events. These findings make several contributions to work on the vision-to-language connection. Broadly, they:

1. Establish a detailed characterization of descriptive text.
2. Provide features that may be used in a classification task to discriminate between descriptive and non-descriptive text.
3. Define basic characteristics that a system generating descriptive text should aim to achieve.

Below we list our findings and discuss their connection to visually descriptive text.

3.1.2 Phrase Types

Flickr data, there is usually just one sentence/main phrase) in a caption; occasionally there are two, where a line break designates a new unit. We will use the term *main phrase* to distinguish these kinds of full descriptive units from other kinds of syntactic subtrees. Each main phrase is analyzed individually in the following statistics.

number of main phrases	count	prob
1	520330	0.81
2	100826	0.16
3	15866	0.03
4	2547	0.00
5	559	0.00

Table 1: Counts and relative frequency of different amounts of main phrases in a caption.

We find that the number of words per sentence is significantly different than the number of words per sentence in other kinds of corpora, such as the NYT corpus. In general, Flickr descriptions are relatively short, with about 10 words per sentence (or phrase).

Flickr:	10.01
NYT:	19.96

Table 2: Average number of words per sentence/phrase.

description form in the Flickr data is usually a sentence, however, noun phrases (NPs) are quite common. Initial work on hand-parsing the data illustrates that the amount of noun phrases may be much higher, however, without domain adaptation, the parser has a bias to return full sentence structures from the given descriptions. Despite this bias, NPs still emerge as a common descriptive form in the captions. In future work, we will see how these trends play out when the parser has been trained on the hand-parsed Flickr data and so adapted to the Flickr domain.

3.1.3 Verbs and Prepositions

examine the distribution of verbs in the corpora by looking at all terminal nodes tagged as VB* or MD*. We find that Flickr main phrases tend to have just one verb, or no verb at all. In contrast, the NYT data has a much more even spread of verbs, which suggests that newswire text has phrases with much more varied complexity.

To further understand the kinds of verbs at play, we examined the difference in spread between light verbs and main verbs. We list the light verbs below.

In accord with our earlier findings, the NYT data tends to use more verbs than the Flickr data. It is more common to include a light verb in each sentence, while this is relatively rare in the descriptive text from Flickr.

The Flickr data also usually has 1 or 2 prepositions. This can be contrasted with the NYT data, which tends to have 2 or 3. This further reflects the idea that the descriptive Flickr text is in general less syntactically complex than newswire text.

3.1.4 Nouns

Most commonly, nouns in Flickr mention physical objects/things – they are concrete nouns – while nouns in the NYT data tend to be abstract objects, such as corporations.

In the Flickr data, 2 to 3 nouns tend to be mentioned in each main phrase, where 1 to 2 of those nouns are physical objects.

Flickr			**NYT**		
type	**count**	**prob.**	**type**	**count**	**prob.**
(S	455045	0.58	(S	896125	0.90
(NP	217197	0.28	(SINV	42441	0.04
(FRAG	24522	0.03	(NP	24465	0.03

Table 3: Types, counts, and relative frequencies of main phrases.

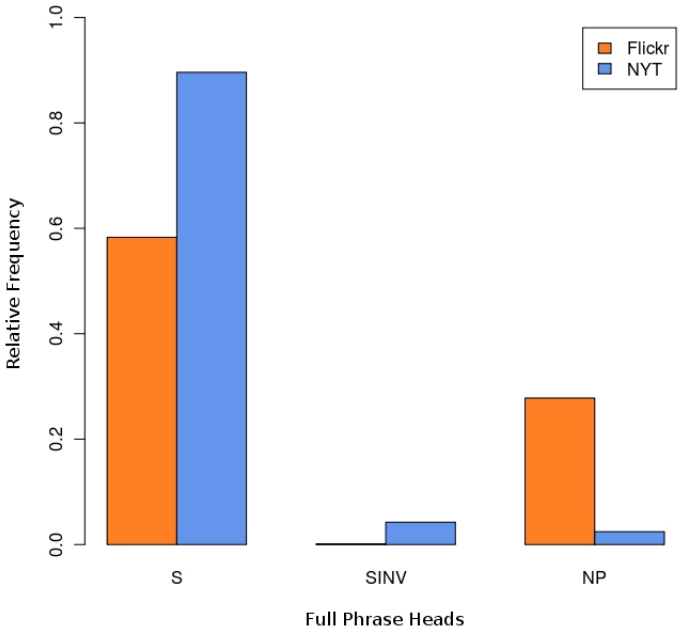


Table 4: Relative frequencies of main phrases.

am, *'m*, *are*, *'re*, *be*, *been*, *being*, *can*, *could*, *do*, *does*, *did*, *had*, *has*, *have*, *'ve*, *is*, *'s*, *may*, *might*, *must*, *shall*, *should*, *was*, *were*, *will*, *'ll*, *would*, *'d*

Table 5: Light verbs.

Flickr			**NYT**		
num	**count**	**prob.**	**num**	**count**	**prob.**
0	290778	0.37	0	38796	0.04
1	288088	0.37	1	154855	0.16
2	130000	0.17	2	213965	0.21
3	51452	0.07	3	197642	0.20
4	17680	0.02	4	152321	0.15

Table 6: Counts and relative frequencies of verb amounts per main phrase.

3.1.5 Adjective Classes

Our technique for clustering modifiers into semantic types so far only looks at modifiers tagged as JJ – adjectives. Modifiers can also have other visual modifier tags, e.g., VBG (*glowing* eyes), VBN (*cleaned* car), NN (*toy* car). We examine the coverage of the semantic types for all kinds of modifier tags: JJ, JJR, JJS, VBG, VBN, VBD, RB, RBR, RBS, NN, and NNS. Even when expanded to this quite large set, the semantic classes do quite well, covering around 30% of the words modifying physical objects in the Flickr corpus.

We next look at the amount of words that tend to modify physical objects. We find that there are usually 0 modifiers per physical object, but there is occasionally 1.

Further examining the coverage of the adjective classes, we find that Flickr is composed overwhelmingly of color, followed by size. This can be contrasted with the NYT data, where size adjectives are most common.

	Flickr	**NYT**
Average main:	0.79	2.25
Average light:	0.24	1.04

Table 7: Average number of light and main verbs per main phrase.

Flickr			**NYT**		
num	**count**	**prob.**	**num**	**count**	**prob.**
0	85838	0.11	0	181344	0.18
1	314313	0.40	1	213298	0.21
2	220320	0.28	2	195571	0.20
3	112605	0.14	3	154221	0.15
4	39308	0.05	4	106856	0.11
5	9549	0.01	5	68048	0.07

Table 8: Counts and relative frequencies of preposition amounts per main phrase.

	Flickr	**NYT**
Average:	1.67	2.40

Table 9: Average number of prepositional phrases per main phrase.

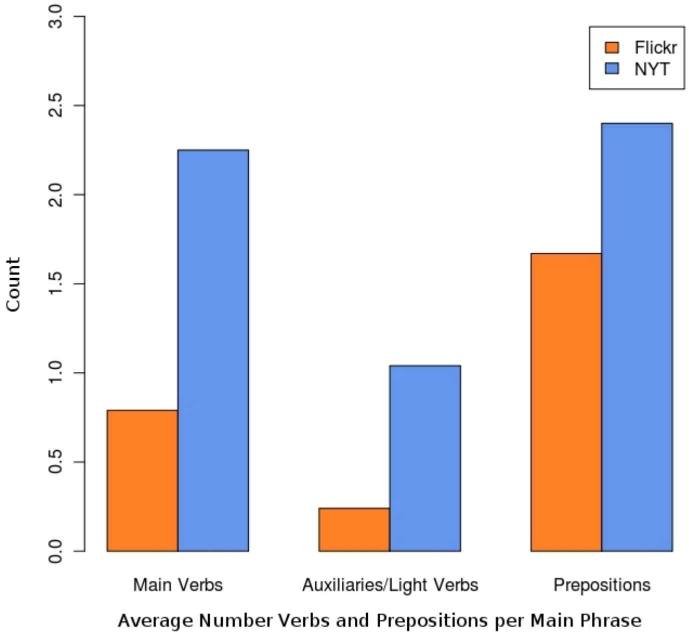


Figure 1: Average number of light verbs, main verbs, and prepositions per main phrase.

	Flickr	**NYT**
Physical objects:	0.63	0.31
Abstract objects:	0.23	0.51
Unk objects:	0.13	0.18

Table 10: Relative frequency of object types in Flickr and NYT (using WordNet).

3.1.6 Ordering of Nominals

One question that arises when generating descriptions of visual scenes is how to order the objects that are described. Saying something like *the portrait above the couch underneath a girl* sounds

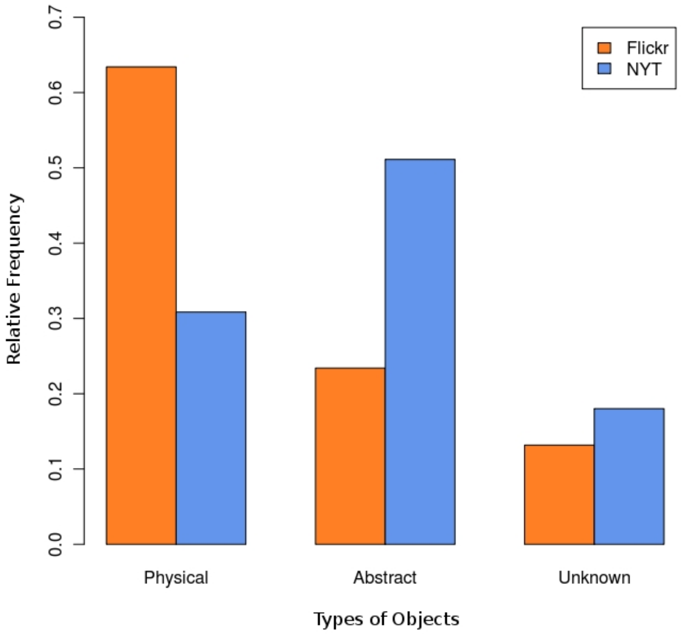


Figure 2: Relative frequency of object types in Flickr and NYT (using WordNet).

Flickr			NYT		
num	**count**	**prob.**	**num**	**count**	**prob.**
0	13797	0.02			
1	84452	0.11	1	52362	0.05
2	210350	0.27	2	96395	0.10
3	190543	0.24	3	125196	0.13
4	139373	0.18	4	134779	0.14
5	85339	0.11	5	129678	0.13
6	41089	0.05	6	115475	0.12
7	14386	0.02	7	96812	0.10
8	3581	0.01	8	75965	0.08

Table 11: Counts and relative frequencies of object amounts per main phrase.

Flickr			NYT		
num	**count**	**prob.**	**num**	**count**	**prob.**
0	98787	0.13	0	253751	0.26
1	211439	0.27	1	298242	0.30
2	273155	0.35	2	211092	0.21
3	136664	0.17	3	121107	0.12
4	48695	0.06	4	60752	0.06
5	12218	0.02	5	27640	0.03

Table 12: Counts and relative frequencies of physical object amounts per main phrase.

less natural than something like *the girl on a couch, underneath the portrait*. Further, such awkward structuring of descriptions can lead to garden-path sentence constructions or false conversational implicature [37], where the reader derives an unintended meaning from the description.

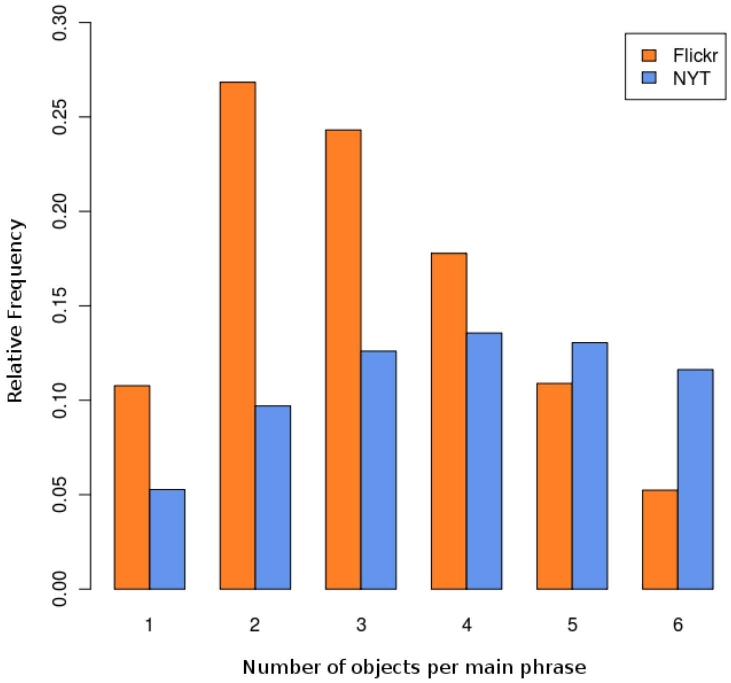


Figure 3: Relative frequencies of object amounts per main phrase.

Flickr		NYT	
Physical	**Abstract**	**Physical**	**Abstract**
0.29	0.22	0.19	0.18

Table 13: Semantic class coverage of physical and abstract objects.

Flickr			NYT		
num	**count**	**prob.**	**num**	**count**	**prob.**
0	880243	0.61	0	1113820	0.69
1	424782	0.30	1	364147	0.23
2	103727	0.07	2	88328	0.06
3	24278	0.02	3	30942	0.02
4	5690	0.00	4	7974	0.01
5	1535	0.00	5	1562	0.00

Table 14: Counts and relative frequencies of modifier amounts per physical object.

One key component of generating descriptions of images, then, is to solve where each item should appear in the description. This in turn guides the kinds of spatial and verbal relationships that will be generated between the objects, discussed in Section 5.1. With noun positions defined, the rest of the sentence can be fleshed out in light of these positions; this makes it more probable to generate something like *the man holding the umbrella* than *the umbrella above the man*.

We take a data-driven approach to this problem. Using the Flickr corpus, we isolate all noun phrases in each main phrase, and within each noun phrase, extract the head noun. For each head noun, we collect all hypernyms $h_1...h_i \in H$ of that noun listed in WordNet, the position pos of the head noun relative to the other head nouns in the main phrase, and the number of head nouns N in the sentence. The probability o of a noun position given its hypernym and the number of nouns in the sentence is

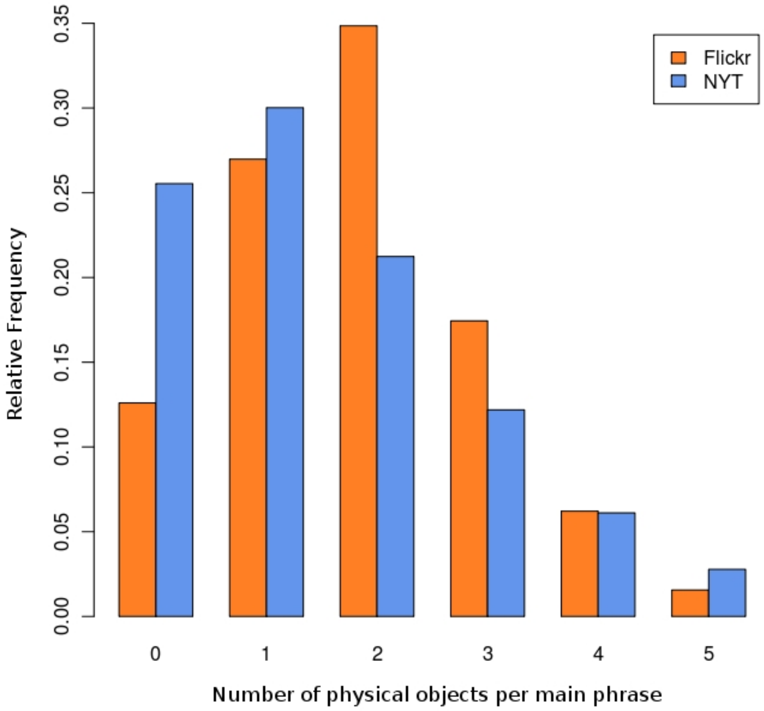


Figure 4: Relative frequencies of physical objects per main phrase.

Flickr			**NYT**		
attribute	**count**	**prob.**	**attribute**	**count**	**prob.**
direction	9325	0.04	direction	14195	0.05
beauty	26650	0.10	beauty	13213	0.04
color	123731	0.47	color	14866	0.05
pattern	3565	0.01	pattern	3342	0.01
age	8503	0.03	age	38502	0.13
material	15958	0.06	material	5516	0.02
surface	8586	0.03	surface	6360	0.02
shape	2974	0.01	shape	3574	0.01
quality	12999	0.05	quality	7820	0.03
ethnicity	7640	0.03	ethnicity	52197	0.17
size	44316	0.17	size	148334	0.48

Table 15: Coverage of semantic attribute classes.

p($pos|h_i, N$) for all $h_i \in H$, for all observed nouns. These probabilities are output as a database file where each line is a vector $<N, pos, h_i, o>$.

Based on these statistics, clear trends emerge. One trend is that animate things tend to be mentioned closer to the beginning of the sentence than inanimate things (see Figure 7 below). Animals tend to be mentioned closer to the front of the sentence, while structures tend to be mentioned closer to the end of the sentence (Figure 9). Given a set of nouns such as *box*, *cat*, and *light*, the model can predict that an ordering like *cat*, *box*, and *light* will be most probable, which makes it possible to generate phrases such as *A cat in a box contemplates the light*. In general, although we see similar trends in the NYT data, the trends are not as strong.

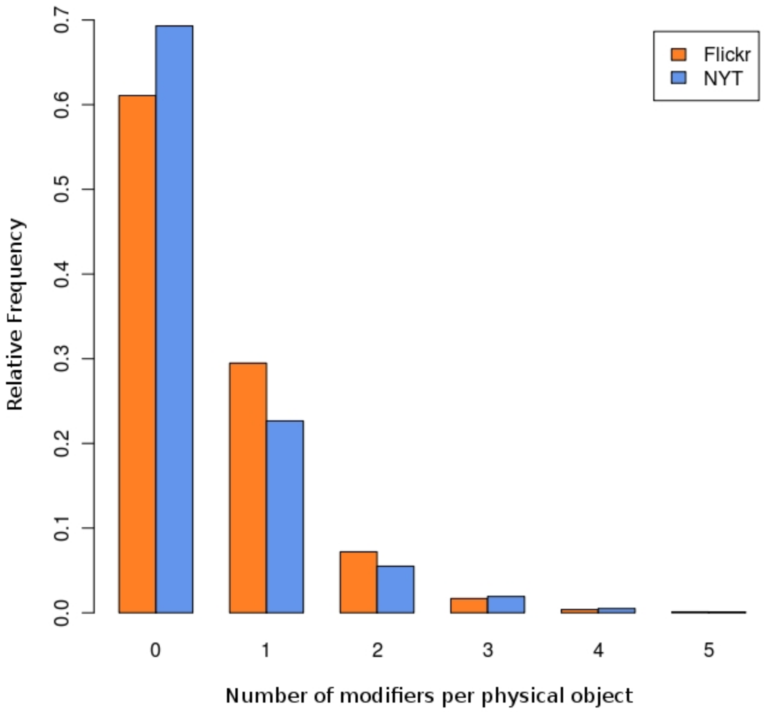


Figure 5: Relative frequencies of modifiers per physical object.

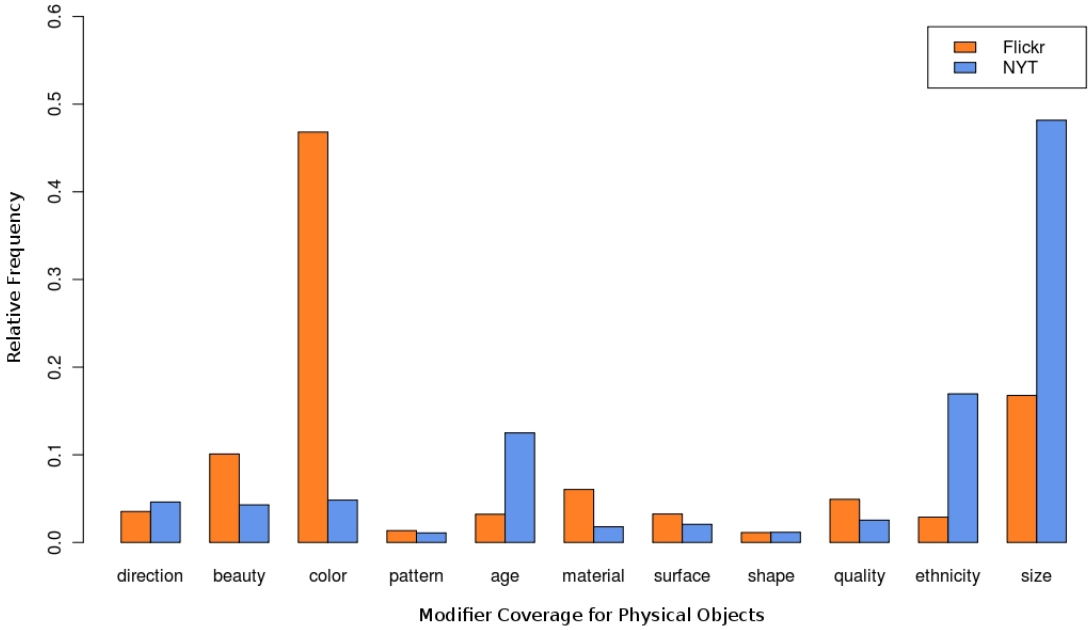


Figure 6: Coverage of semantic attribute classes.

N	pos	hypernym	Flickr	NYT
3	1	physical_entity	0.35	0.38
3	2	physical_entity	0.37	0.32
3	3	physical_entity	0.28	0.30

Table 16: Likelihood of the *physical entity* hypernym in each position of 3-noun main phrases.

N	pos	hypernym	Flickr	NYT
3	1	abstraction	0.18	0.17
3	2	abstraction	0.41	0.36
3	3	abstraction	0.41	0.47

Table 17: Likelihood of the *abstraction* hypernym in each position of 3-noun main phrases.

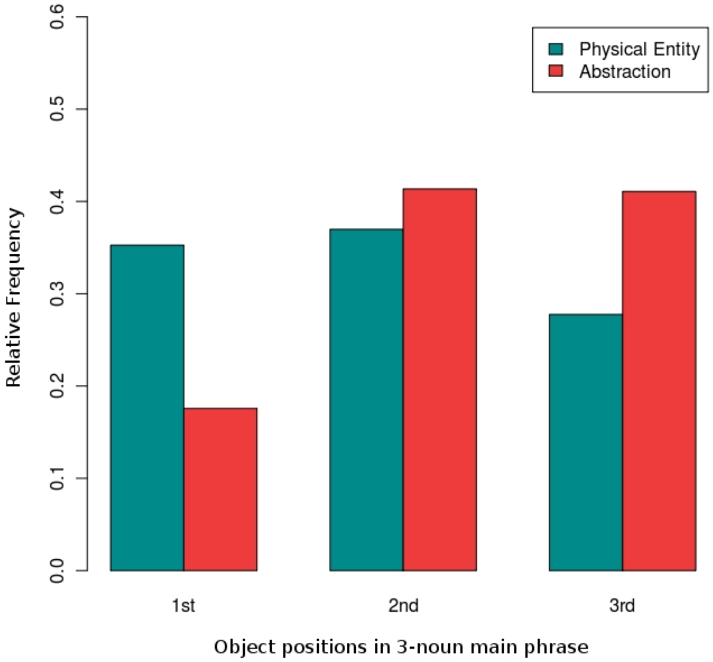


Figure 7: Likelihood of physical and abstract objects for 3-noun main phrases.

N	pos	hypernym	Flickr	NYT
3	1	animal	0.56	0.32
3	2	animal	0.32	0.36
3	3	animal	0.12	0.33

Table 18: Likelihood of the *animal* hypernym in each position of 3-noun main phrases.

N	pos	hypernym	Flickr	NYT
3	1	structure	0.27	0.22
3	2	structure	0.41	0.37
3	3	structure	0.32	0.41

Table 19: Likelihood of the *structure* hypernym in each position of 3-noun main phrases.

3.1.7 The Grammatical Structure of Visually Descriptive Text

One of our main hypotheses was that the writing style of visually descriptive text is distinct from that of non-descriptive text. In trying to verify this claim, we extracted the probabilistic context-free grammar (PCFG) from the parse trees of the free-text descriptions in the ImageCLEF dataset. As expected, the grammatical structure is heavily biased. The top 20 percent of the rules are responsible for over 50 percent of the sentences (Figure 1).

It is interesting to note that the most frequent structure is

(S (VP (NP (DT) (JJ) (NN)) (PP (IN) (NP (DT) (NN)))))

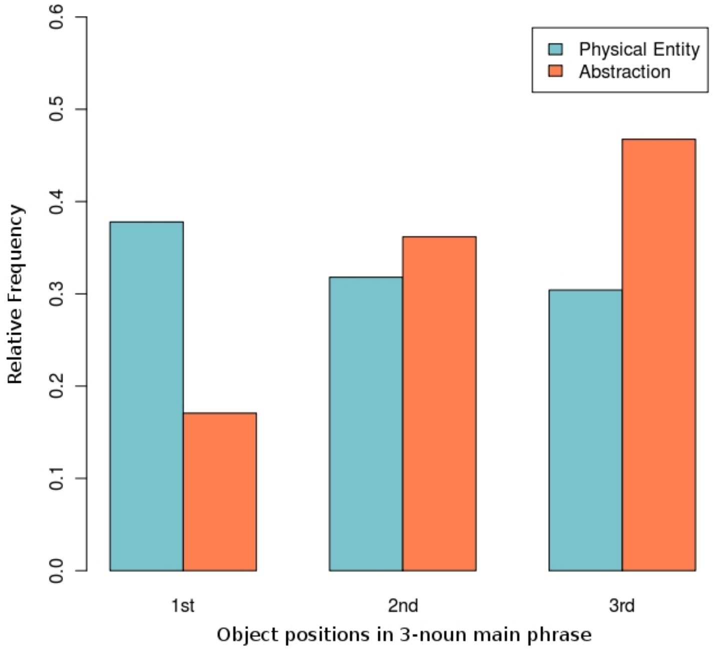


Figure 8: Likelihood of physical and abstract objects for 3-noun main phrases in NYT data.

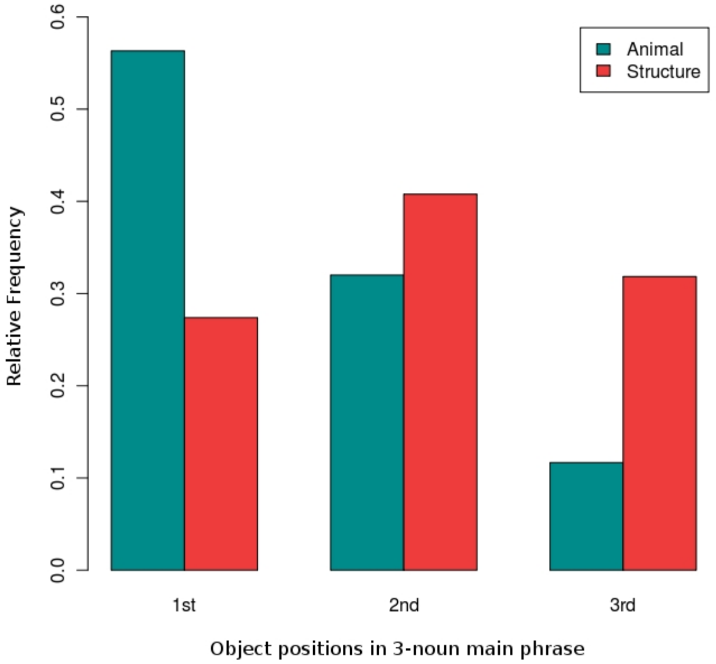


Figure 9: Likelihood of animal and structure positions for 3-noun main phrases in Flickr data.

bearing probability 0.05 (out of 21383 rules). This characterizes incomplete sentences (a single noun phrase) predominantly used for quickly delineating the content of an image, and the parser's attempt to interpret it (as a verb phrase). A typical example of such structure is

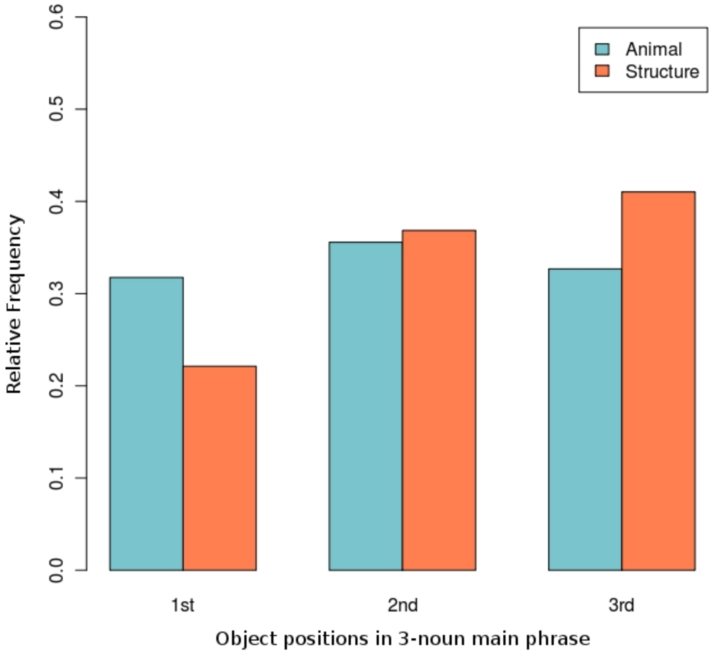


Figure 10: Likelihood of animal and structure positions for 3-noun main phrases in NYT data.

"A round table in the kitchen."

Top five structures that occupy much of the probability mass are shown in Table 1.

The extraction of the PCFG was initially motivated by the hope for data-driven language generation, where this distinctive style of descriptive text will allow for similarly colorful and appropriate language. Given a set of specified subtrees $\mathcal{S}$ (e.g., detected objects), we can build a complete tree in a recursive, bottom-up fashion using the PCFG. At each stage of the ascent, we choose a parent node that has the head nodes of those subtrees as children. The parent has to be the left-hand side (LHS) of the rule R that includes the children in its right-hand side (RHS), with maximum $P(R|\mathcal{S} \subset RHS(R))$. This can be done by simply picking the most heavily weighed rule that has the right children.

In order to tune the branching factor of the parents, we define a linear function that scores the rule by the precision and recall of the rule's RHS with respect to $\mathcal{S}$:

$$\begin{aligned} score(R) &= \alpha \cdot P(R|\mathcal{S} \subset RHS(R)) \\ &+ \beta \cdot precision(RHS(R), \mathcal{S}) \\ &+ \gamma \cdot recall(RHS(R), \mathcal{S}), \end{aligned}$$

The values α, β, γ control the importance of these features. Using high β gives a more succinct sentence. In this way, we generate a pool of candidate sentences, and choose the best using both the tree probability (composite product of the probabilities of the rules used in the tree) and the N-gram model trained on the same dataset (ImageCLEF).

However, the main drawback of this method is that we have no deterministic control over generation due to the fact that the search space is ill-defined (a tree can be arbitrarily deep), although this very lack of predictable behavior was an attempt to generate "creative" descriptions. It may be possible to enforce more syntactic and semantic constraints so that the space is well-defined for principled optimization. Nevertheless, as it stood, it generated sentences either wildly hallucinated (e.g., "crystals fallen over claustrofobic old girl" for "girl") or trivially glued together (e.g., "a girl in the beach in sun" for "a girl", "the beach", and "in sun"), depending on the choice of α, β, γ.

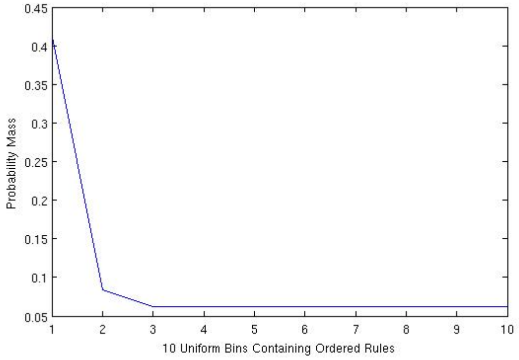


Figure 11: The probability mass of the rules in 10 uniform bins.

Structure	Prob
(VP (NP (DT) (JJ) (NN)) (PP (IN) (NP (DT) (NN))))	0.050
(NP (NP (DT) (JJ) (JJ) (NN)) (PP (IN) (NP (DT) (NN))))	0.028
(NP (NP (JJ) (NNS)) (PP (IN) (NP (DT) (NN))))	0.011
(NP (NP (JJ) (NNS)) (PP (IN) (NP (NP (DT) (JJ) (NN)) (PP (IN) (NP (DT) (NN))))))	0.008
(NP (NP (DT) (NN)) (PP (IN) (NP (DT) (NN))))	0.005

Table 20: Top five structures in the ImageCLEF.

3.2 Studying What People Describe in Images

In this section we explore what aspects of images, *e.g.* which objects, attributes, or scene terms, are used in descriptions. We begin by considering factors that determine what is mentioned, and then move on to consider what words refer to what aspects of image content.

3.2.1 Modeling Description Factors – Karl Stratos Columbia University

Consider the picture in Figure 2. We see many things—floor, chair, wall, tree, person, and so on. However, when asked to briefly describe what we see in the image, we will not say "I see three people, seven chairs, one floor, one bag, ...". Rather, we will choose a few objects that we think matter the most, and describe them, as in "I see some people" or "Viewing people in chairs between walls". The question we want to answer is, given an image, what factors influence our decision to describe certain objects in it, but not others?

We automatically learn these descriptive factors by exploiting the dataset ImageCLEF [39, 21], and incorporate them in a model that captures *what* to describe, which can potentially be used to avoid the enumeration of all detections. Various models are explored, although the performance evaluation is tough because the dataset is constructed arguably without a clear pattern. It should be emphasized, however, that the method here can in principle be applied to any similar dataset if such one is available, and thus it provides a general framework for further pursuit of the subject.

3.2.2 Previous Work

The work by [73] is probably the closest to ours in spirit. Given an object O in a particular image I, they define the object's importance in the image as:

$$Importance(O|I) = P(O_1 = O|[O_1, \ldots, O_n]),$$

where $[O_1, \ldots, O_n]$ is an ordered list of objects in I mentioned by a viewer. They build a generative model (the "Urn") that also takes account of the missing descriptions, and fit it to data collected from Mechanical Turkers by MLE. They justify the complexity of their method by showing that it does better than the mere frequency of words.

However, note that this method does not explicitly tell us *what* objects to mention, other than the indirect pruning based on the importance score. For our task, it is far more natural to directly observe

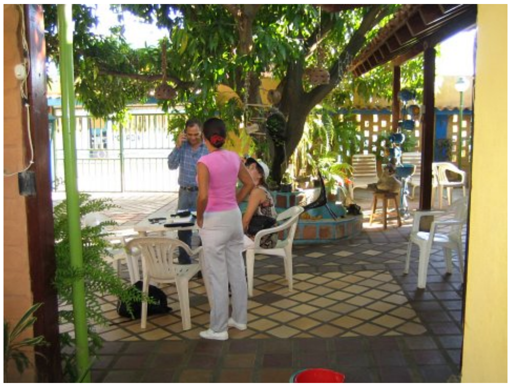


Figure 12: A picture with varied content.

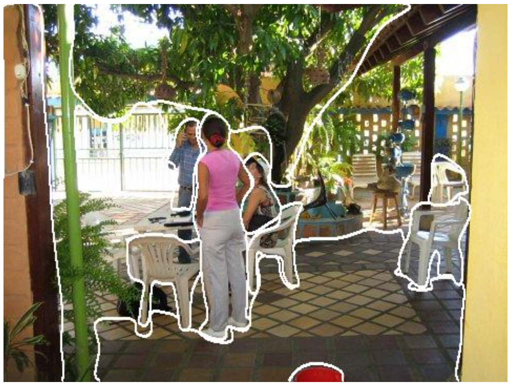


Figure 13: Labels: ‘door’, ‘floor’, ‘wall’, ‘tree’, ‘person’, ‘chair’; Descriptions: “two women and a man are standing and sitting in a yard on white chairs around a white table in the foreground”

the probability

$$P(O \text{ is mentioned}|O \text{ is an object in } I).$$

This way, there is no need for hand annotation of ordered lists of objects, which is both costly and artificial. The limitation of the mere word frequency can be alleviated by delving more semantic features, such as the type, size, and location of O.

Clearly, this cannot be done without knowing (1) all objects contained in I, (2) whether O is mentioned or not, and (3) O’s semantics. The difficulty of having access to all this information was probably part of Spain and Perona’s roundabout method.

3.2.3 Exploiting ImageCLEF to Estimate Description Factors

In ImageCLEF, images are both tagged with both free-text descriptions and segmented into regions based on a small set (275) of labels. Therefore, given an image, we roughly know (1,3) what things are present and their semantic information by looking at the segmented label objects, and (2) which of them are mentioned by looking at the descriptions. Figure 3 shows the information we have in ImageCLEF for the picture shown in Figure 2. In general, there can be more than one description per image.

We must make it clear that ImageCLEF has many problems for our purpose. The labels (which form a hierarchy on their own) are very unpredictable and inconsistent. For instance, in an image of a couple, we might have either {‘man’, ‘woman’}, ‘two-persons’, or ‘group-of-persons’ as the choice of label for no particular reason; in an image of a lion, we might have ‘lion’, ‘animal’, or ‘entity’. We systematically collapsed the label hierarchy by leaving labels that are exact matches of our detector types, and grouping the rest under the appropriate labels when possible. As a result,

Threshold	0.8	0.85	0.9	0.95
F1	0.89	0.91	0.94	0.91

Table 21: The F1 scores for mapping with different thresholds.

Top5	Freq	Last5	Freq
firework	0.85	sidewalk	0.06
crab	0.83	tire	0.05
coral	0.82	smoke	0.05
lion	0.81	fabric	0.04
whale	0.71	instrument	0.04

Table 22: The frequency of words in the sampling.

	Mention	Present	Freq
Animate	10215	17292	0.59
Inanimate	35408	117861	0.3

Table 23: Animate v.s. Inanimate objects.

the original 275 labels are reduced to 143. Also, we removed the duplicates for simplicity, since we concentrate on the *presence* of an object. In this way, we now have just one ‘person’ label for the first example. The labels shown in Figure 3 is the result of collapsing the original set: {‘tree’, ‘floor’, ‘chair’, ‘chair’, ‘chair’, ‘chair’, ‘woman’, ‘man’, ‘woman’, ‘door’, ‘wall’}.

The free-text descriptions are also problematic in that they are mostly dictated by a small group of people participating in the project. Although they are relatively clean in spelling and grammar, the scope of their structures is inevitably small. However, despite this heavy bias, the dataset still manages to shed light on what a human may choose to describe in an image, as shown in the reasonable statistics below.

3.2.4 Method

Assuming the labels are all observable objects in the image, we wish to know which ones are mentioned.[1] To do so, we look at all nouns in the description, lemmatized. For instance, we would have ‘woman’, ‘man’, ‘yard’, ‘chair’, ‘table’, and ‘foreground’ for Figure 2. Then for each of these nouns, we try to determine if it refers to one of the labels.

Various techniques of doing this word mapping have been examined, such as computing word similarity based on the frequency of dependency relations in large corpora. In the end, a simple similarity metric based on the WordNet hierarchy was deemed the most suitable for several reasons. First, its usual shortcoming of limited domain of words does not affect us, because we have only a closed set of labels (all of which are covered in the hierarchy). Second, the similarity computation based on the edge counting method (Wu-Palmer in our case) does a good job of capturing sibling concepts (e.g., ‘ship’ and ‘boat’). Most important, the hierarchy allows us never to miss a mapping instance when one term is an ancestor of another (e.g., ‘cyclist’ and ‘person’). The threshold value for the mapping was tuned by manually estimating the F1 score on 30 randomly sampled images (Table 2).

After we have the mapping, we can simply count to yield the probabilities

$$\begin{aligned}
& P(O \text{ is mentioned}|O \text{ is an object in } I) \\
\approx \quad & P(L \text{ is referred to}|L \text{ is a label of } I) \\
= \quad & C(\text{L label referred})/C(\text{L label}).
\end{aligned}$$

Table 3 shows the first 5 and last 5 labels when sorted in frequency. Note that unusual objects tend to be highly likely to be mentioned; we don’t see fireworks or corals so often. In contrast, low-frequency objects tend to be very common (sidewalk, tire) or too generic (smoke).

[1] This assumption is clearly too strong. We have no ‘table’ as a label in Figure 2, even though it is referred to in the description. But this does not affect our way of computing the conditional probability.

Figure 14: The bigger the object's relative size is, the more likely it is mentioned.

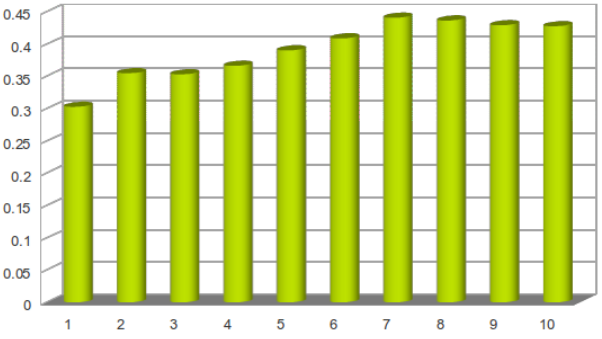
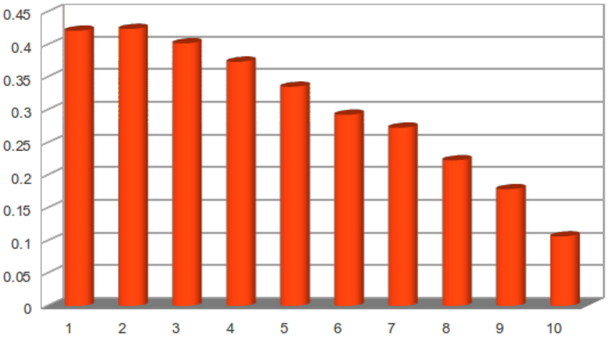


Figure 15: The further away from the center the object is, the less likely it is mentioned.



It is easy to consider deeper discriminating semantics. Table 4 shows the statistics of the labels when partitioned into animate and inanimate categories. We include all the non-plant life forms under the animate, and the rest under inanimate. Despite the fact that animate objects occupy a small portion of the label occurrence space, they are more often noted when present, giving a frequency of 0.59 that is much higher than 0.3 of inanimate.

Since we have the pixel coordinates of the segmented labels, we can look at the statistics of their influence as descriptive factors. As the images have different sizes, they must be normalized. Given an object O in image I, let $pix(O) = \{(x, y)|(x, y)$ is a pixel of O in $I\}$. We define the relative size of O as

$$\frac{|pix(O)|}{C(I\text{'s pixels})},$$

and the relative location (from the center) as

$$\frac{||center(I) - COM(O)||}{||center(I)||}$$

where $COM(O) = (\sum_{(x,y)} pix(O))/|pix(O)|$ is the center of mass of O, and $||\cdot||$ is the Euclidean norm. Figure 3 and 4 show the effect of size and location (put into 10 uniform bins) on the likelihood of being mentioned. Clearly, an object is more frequently described when its size is bigger and its location is nearer to the center of the image. The plateau that arises in the last few bins in size is probably due to the fact that many of the less salient objects occupy much space, e.g., 'sky' or 'floor'. However, the pattern holds true in general.

3.2.5 Modeling the Description Decision Process

Ultimately, we want to use the description factors that we have explored as features in discriminating what to mention in an image. Clearly, this is an instance of binary classification. If $\Phi(O) \in \mathbb{R}^n$ is

	Features	Accuracy
BS("Yes")		0.57
Model 1	T	0.68
Perceptron	T+S+L+A	0.59
NB	T+S+L+A	0.69
SVM	S+L	0.60
SVM	T	0.68
SVM	T+S+L	0.69
SVM	T+S+L+A	0.69

Table 24: Accuracy and F1 score for various models, T for type, S for size, L for location, A for animacy.

an n-feature representation of an object O, the input and output domains X and Y are

$$\begin{aligned} X &= \{\Phi(O) : O \text{ is an object in the current image}\} \\ Y &= \{+1, -1\} \end{aligned}$$

(+1 means "Mention O" and -1 means "Do not mention O"). We assume there is some true distribution $D : X \times Y \to [0, 1]$ generating both training and test examples, and try to come up with a hypothesis $h : X \to Y$ that minimizes the error on the data.

We divided the dataset into three parts: training (80%), development (10%), and test (10%). As acknowledged, evaluating with ImageCLEF is not an easy task because the descriptions are arguably unnatural (e.g., some only mention a chair in an image that contains people and tables). So the assumption that there is a true underlying distribution for the data may be unreasonable.

We first look at the simplest of the baselines: giving the same answer to every example. An important point here is that the result differs greatly depending on whether we consider *all* descriptions as referring to the image as a single instance, or we consider *each* as a single instance. In the former, the dataset is heavily tilted towards the negative answer. Therefore, if we say "No" to every instance, we get an accuracy of 0.66. On the other hand, in the latter, the dataset is tilted towards the positive answer. If we say "Yes" to every instance, we get an accuracy of 0.57. We decided to choose the latter, because some descriptions were clearly meant to accompany the previous ones (e.g., "And there are also...").

We can do significantly better by simply using the label type probability (call this Model 1). That is, given a label, say "No" when its probability is less than 0.5, and say "Yes" otherwise. We get an accuracy of 0.68.

Now we define the feature representation of a label. There are 143 types of label, and the relative size and location is a real number in $[0, 1]$. So it is natural to have a 145-long feature vector where each type is a binary feature (0 or 1), and the last two features are size and location values. Concretely, for an object O and indicator functions I_n for the n^{th} label type,

$$\Phi(O) = (I_1(O), \ldots, I_{143}(O), size(O), loc(O)).$$

We can put another indicator for animacy, and compare the performance.

A basic perceptron classifier with threshold 0, a Naive Bayes (NB) classifier (for which the size and location values were put into discrete bins), and a linear SVM using the LIBSVM package have been employed. These scores are summarized in Table 5. The fact that a model as simple as an NB can compete with an SVM supports the lack of coherent distribution; the factors may well be conditionally independent given the description decision. Also, the animacy feature turns out to be not quite useful for discrimination, as it provides not much novel information about the type feature that we already know.

3.2.6 Scenes and Attributes in Descriptions – Alyssa Mensch, MIT

When people describe a given scene or an image, there are certain objects or scene names that they might tend to say first. For example, people might see a picture of sand and the ocean and describe it as a beach. I've been looking into which sorts of things people mention when presented with an image, and seeing what are the factors that influence whether people will describe a certain aspect of an image, both by looking at data from annotated image datasets and probing people on Mechanical Turk. The factors I looked at were the "scene-ness" of an image, the presence of people in an image, and unusualness. To see the effect of the "scene-ness" of an image, I took 10 images from the SUN09 database, half of which were typical "scene" images and half of which did not represent a coherent scene that could be described using a scene word or phrase (e.g., "beach" or "living room"). The "scene-ness" of the images was determined based on human judgment. These 10 images I put on Mechanical Turk and had nine workers describe each image. In order not to bias the Mechanical Turk workers towards a particular type of description, the instructions were simply to describe the image: no example or further restrictions on the description were given. This may have led to a bias towards shorter descriptions: since Mechanical Turk workers are paid for each Human Intelligence Task (HIT) they complete, independent of the length of their response, it is financially in their interest to write shorter descriptions, completing each HIT more quickly so as to finish more HITs in a given amount of time. However, hinting at a certain type of description might make workers more likely to describe certain objects than others, or to include certain adjectives, and we were interested in the default descriptions that human annotators will give.

In images where there is no coherent scene, it was not possible for Mechanical Turkers to assign a scene word to the image, since none exists. Therefore, it would be nonsensical to compare the frequency of scene words for the scene vs. non-scene images. A better measure for comparison would be in how frequently the background of the image is described in non-scene images with how frequently the scene is mentioned in scene images. Effectively this is a comparison of how often the parts of the image that are not the objects are mentioned.

The results show that people are likely to mention a scene if the image depicts a scene, and are unlikely to mention the background if it does not depict a scene:

$$\begin{aligned} \mathbf{P}(\text{mention bkgd}|\text{non-scene}) : &\quad 0.241 \\ \mathbf{P}(\text{mention scene}|\text{scene}) : &\quad 0.861 \\ \mathbf{P}(\text{enumerate objects}|\text{scene}) : &\quad 0.5 \end{aligned}$$

I also looked at co-occurrences of object labels and scene names in the SUN09 database. All the images in the database have been segmented into labeled objects, and each image is classified into the sort of scene it belongs to.

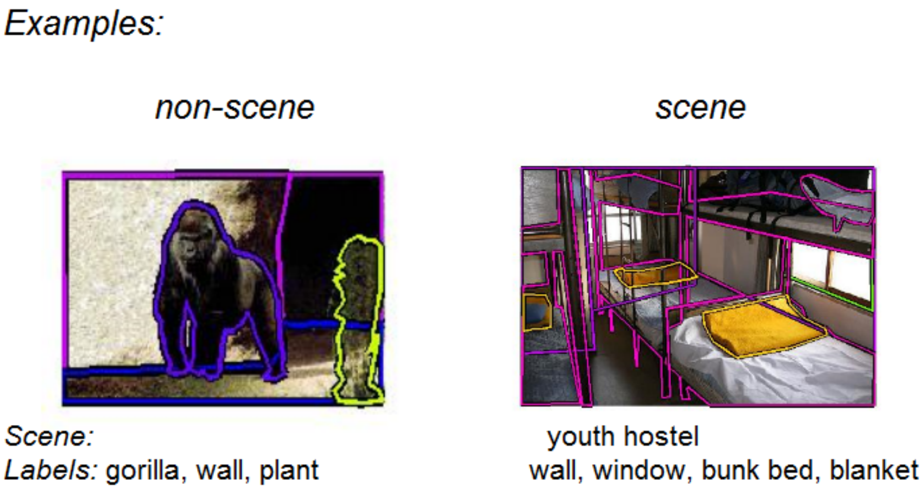


Figure 16

When I looked at the co-occurrence matrix of labels and scenes, there seemed to be a pattern where there were more labels beginning with the same letter as the scene name. This could be explained by images whose scene is "zoo" being more likely to have "zoo" objects in them. However, when I looked only at the most frequently occurring labels, this particular pattern between labels and scenes did not appear. Certain labels, such as "person," occurred in many different scenes.

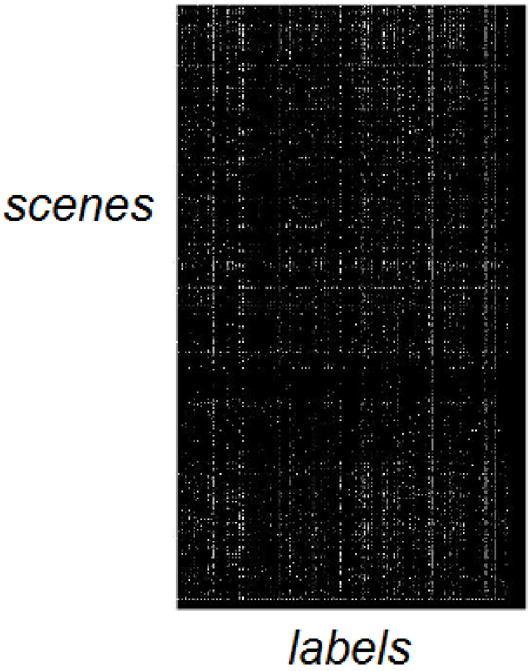


Figure 17

It did appear that certain unusual objects were labeled most frequently in the database, such as "window" and "wall," which one would not intuitively think of as the first thing described in an image. This could be explained by a labeling bias by which researchers specifically segment and label objects they are interested in.

wall	10276
window	8929
sky	5309
building	5044
trees	4520
floor	3962
tree	3558
ceiling lamp	3408

I also looked at the effect of unusualness on peoples descriptions of images. To do this, I took 413 images of trees from a data set of 708,000 flickr images and asked Mechanical Turk workers to give up to five colors to describe the trees in the images. The images were chosen by taking the first few hundred of images in the dataset where the word "tree" was in the flickr caption and the tree-detector fired for the image; these were then manually sorted through to find images that did in fact have trees in them. (Images that did not have trees in them might be of an object near or under a tree, where the tree itself was not included in the image, and the detector fired by mistake.) I then compared the colors used by Mechanical Turkers to describe the images to the colors in the flickr descriptions to see if there are certain contexts in which people are more likely to describe trees using colors. The hypothesis was that if a tree color is unusual (e.g., red), people will be more likely to describe the tree using the tree color, whereas if a tree color is usual (e.g., green), people will be less likely to describe the tree with that color.

Results: The Mechanical Turk results confirmed our hypothesis that people are less likely to include a typical tree-color in their description of a tree. Specifically, the probability that there is no tree-color in the flickr description given that the tree is green is 0.366, higher than the probability for any of the other possible tree colors. Furthermore, the results confirm that green is the default color for trees: given that a tree is not described using a color in an image's flickr caption, it is more likely that the tree color is green than that it is any other color.

none	violet	0.001	none	purple	0.005	none	blue * 0.006	
none	orange * 0.013	none	grey	0.020	none	pink	0.028	
none	gray	0.035	none	red	0.037	none	white	0.046
none	yellow	0.049	none	black	0.097	none	brown	0.248
none	green	0.0366	yellow	green	1.0	pink	pink	0.125
pink	brown	0.25	pink	green	0.5	green	green	0.167
green	white	0.167	green	brown	0.333	green	red	0.333
white	yellow	1.0	red	black	0.1	red	grey	0.1
red	yellow	0.1	red	brown	0.3	red	green	0.4
blue	yellow	0.333	blue	green	0.667	gray	green	1.0

The next probing experiment I ran on Mechanical Turk was to see how unusualness affects description of images in the case of people, shirts and hats. The goal was to determine how commonly hats and shirts occur in an picture of a man in the flickr database, and to determine how likely each is to be mentioned, given that it is present in the image. The hypothesis was that the more unusual article of clothing would be more likely to be mentioned if it was present: i.e., hats would be mentioned when they were present at a higher rate than shirts, even though there would be more shirts in the database overall.

To test this hypothesis, I searched for flickr captions with the word "man" in them. I then took the 3009 image results and created two sets of Mechanical Turk HITs with all 3009 images. In the first of these sets, Mechanical Turk workers were instructed to categorize each image as having a man wearing a hat, a man not wearing a hat, or no man at all; the second set of HITs asked the same questions, but for shirts instead of hats.

We determine that an image shows a man with a shirt or hat if at least two of the three Mechanical Turk workers labelled the image that way. For most images, there were three responses; however, occasionally an image was not labelled by one of the Mechanical Turkers, though it was labelled by the other two. There were 21 unlabelled images from the hat HITs, and 18 unlabelled images from the shirt HITs (out of a total of 3009 images). For these images, if the other two Mechanical Turkers labelled the image as showing a shirt or hat, we accept this as the ground truth.

Of the 828 flickr images labelled by Mechanical Turk workers as showing a man wearing a hat, 178 of the corresponding captions contained the word "hat." By comparison, of the 1892 images labelled by Mechanical Turkers as showing a man wearing a shirt, 183 of the corresponding captions contained the word "man." Thus the probability of a hat being mentioned given that a man is mentioned in the caption, and there is a man wearing a hat in the image, is 21.5

These results seem to support the hypothesis that people are more likely to mention unusual things in an image: it is more typical to find a man wearing a shirt than to find a man wearing a hat in reality, but hats are more likely to be mentioned in flickr captions if they are present than shirts are if they are present. Furthermore, images of a man wearing a shirt are more common in the flickr corpus than images of a man wearing a hat, which is as we would expect, given that this is more common in reality, as well.

It is possible that other factors biased these results. The word "hat" was one of the search terms used to get images from flickr to include in the database. This could explain why shirts are nearly half as likely as shirts, whereas the ratio in real life seems much lower (there are a lot fewer people wearing hats than wearing shirts). This fact might influence the probability that a hat is mentioned given that it is present in an image: there might be a correlation between an image having the tag "hat" and the word "hat" being present in the image's caption, since the person who posted the image wrote both the tags and the caption.

Future Work: Future experiments could repeat the work described above, except in ways that would reduce noise in the data and might lead to better results. One such experiment would be to a repeat of the tree-color experiment, only this time further restricting the number of color descriptions allowed. Previously we allowed up to five colors, though in the flickr descriptions there were nearly always zero, one or two colors per tree description. Restricting the number of colors to two or one could give a better estimation of the colors flickr users would use in their captions. Another change to this experiment would be to restrict the colors to a small set of colors that were present in the flickr data. In the original HIT results, one of the Mechanical Turkers appeared to have used the Wikipedia list of colors in describing the trees, making it difficult to compare to the flickr colors. Restricting to usual colors (e.g., "red") and disallowing unusual colors (e.g., "halaya ube") would better allow us to compare default colors.

Another experiment to repeat would be that comparing descriptions of hats and shirts, but with some accessory or article of clothing other than hats, as "hat" was one of the search terms used to create the database of flickr images. One alternative might be watches, though these might be difficult to identify in a photograph.

3.2.7 Learning semantic categories of Attributes – Amit Goyal, University of Maryland with help from Jesse Dodge, University of Washington

In general, objects are defined using a set of attribute values. For example, "apple" can be defined using its *color i.e red*, *shape i.e. round*, and *size i.e. small*. In this work, we define 11 attribute classes (just focusing on adjectives) which can be used to define or describe visual aspects of objects. For each of the 11 attribute classes, we manually assign 5 seeds or values to each class. The attribute classes and their seed values are shown in Table 31. From these set of 5 seeds for each class, our goal is to automatically learn more values for each attribute class.

These classes are important for recognizing attributes of object from images. Once, we know objects have certain classes of attributes associated with them. We can train attribute classifiers [47, 24] for an object in a greedy fashion by using most common attribute values first. For example, for "apple", for color, we train for "red" value and for shape, we train for "round" shape. Learning semantic categories of attributes is quintessential for natural language generation of captions of images. Once, we know what attribute values are associated with objects, we can generate more interesting, verbose, and natural captions [88].

3.2.8 Graph Construction

We draw a graph between adjectives by computing distributional similarity [79] between them. For computing distributional similarity between adjectives, each target adjective is defined as a vector of nouns which are modified by the target adjective. To be more precision oriented, we use only those adjectives as modifiers which appear adjacent to a noun i.e. JJ NN construction[2]. For example, in "small red apple", we consider only *red* as a modifier for noun. Pointwise Mutual Information (PMI) [14] is used to weigh the context vector of nouns, and only top 1000 contexts (which are nouns) are selected for each adjective. We construct the graph using only top 50 distributional similar adjectives with respect to each target adjective. Cosine similarity is used to find similar adjectives. A similar restriction is used by [83] for constructing distributional similar graph for learning web-scale polarity lexicons.

3.2.9 Bootstrapping

Once the graph is constructed using adjectives as nodes and distributional similarity between adjectives as weights on the edges. We use bootstrapping to learn values for attribute classes. We use In-degree to compute the score for each node which have connections with known or already extracted reliable class nodes. The In-Degree (inDgr) score for node v is the sum of the weights of all incoming edges (u, v), where u is a trusted attribute class member. Intuitively, this captures the popularity of v among attribute values that have already been identified as good values. The idea of inDgr was exploited by [43] to learn homonym relations from the web. They also used other graph-based measures to weigh new extractions/values for each of the classes. However, there was not a substantial difference in using different measures, hence for this work, we only use inDgr as it is very simple and intuitive.

[2] Note: We can easily extend our framework to other noun modifiers using NN NN constructs and it also works, however for this work, we only focus on learning adjective classes.

Color:	purple	blue	maroon	beige	green
Material:	plastic	cotton	wooden	metallic	silver
Shape:	circular	square	round	rectangular	triangular
Size:	small	big	tiny	tall	huge
Surface:	coarse	smooth	furry	fluffy	rough
Direction:	sideways	north	upward	left	down
Pattern:	striped	dotted	checked	plaid	quilted
Quality:	shiny	rusty	dirty	burned	glittery
Beauty:	beautiful	cute	pretty	gorgeous	lovely
Age:	young	mature	immature	older	senior
Ethnicity:	french	asian	american	greek	hispanic

Table 25: Attribute Classes with their seed values

10;10	10;25	10;5	1;100
1;50	25;10	25;5	50;5
5; 25	5;50		

Table 26: 10 bootstrapped systems with different $iter$;m parameters.

In addition to using inDgr, we also make use of mutual exclusion principle by learning for all semantic classes at the same time. Each new instance learned can belong to only one class. This idea has been exploited in many other bootstrapping frameworks [75, 58] too. Moreover, after each iteration, we harmonically decrease the weightage of indegree associated with instances learned in later iterations compared to former ones.

There are two parameters to tune for this work. First, number of values/instances to be added after each iteration. Second, for how many iterations do we want to perform bootstrapping. These parameters are important, otherwise there is a high chance of having semantic drift [75, 59].

3.2.10 Data for studying context

We built a large corpus by concatenating the freely available Web-derived ukWaC corpus[3], a freely available 2009 dump of the English Wikipedia (http://en.wikipedia.org) and The New York Times Newswire Service (nyt) section of the Gigaword [35] Corpus. The Web-derived ukWaC and Web-derived ukWaC is already tokenized, POS-tagged with the TreeTagger [72]. The nyt corpus is tokenized and POS-tagged using TagChunk[4] [18]. The ukWaC and Wikipedia sections can be freely downloaded, with full annotation, from the ukWaC site.

3.2.11 Manual Evaluation of Attribute Classes

We conduct a manual evaluation to directly measure the quality of attribute classes. We recruited 3 annotators and developed annotation guidelines that instructed each recruiter to judge whether a learned value belongs to a attribute class or not. The annotators assigned “1“ if learned value belongs to a class, otherwise “0”. We provided the evaluators with the following guidelines:

- If a value belongs to a semantic class. For example, Color can be “red“, “green”, and “white”.
- If a value can be associated with a semantic class. For example, Color can be associated with “colorful“ and “multicolor”.

There are two free parameters to tune for this work. First, number of values/instances (m) to be added after each iteration ($iter$). Second, for how many iterations do we want to perform bootstrapping. To figure out which parameters work the best for bootstrapping, we build 10 systems ($iter$;m) with different $iter$ and m as shown in Table 26.

[3] http://wacky.sslmit.unibo.it/doku.php

[4] http://www.umiacs.umd.edu/~hal/TagChunk/

H1&H2	H2&H3	H1&H3
.45	.48	.48

Table 27: Inter Annotator Agreement (κ) over 4 (age, beauty, color, and direction.) semantic classes

H1&H2	H2&H3	H1&H3
.55	.59	.57

Table 28: Inter Annotator Agreement (κ) over 3 (ignoring age as it is very subjective class) semantic classes

We conduct an Information Retrieval (IR) Style Human Evaluation. Analogous to an IR evaluation, here the total number of relevant values for attribute classes can not be computed (In IR, we can not compute the total number of relevant documents with respect to a user-defined query.). Therefore, similar to an IR evaluation, we assume the correct output of several systems as the total recall which can be produced by any system. Now, with the help of our 3 manual annotators, we get the correct output of several systems from the total output produced by these systems.

First, we measure an agreement on whether learned value belongs to a semantic class or not. We computed κ to measure inter-annotator agreement for each pair of annotators. Currently, we have performed manual evaluation over only 4 classes: age, beauty, color, and direction. The results are shown in Table 27. Overall, the best κ score is around .48, which is probably low. However, if we evaluate them individually, we found that age has the lowest κ. This is intuitive, since age is one of the most subjective classes. Hence, we also report κ on only 3 semantic classes (excluding age). The results are reported in Table 28, and the best κ score is around .59, which is reasonable, as we did not define any hard annotation guidelines. Moreover, we also did not do an iterative process to improve the inter-annotator agreement between annotators.

Second, we compute Precision (Pr), Recall (Rec) and F-measure (F1) for all the systems for 3 semantic classes. The two system performed consistently (10;25 and 5;50) better than other systems, hence we report only result for these systems over 3 semantic classes by our 2 annotators. One of these annotators is more strict than the other, while labeling if a word belongs to a semantic class or not. "H1" is the more strict annotator, and hence our best systems get better recall than precision. However, for "H2" being less strict provides overall better F-score. Overall, results on the task of learning semantic classes is reasonable, as adjective are not as common as nouns. Moreover, there has been very less work done in learning adjective classes. Moreover, we are not aware of any work, which focuses on automatically learning semantic classes of adjectives with respect to helping vision with language world knowledge.

3.2.12 Learning if a word is visual or non-visual

We want to know if we can learn visual and non-visual words automatically. The task of finding if a word is visual or non-visual is different from learning if a word is concrete or non-concrete. It is true that in many cases, concrete words will be visual, and abstract words will be non-visual but this is not always true. For example, "party" is an abstract word but still is visual. In this work, we focus on learning only visual and non-visual nouns and adjectives.

To learn such classes, we draw a distributional similarity graph for both nouns and adjectives using "JJ NN" construction similar to Section 3.2.8. We construct the graph using only top 10 distributional similar adjectives or nouns with respect to each target adjective or noun. Adjectives are used as context for nouns and nouns as contexts for adjectives for graph construction.

System	Pr	Rec	F1
10;25	.53	.71	.60
5;50	.54	.72	.62

Table 29: Results on H2 annotations

System	Pr	Rec	F1
10;25	.46	.85	.59
5;50	.46	.84	.58

Table 30: Results on H1 annotations

Nouns		**Adjectives**	
Visual	Non Visual	Visual	Non Visual
car	idea	brown	public
house	bravery	green	original
tree	deceit	wooden	whole
horse	trust	shiny	initial
animal	dedication	rusty	total
man	anger	rectangular	personal
table	humour	furry	intrinsic
bottle	luck	striped	individual
woman	inflation	orange	political
computer	honesty	feathered	righteous

Table 31: Visual and Non visual seeds for nouns and adjectives

Once, we have such a graph, we can use bootstrapping from Section 3.2.9 to learn visual and non-visual nouns and adjectives. The seeds used for bootstrapping are shown in Table 31.

3.2.13 Evaluation

By manual inspection of the data, the learned classes look really clean. In future, we will like to do a mechanical turk evaluation to directly evaluate the visual and non-visual nouns and adjectives. For now, we show the coverage of these classes in Flicker data-set below:

- Visual Nouns: 64.23%
- Non Visual Nouns: 17.71

- Visual adjectives: 51.79%
- Non Visual adjectives: 14.40%

Overall, we find more visual nouns and adjectives cover flicker data-set, which makes sense, since flicker is a visually descriptive data-set.

Second, we show how many visual nouns are physical entities in word net and how many non-visual nouns are abstract entities in word net? The motivation behind this evaluation is that most of the visual nouns should be physical entities and non-visual nouns must be abstract entities. In addition, many words will be unknown, due to the limited coverage of WordNet. Our results shown below support our theory.

How many visual nouns are physical entities in word net?

- Physical: 49.17%
- Abstract: 19.81%
- Unknown: 31.02%

How many non-visual nouns are abstract entities in word net?

- Physical: 25.30%
- Abstract: 42.15%
- Unknown: 32.56%

4 Computer Vision

We describe in detail the components and the logic of our object detection, image parsing, and scene recognition sub-systems that answer the questions “what is in the picture and where is it?” for other systems in our project. We also describe a few supervised and unsupervised learning methods developed to “clean” the initial noisy object detection results. Object detection re-ranking results using those methods are reported and their performance are discussed. These are used later as input to our approaches for description generation.

4.1 Object Detection Sub-system– Xufeng Han, Stony Brook University

Object detection system tries to answer central questions about an image: “What are there in the image and where exactly are those things?”. Answers to these questions help us study many aspects of visually descriptive text. On the generation side, for example, the detected objects in a picture along with their location, their attributes such as color and texture, and the detector confidence may be fed into a text generation system to form a grammatical and natural-sounding caption for that image. On the analysing side, object detection results provides an annotation of our large captioned image dataset, which together with the text, facilitates studies on different factors that affects the probability of an object being described, and to what extend it will be described. Besides, object detection made visual attributes learning possible. Objects are the entities that bear different visual attributes, and hence from their visual patterns we can build models of different visual attributes that humans use to describe objects.

The system consists of a database, a keyword filtering program and a detector coordinator. Initially the database stores basic information of an image such as the image ID, the location of the image file. Later, it is filled with other information such as what are the objects in an image, visual features of an image and even parse tree associated to the caption of an image. The keyword filtering program retrieves the caption of an image from the database, filters out using a synonym dictionary keywords corresponding to objects we can detect, and stores each image’s keywords back into the database. The detector coordinator takes an image and a keyword as input and runs the right type of detector with the right model on the image (we have a collection of object detectors and models from different sources). It will write the detection results, if there is any, back to the database. For person and certain kinds of animals, on detection of at least one instance, it also runs the corresponding action detectors. For details, see Table 32.

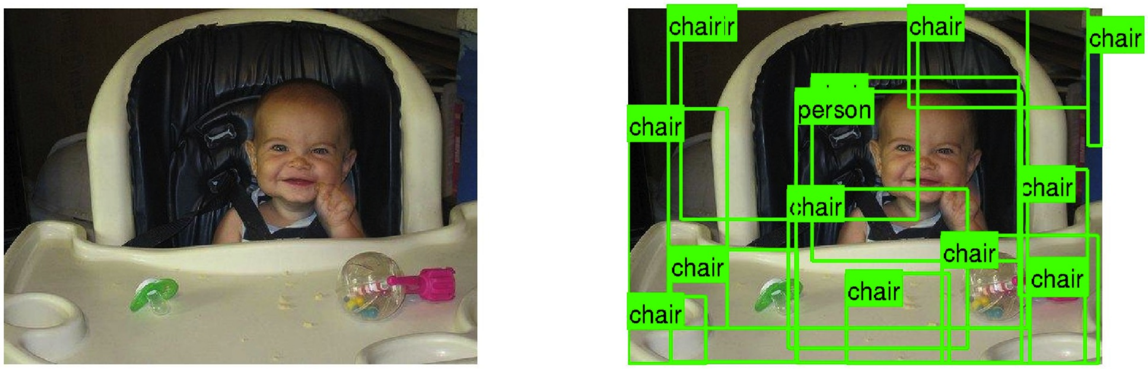


Figure 18: Left: original image. Right: keyword triggered object detection result. Keywords are “person” and “chair”, extracted from the original caption.

For example, The original image in Figure 18 was captioned as

> Tiny E sitting in the big **boy** high **chair** with the tray eating rice crispies for the first time.

The keyword filter program pulled this caption from the database and found two keywords *boy* and *chair*. *Boy* was further mapped to *person* and stored back in the database along with the other keyword *chair*. The detector coordinator retrieved both keywords (person and chair), ran person detection using *Poselet* detector and ran chair detection using *part-based* detector. Since a person was found, the coordinator also run 12 human pose detectors. The detection results are mainly the bounding box coordinates and a detection confidence for each box.

The person detector found one person at the right location. The chair detector found 13 chairs all over the image, obviously including many false positives within whose bounding box the pattern was neither a chair nor part of a chair. False positives are common among object detectors. It gets worse if we blindly run every object detector we have in the collection on each image. Not only it will take much longer, but the result will be too noisy to be useful. The limitation of the performance of those detectors makes us go for the keyword-triggered detection scheme.

4.1.1 Keyword Filtering

We perform keyword filtering using a hand-coded synonym dictionary that maps words or phrases to keywords (things that we are able to detect). For example, in our dictionary the following words and phrases are mapped to the keyword *dog*: *dog, puppy, doggy, doggie, golden retriever, afghan hound, dalmation, akita, spaniel, terrier, pit bull, shepherd, hound, collie, bloodhound bichon, frise, bulldog, pup, blood hound, afganhound, goldenretriever.* Note as in the case of "afganhound" and "goldenretriever", we concatenate two or more words in a phase to form a new word to deal with various typos commonly found in real Flickr image captions.

Although the synonym dictionary does most of the job, in order to catch more keywords, we replace non-alphabetical characters with a blank space. We also lemmatize each word in the caption before applying synonym mapping to them so that plural forms like "Dalmations" will also be correctly mapped. Not only to each word, we also apply the mapping to each pair of adjacent words, so we do not miss phrases like "golden retriever" where neither word could be mapped to "dog" by itself.

4.1.2 Object and Action Detectors

We use pre-trained object models from different sources. Table 32 summarizes the detectors we used and their sources.

4.1.3 Speed and Precision of detectors

The approximate average running time of different detectors are listed in Table 33. The computation was performed on a 3GHz 12-core dual processor computer with 6 jobs running in parallel.

Detection of the same kind can be sorted by the detector confidence. We checked by hand the precision of the top 20 detections of 13 detectors and listed the results in Table 34.

Detector	Source	Note
bird, cat, cow, dog, horse, sheep, airplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, table, potted plant, sofa, monitor	LSVM-MDPM[28]	19 PASCAL Visual object classes (without person)
balcony, bowl, butterfly, cake, castle, chicken, cup, flower, glass, hill, house, laptop, oven, pig, pillow, pizza, pole, roof, shirt, tiger, window	LSVM-MDPM	21 models trained by Girish Kulkarni (Stony Brook University)
attire, bag, ball, basket, bathtub, bear, bed, book, bouquet, box, bridge, cabinet, camera, candle, clock, cloud, computer, cross, curtain, door, duck, elephant, elevator, fish, fruit, guitar, hat, keyboard, lion, microwave, monitor, monkey, mountain, mug, newspaper, plate, pot, sand, shelf, ship, shoe, sidewalk, sign, streetlight, toilet seat, tower, truck, turtle, wall	Object Bank[51]	49 models
bird: fly, stand cat: face, lie down, sit, stand cow: face, sit, stand, stand and eat dog: face, lie down, run, sit, stand duck: stand, swim horse: face, run, stand lion: face, sit, stand tiger: face, sit, stand	LSVM-MDPM	26 animal action models trained by Girish Kulkarni
person	Poselet[10]	
person: phone, play instrument, read, riding bike ride horse, run, take picture, use computer walk, stand, sit, face	Poselet	12 human action models

Table 32: 128 Detectors used in our system.

Detector source	Approx. time of scanning an image	Number of models
Object Bank	1 sec	49
LSVM-MDPM	10 sec	66
Poselet	120 sec	13

Table 33: Speed of detectors

Detector	Precision of top 20	Detector	Precision of top 20
Attire	0.95	Ball	0.80
Fish	0.25	Fruit	0.95
Window	1.00	Airplane	1.00
Bicycle	1.00	Horse	1.00
Oven	1.00	Flower	1.00
Bird flying	0.95	Cat lying down	1.00
Duck swimming	1.00	Horse standing	1.00
Person	1.00		

Table 34: Precision of detectors

4.2 Detection Cleaning

False positives are inevitable when we run a hundred detectors on hundreds of thousands of images. In fact, even though we only run keyword-triggered detectors false positives are still common, as we see in Figure 18, let alone running all detectors on query images that have no captions. On the other hand, we want to re-evaluate the detections because given that at the top of this large set of detection results are mostly true positives for many categories (see Table 34), we could potentially do a better job at refining the detections. For those two motivations we do detection cleaning by re-ranking detections based on their visual similarities to exemplars among the original detections.

Our approach includes two steps. First compute similarities between the query image and a set of exemplar detections. Then depending on whether an unsupervised method or a supervised one is used, we compute a new score either by averaging scores of the exemplars or from a discriminative classifier. To evaluate these new scores, we re-rank a set of labelled test images using these scores and compare the precision-recall curve with that of ranking using the original detection scores.

4.2.1 Features and Similarities

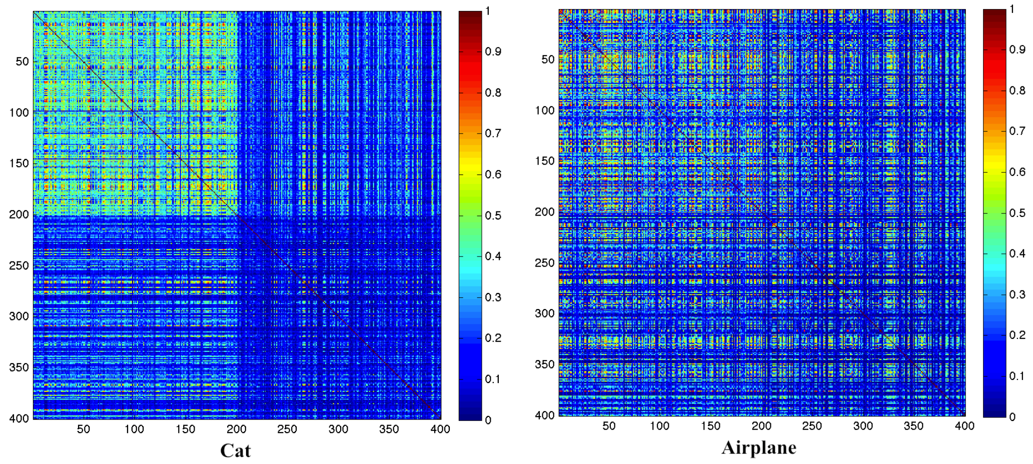
Several common features, from computer vision were used to represent detections and to compute similarity. They are spatial pyramid of bags visual words[50] based on SIFT[53], color and texton[57] features. A visual word dictionary is built by clustering the raw features and keeping the cluster centers. The corresponding visual word of a raw feature is the index of the entry in that dictionary to which the raw feature has the smallest distance among all entries. A bag of visual words is represented by a histogram of visual words over a region of the image. A simple bag of visual words over the whole image does not keep track of where those visual words are from and thus lose important information on the spatial configuration of the image. Spatial pyramid of bags of visual words encode spatial configurations by recursively divide the image into smaller regions and append bag of words features over those regions to histograms of the previous levels. For a 3-level pyramid, we have one bag for the whole image, 4 bags for its four sub-regions and 16 bags for smaller regions (4 for each of the previous four regions) concatenated together to form a large feature vector.

The "cosine" similarity $\mathrm{s}(\vec{v}_1, \vec{v}_2)$is used to measure the similarity between to feature vectors $\vec{v}_1$ and $\vec{v}_2$:

$$\mathrm{s}(\vec{v}_1, \vec{v}_2) = \frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1||\vec{v}_2|}. \tag{1}$$

$\mathrm{s}(\vec{v}_1, \vec{v}_2)$ is between 0 and 1. The higher the value, the more similar $\vec{v}_1 and\ \vec{v}_2$ are.

In order to check that the similarity measure makes sense, similarities between each pair of detections in the set of the top 200 detections plus the bottom 200 detections are computed and illustrated in Figure 19.



Cat

Airplane

Figure 19: Similarity matrices based on spatial pyramid of bag of word SIFT feature.

Figure 19 shows the similarity matrices for cat and airplane detections. The first 200 rows and the second 200 rows correspond to the top 200 detections and the bottom 200 detections respectively. For example, the upper-left section of the matrix represents similarities between pairs of detections of the top 200; the lower-right section represents that within of bottom 200. The more reddish the color, the more similar it indicates.

We see a clear block structure in the cat matrix, implying the top cat detections are similar to themselves and the bottom detections are different in different ways, which verifies both that the feature plus similarity make sense and that top detections are good enough to be exemplars. Although in airplane detections we do not see a block structure as clear as in cat detections. We still see a brighter upper-left section. The hope is by using some of the machine learning methods, we can still extract some information from those exemplars and improve the ranking.

4.2.2 Reranking by Unsupervised and Supervised Learning

In the unsupervised setting, we do not label the original detections but trust the similarity measures and the quality of the exemplars. The algorithm is simple. We find the K most similar detections to the query among the *exemplar set* (the top and bottom 200 detections) and re-score the query using the average score of those detections.

In the supervised setting, the exemplar set is labeled with binary values indicating whether the detection is a true positive. Each query is represented with a feature vector of the same length as the exemplar set. Each dimension of the feature vector corresponds to the similarity of the query and one of the exemplar detections. Different classifiers can be trained to predict the label for the query as well as output a confidence of the prediction. In our experiment, we evaluate K-Nearest Neighbor and SVM with RBF kernel.

4.2.3 Experiments and Results

The re-ranking of detections is evaluated in a labeled test set of size 200 for the category airplane. We plot precision-recall curves generated from the new scores and the original scores for comparison.

In the unsupervised setting (Figure 20), $K = 7$. In the supervised setting (Figure 21), parameters such as the number of neighbors for K-Nearest Neighbor classifier and γ and C for SVM's RBF kernel were selected by optimizing classification accuracy on the training set. The parameters in the test set evaluation are $K = 5$, $\gamma = 10$ and $C = 10$.

In both plots we see the new rank outperform the original rank in the low-recall-high-precision area, although neither the supervised method nor the unsupervised one is better then the original score overall. KNN does not drop as fast as SVM or the original in the high recall region. However we do not have the data in the low-recall region due to our relatively small labeled test set.

We also see in the unsupervised plot the high-recall part of the curves of new scores are under the one curve of the original score, whereas in the unsupervised plot, the curves in the high recall regions are comparable to or even better than the original ones. Therefore, the supervised methods are better than the unsupervised ones overall.

To further examine the reason our re-ranking methods' performance, we show in Figure 22 the most similar images found in the exemplar set to 5 queries. The problem is many images on the top list are not visually similar to the query, which is not surprising given the similarity matrix shown in Figure 19. We need better features to get good similarity.

4.3 Discussion of object detection strategies

We introduced our object detection system and explored cleaning the detection results based on visual similarity. We tried using unsupervised and supervised methods to re-rank noisy detections given a set of also noisy exemplar detections. Overall, our supervised methods outperform our unsupervised ones, although none are substantially better than baseline. Our new rankings achieve better precision than baseline when recall is not high. In order to make further improvement, a better visual feature for the measure of similarity is necessary.

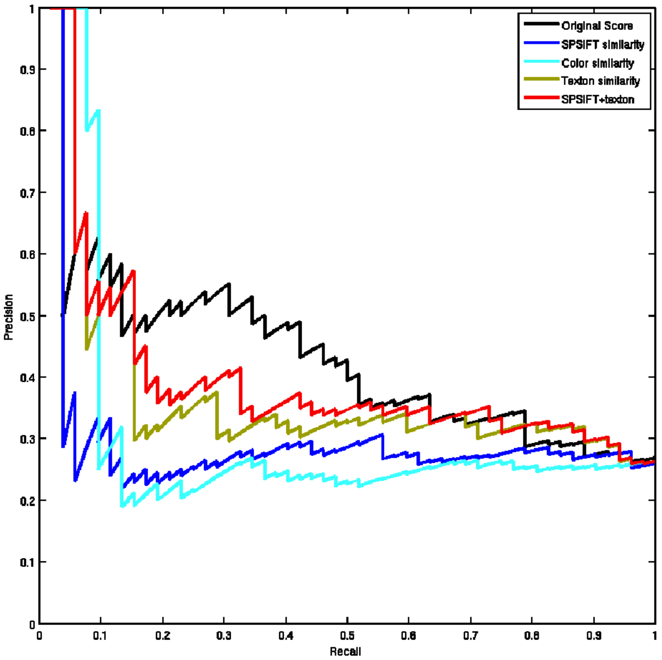


Figure 20: Precision-recall using unsupervised re-ranking with different features.

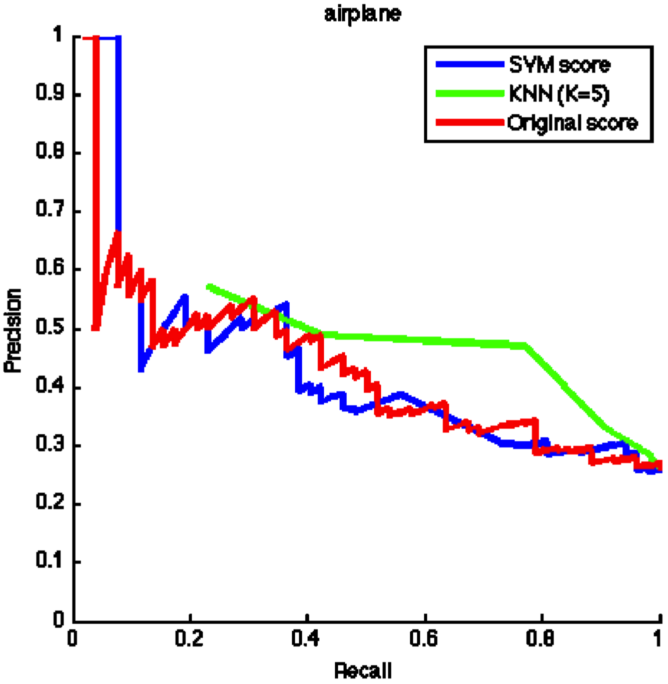


Figure 21: Precision-recall using supervised re-ranking with different classifiers.

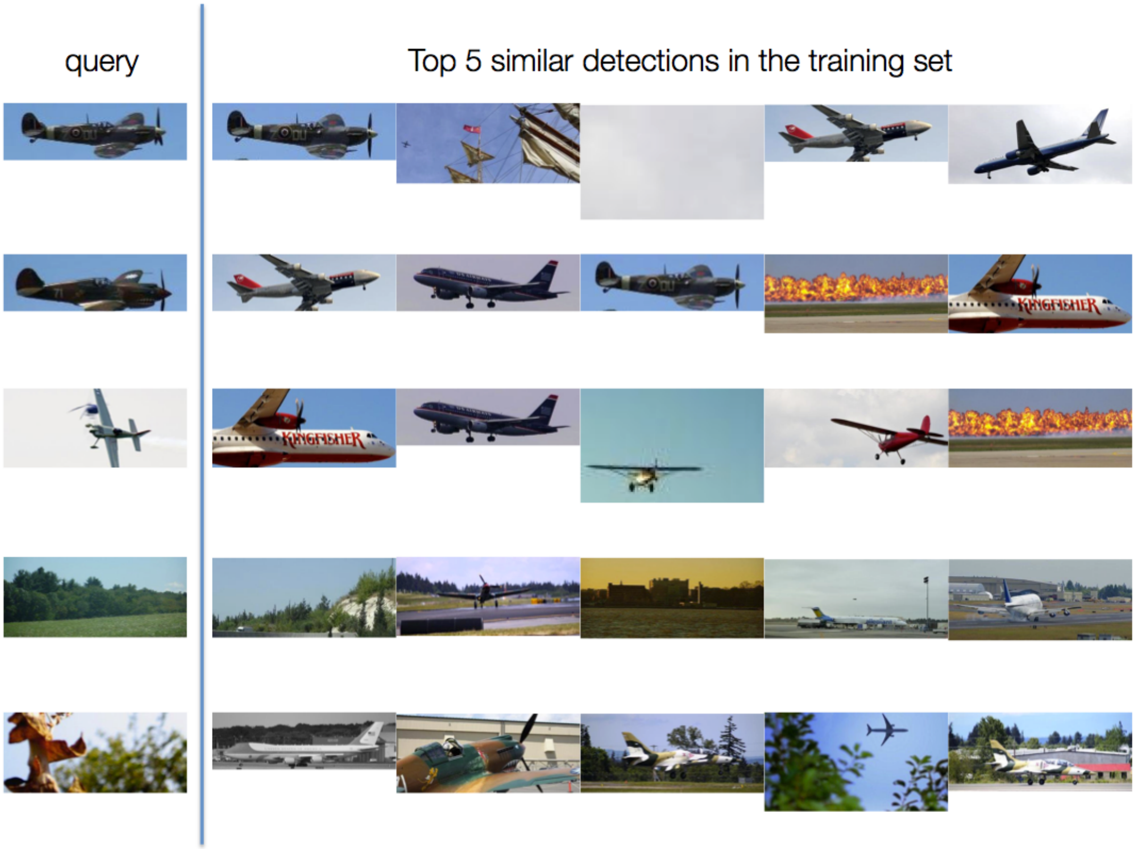


Figure 22: Top 5 similar detections found the exemplar set for 5 queries.

4.4 Mass noun detection – Kota Yamaguchi, Stony Brook University

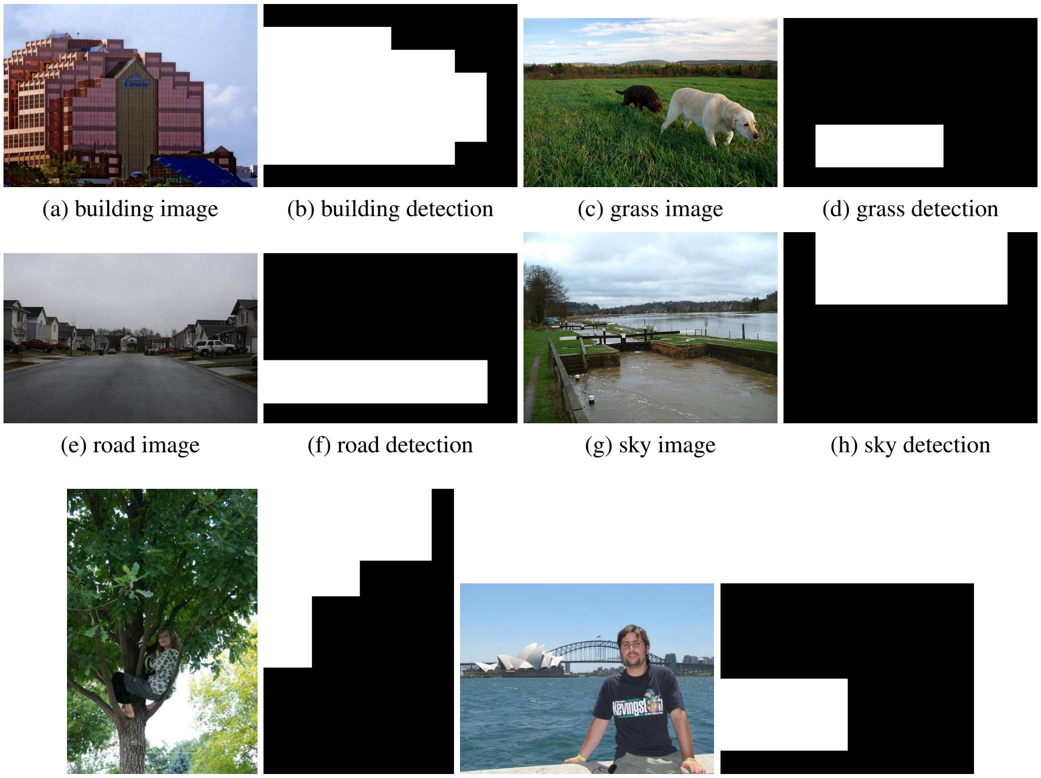
In image understanding, we do not necessarily see an object that we can locate with a bounding box around the entity. For example, it is common to find sky, grass, or trees in the outdoor scene. These mass noun entities usually take most of the space in the picture frame and specifying them with a rectangular bounding box can result in the whole image region. It is desirable to have finer control over localization of the mass entities in the scene. For this purpose, we build a stuff detector that is capable of producing a map of detection result instead of bounding boxes in the picture frame.

The stuff detector is technically the same to the classic object detector in computer vision, except that detectors are fired up in the certain grid points in an image. In overall, the detection algorithm is described in the following procedures:

1. Sample image patches at the grid points
2. For each image patch, compute feature representation
3. Classify the category of each patch based on the feature representation
4. Assign the most confident category label to the patch region

Grid points determines the resolution of the detection. The extreme case would be to sample at every pixel, and in this case we would be able to produce a pixelwise labeling of the stuff category. In our experiment, we define 8×8 grid over the image coordinate, and sampled image patches at $1/4$ size for both width and height of the image. In the end, we get overlapping 64 patches for each image.

We compute 4 different image features from the patch. The first feature is the normalized histogram of quantized HOG feature in the patch region [26]. We first compute HOG descriptor at dense



(a) building image (b) building detection (c) grass image (d) grass detection

(e) road image (f) road detection (g) sky image (h) sky detection

(i) tree image (j) tree detection (k) water image (l) water detection

Figure 23: Stuff detection

grid defined over an image at different resolutions, and quantize each descriptor by assigning a so-called *visual word*, a unique identifier of the vector. This assignment of a visual word is based on the nearest neighbor search to the dictionary trained with k-means clustering algorithm. After the quantization, we look at the the visual words contained in the image patch and compute histogram of them. Similarly, we compute the normalized histogram of quantized Lab color, Texton, and Geometric context [42] for each patch.

We use a support vector machine for classification [12]. The radial basis function is used for the kernel of the classifier. In our experiment, we pick 6 mass nouns, *building, grass, road, sky, tree*, and *water*. The number of training images are 200, 200, 163, 200, 200, and 30, respectively for each category. These training images are obtained from ImageNet [19]. Figure 23 shows an example of the detection.

We applied this stuff detector to images in Flickr708k dataset. The detection results are used to compute image similarity in content-based image retrieval.

4.5 Scene classification

Another kind of important information people get from an image is the type of scene. We understand a picture not only from individual objects present in a picture but also from the context in which those objects appear. For example, given a picture of the office environment, we probably say that the picture is taken in an office but not that the the picture shows two chairs and a desk on top of which there are a notepad and a pen. Since scene comprises of an entire region of a picture, scene is a global information. Therefore, scene understanding is a top-down approach to interpret an image, in contrast to object or mass noun detection that try to understand a picture in a bottom-up approach.

Table 35: Plain scene classification performance

Category	P@20	Category	P@20	Category	P@20
bar	15	dining room	20	market / outdoor	15
bathroom	25	field / cultivated	60	mountain	5
beach	50	forest / broadleaf	20	ocean	50
bedroom	30	highway	0	office	0
building facade	90	kitchen	45	pasture	80
canyon	25	lake / natural	90	river	35
classroom	0	library / indoor	20	street	80
coast	95	living room	15	theater / indoor proscenium	0
corridor	0	market / indoor	5		

Scene understanding in computer vision is formulated as an image classification problem. That is, given an image, we would like to associate a scene category to it. Thus, the generic approach is described in the following two steps:

1. Compute feature representation of the entire image
2. Predict a scene category given a feature representation

In our experiment, we use the approach of [86] for scene classification.

In [86], an image is represented by the combination of the following 12 features: GIST, HOG 2X2, Dense SIFT, Local Binary Pattern, Sparse SIFT Histograms, Self Similarity Image, Tiny Image, Line Features, Texton Histograms, Color Histograms, Geometric Probability Map, and Geometric Specific Histograms.

The classifier is a SVM with linearly combined kernels from each feature representation. We first compute kernel matrix for individual features with either of histogram-intersection kernel, χ^2 kernel, radial basis kernel, or radial basis with χ^2 kernel. The computed kernel matrices are then linearly combined with associated weights to give the final kernel matrix.

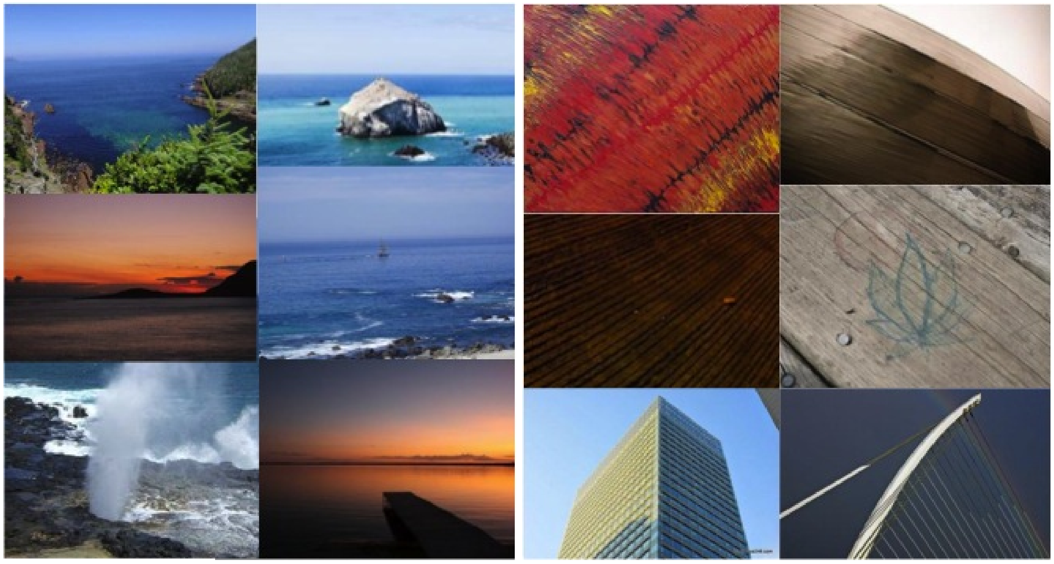
We trained classifiers for 26 categories. The training images are obtained from SUN dataset [86]. We picked 50 training images for each 26 category with negative samples randomly picked within this training set. Categories are shown in Table 35.

Plain classification performance The scene classification is very challenging in general. For preliminary evaluation, we applied the scene classifier to 34k images in Flickr708k dataset and see the performance in the plain classification task. Table 35 shows the precision of the top 20 images from the 34k images ranked by the confidence of the classification of each category. Note that these 34k images do not necessarily contain the evaluated 26 scene categories and the upper limit of the precision could be 0 if the category is not contained at all.

As seen in Table 35, the performance significantly depends on the type of the scene. Outdoor scenes such as *building facade*, *coast*, or *lake / natural* have strong prediction performance, while indoor scenes in general have lower performance due to the confusion among them. Another possible reason of the lower performance is the overfitting to specific feature in the training. Figure 24 shows some examples of *coast* and *mountain* classification. While *coast* classifier could be correctly trained from the dataset, *mountain* category is likely to be too strongly affected by the presence of diagonal edges in the picture. In general, it is essential that the scene is visually well defined and the training condition well reflects the testing condition for the best scene classification performance.

Scene descriptor performance Although it is not easy to get stable performance for scene classification for all the categories, the confidence output of the 26 classifiers can give compact yet stable representation for the image. We treat this representation of an image as scene descriptor. The scene descriptor can be used to measure similarity between images in terms of scene semantics.

In an attempt to retrieve captioned images from the Flickr708k dataset using image query, we compared the retrieval performance between scene descriptor and the low level feature as a preliminary



(a) coast

(b) mountain

Figure 24: Scene classification

Table 36: Caption retrieval performance

Retrieval method	Scene Descriptor	HOG+Texton+Color	Random
Feature dimension	26	1384	n/a
Mean AP	0.058	0.058	0.046

experiment. In this experiment, we use the 1384 dimensional feature consisting of quantized histogram of HOG, Lab color, and texton features sampled from the entire image.

The retrieval experiment starts by preparing a captioned image to be used as a query to the Flickr708k dataset. Next, captioned images in the dataset are ranked according to the similarity to the query image. The ordered list of captioned images are the result of this retrieval. This retrieval procedure is a core of the automatic image annotation system based on caption transfer.

The retrieval performance is measured in terms of the overlap ratio of the words in the caption. First we define the precision and recall of the retrieval of the top k as the following:

$$\begin{aligned} \text{Precision}(k) &\equiv \tfrac{|Q \cup R(k)|}{|R(k)|}, \\ \text{Recall}(k) &\equiv \tfrac{|Q \cup R(k)|}{|Q|}, \end{aligned}$$

where Q is the set of words in the caption of the query image and $R(k)$ is a set of words contained in the top k list of retrieved captions. With these definition of precision and recall, we use mean average precision as the performance of the retrieval system. Note that this is only a preliminary method to evaluate the retrieval performance and these definition give low mean average precision compared to the typical evaluation protocol in information retrieval.

Table 36 summarizes the retrieval performance. As the baseline comparison, we also include the random retrieval in the table. The result shows that only 26 dimensional representation gives retrieval performance comparable to 1,384 dimensional representation. As seen from this result, the scene classifier could give compact and efficient representation of an image even if the performance of the plain classification is not stable. We use this result in the sentence generation approach described in the next section.

4.6 Automatic attribute learning – Kota Yamaguchi, Stony Brook University

One essential component to enrich the description of a visual entity is the modifier associated to the attributes of the object. Consider the following two sentences describing the same picture:

1. *There is a dog.*
2. *A brown and white dog looks happy after a good bike ride.*

While the first sentence gives factual information about the picture, it is hard to imagine from the description how the picture looks like since the only information we get is the existence of the dog. In contrast, the second sentence gives additional visual information such as *brown and white* (color) or *happy* (expression). Even the contextual information *after a good bike ride* could affect the visual of the picture, since it wouldn't be difficult for us to think that the dog might be feeling hot and hanging out its tongue. As we see in this example, the richness of the visual description comes not only from the existence of the object but also from the information associated to the object. The goal of attribute learning is to automatically build a visual classifier that analyzes visual attributes associated to the object, with the help of natural language processing techniques.

There has been a number of studies of object attributes in computer vision [7, 47, 84, 49, 26, 31, 81, 87]. The learning of visual attributes basically starts from getting images (or image regions) annotated with attribute label. Once such data samples could be collected, we would be able to use any combination of image feature transformation and classifiers to build a model. However, the challenge in this process is the difficulty of data acquisition. To be able to cover any possible word to describe visual attribute, we need to collect labeled images for any possible modifier or phrase in natural language that is used for visual description. Due to the expense of obtaining such labeled data, most of the work focus on developing a technique to utilize weakly annotated data [7, 31, 81, 87], or combination with an unsupervised approach [47]. In our experiment, we try to learn visual attributes with the help of text mining technique in the context of weakly supervised approach.

Related to the use of attributes in visual description, there is an attempt of using attributes as a higher level representation of images. One characteristic of visual attributes is that it is often coherent across object categories. For example, we can use *red* to describe an apple or a car. The construction of attribute classifier is therefore assumed to be independent of object category. Beyond classification task, this independence actually implies the use of attributes as a category independent representation of an image, which could further be used as a stable feature for object detection or classification tasks [47, 49]. In our experiment, however, we mainly focus on the learning of attributes.

4.6.1 Weakly supervised learning

The goal in our learning method is to build a classifier that covers as many visual attributes as possible while reducing the amount of effort of manual annotation. For this purpose, we consider the entire learning process in two steps.

Visual attribute discovery The first step is the discovery of the visual attributes from text corpus. Through this process, we build up a vocabulary of the visual attributes so that later we can try training a classifier for each word. We take the iterative approach described in Section 3. The approach basically consists of iteration of finding words having a similar context to the known set of attribute words. The initial set of vocabulary is listed in Table 37. Note that these are the only manual annotation to start our learning method. In our experiment, we discover attribute words from Wikipedia corpora and continues iteration until we get 130 words for each of 11 categories listed in Table 37. This result is considered the potential attribute words.

Captioned object detection The next step is to detect attributed objects in a captioned image dataset. In caption texts, we look for modified nouns such as *a red apple* or *a brown bear* that we can detect with an existing object detector. Then, we detect the location of an object in an image for such modified nouns. Now, we select such detections that have a modifier appearing in the attribute vocabulary we mined in the previous step. These modified nouns (=attributed detections) serve as positive sample to train an attribute classifier.

Table 37: Initial visual attributes

Type	Terms
direction	sideways north upward left down
beauty	beautiful cute pretty gorgeous lovely
color	purple blue maroon beige green
material	plastic cotton wooden metallic silver
surface	coarse smooth furry fluffy rough
quality	shiny rusty dirty burned glittery
ethnicity	french lunar american greek hispanic
size	small big tiny tall huge
shape	circular square round rectangular triangular
pattern	striped dotted checked plaid quilted
age	young mature immature older senior

Table 38: Potential visual attributes

Type	Terms (#of samples)
direction	north(59) left(44) south(57) long(73) east(164) northern(39) central(41) native(86)
beauty	beautiful(753) cute(1000) pretty(306) gorgeous(46) lovely(159) wonderful(51) nice(244) famous(58) interesting(127) amazing(66) happy(140)
color	purple(161) blue(1000) green(1000) red(1000) white(1000) black(1000) orange(36) yellow(1000) gray(119) brown(530) pink(1000) dark(108) bright(38) colorful(280)
material	plastic(607) wooden(1000) silver(151) decorative(30) handmade(61) painted(106) antique(53) empty(104) rear(51)
surface	furry(59) fluffy(58) stray(162) hard(41) soft(35) wet(61)
quality	shiny(57) rusty(144) dirty(179) exterior(38) abandoned(82) fresh(82) broken(87) open(91)
ethnicity	french(157) american(57) chinese(73) japanese(134) local(313) asian(96) indian(41)
size	small(946) big(1000) tiny(252) tall(84) huge(125) large(244) largest(40) main(137) biggest(41) great(156) real(61) single(106) massive(33) entire(35) private(37) original(74) clear(63) full(46) high(236) low(84)
shape	round(47) shaped(51) concrete(48)
pattern	striped(76) comfy(43) traditional(218) matching(33)
age	young(1000) good(109) female(248) homeless(80) poor(131) bad(76)

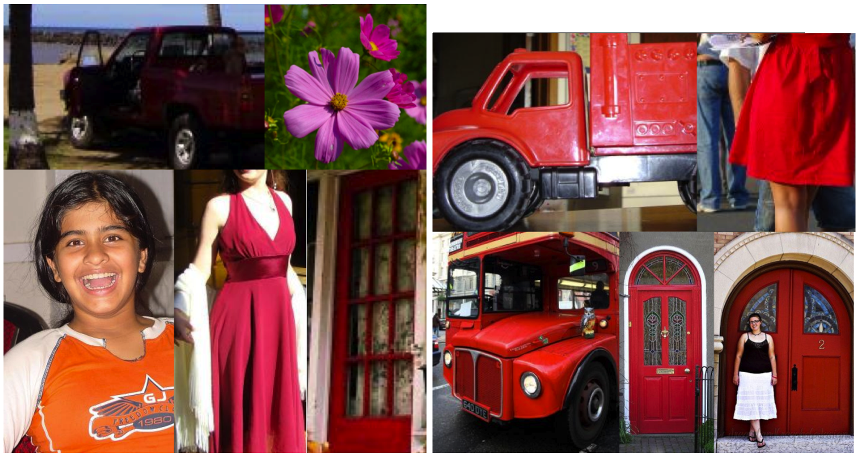
In our experiment, we run object detectors described in Section 4.1 in Flickr708k dataset and pick up positive samples for each attribute word discovered in the previous step. To stabilize training, we discard attributes having less than 30 samples in this step. Table 38 lists the resulting attribute words having more than 30 samples in Flickr 708k dataset.

For feature representation of an image region, we use the concatenation of normalized histogram of quantized HOG descriptor, Lab-color, and texton [26]. We use a support vector machine with radial basis kernel for classification [12].

Note that the resulting positive samples for training are usually noisy. It is absolutely possible that even if the object is mentioned in the dataset that object is not shown in the picture at all. However, if there exists a coherence in these samples, we could still train a classifier.

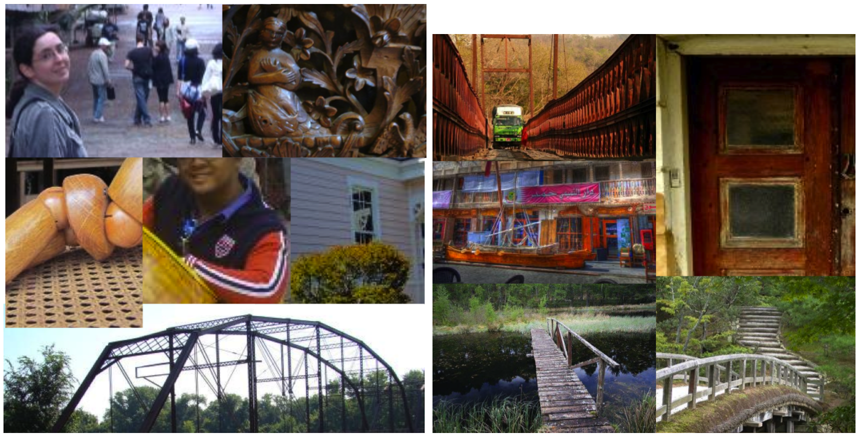
4.6.2 Evaluating attribute classifiers

We evaluate our learning method by the quality of the ranking of the positive samples based on the score of the classification. For samples in each attribute word, we assign a score of the classification using 3-fold cross validation; The positive samples are first split into 3 folds and the score is assigned to each fold using a classifier trained on samples in the other 2 folds. We pick negative samples up to 1,000 randomly from the other attribute words.



(a) Unordered (b) Reranked

Figure 25: Attribute: red



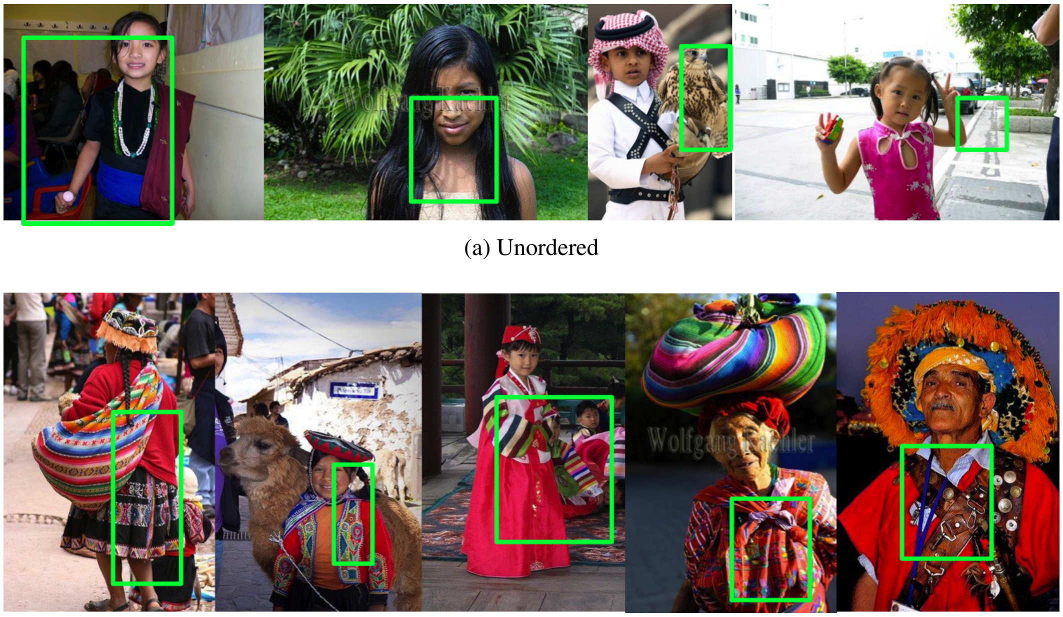
(a) Unordered (b) Reranked

Figure 26: Attribute: wooden

Qualitative evaluation Figure 25, 26, and 27 shows an example of (a) positive samples and (b) highly ranked samples based on the score of the classification, for attributes *red*, *wooden*, and *traditional*, respectively. As we could see in the figure, reranking successfully retrieves visually coherent samples to the top of the list.

Quantitative evaluation We evaluate the ranked samples with two measures. One is the F-score of the binary classification performance for each word, assuming that positive samples have no noise and are truly positive. The other measure is the precision of the top 20 samples evaluated by human for each word. In our experiment, only one same human annotator evaluated the precision of all words. Roughly speaking, the f-score is an automatic approach to look at the goodness of the learning while the precision at 20 is a manual approach. Figure 28 summarizes the result.

There are several observations in the result. The first is that the type of attribute affects the learning. It is clear that *color* has strong results while *direction* is not. This is directly related to the visualness of the attribute word – *color* terms are almost always associated to the color value of the pixels while it is possible the word *north* or *left* have nothing to do with the appearance of the picture. The other easy-to-learn attributes include *pattern* such as *traditional*, surface such as *fluffy*, or *material* such as *wooden* or *plastic*. In a sense, our result shows the visualness or learnability of the attribute word.



(a) Unordered

(b) Reranked

Figure 27: Attribute: traditional

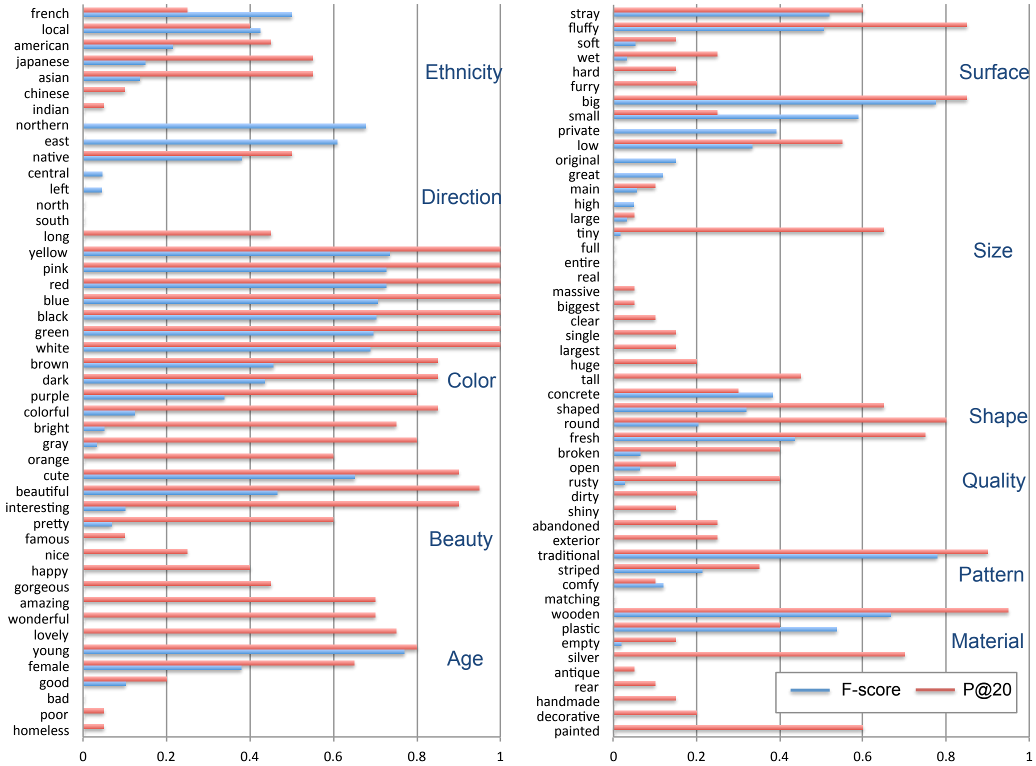
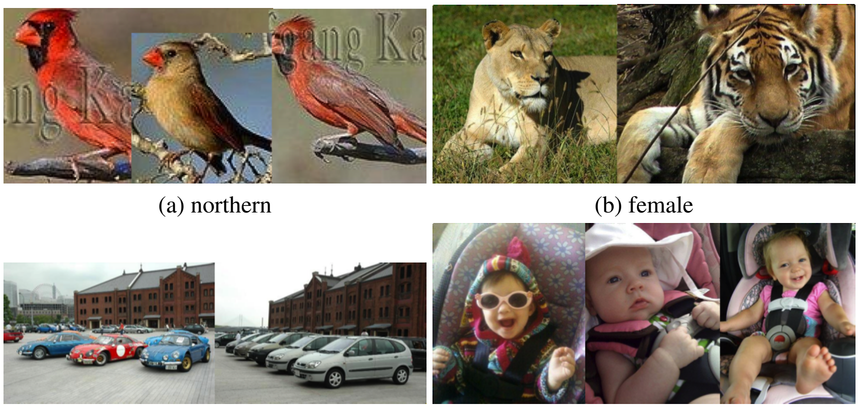


Figure 28: Attribute classification evaluation



(a) northern

(b) female

(c) french

(d) big

Figure 29: Unexpected attribute classification

The f-score and precision at 20 are mostly correlated. However, we can find cases where the precision shows correct classification even if the f-score is zero. In this case, our classifier correctly picked up appropriate samples in the result that are originally marked as negative in the training.

Interesting cases include where we observe high f-score and low precision. This happens when positive samples are strongly biased. In these cases, we observed that the use of the attribute word is tightly coupled to a noun and a classifier learned the joint of attribute and an object. For example, the word *french* always appears with *cars* in our Flickr 708k dataset and the ranking result always shows cars in a certain exhibition. Figure 29 shows examples of those cases. We would need to collect more positive samples to avoid the bias in the dataset, or we would need to incorporate the joint model with a modifier and a noun as opposed to our initial assumption that the attribute is independent of objects.

Another controversial results were attributes related to subject attributes, such as those in *beauty* category. Mostly beauty attributes get high precision in our experiment. However, the criteria to determine *beautiful* or *good* can be highly subjective. Perhaps more agreement between annotator is necessary to get reliable classification for them.

4.7 Future work on attribute learning

We showed that visual attribute classifiers could be learned with our weakly supervised method that utilizes captioned image dataset. The result of reranking indicates which attribute word is easy to learn. At the same time, failure cases occurs when positive samples are biased or we do not have enough positive samples.

For the bias of samples, it would be interesting to develop a technique to include unused samples in the captioned image dataset. Our approach currently discards unattributed object (i.e., object detection without a modifier) in the training. However, we might be able to consider hidden attributes for these samples and incorporate them in the training. This would include the joint modeling of modifier and noun.

Another possibility is to extend the approach to word types other than modifiers. Our approach picks up a modifier of a noun as a label of the sample, but we of course can pick up other kind of words. For example, learning verbs might lead us to the method of automatic action learning.

5 Generation

We explored two aspects of generation, beginning with improving the quality and variety of templates for constructing sentences for known image content (Rule Based Section) and going forward to a novel approach to select phrases for generation based on image contents (Grab and Mash).

5.1 Rule Based Generation – Margaret (Meg) Mitchell University of Aberdeen

We introduce a novel approach to generation, developed to connect computer vision output to natural language generation input. The process of generation is approached as a problem of generating a semantically and syntactically well-formed string based on computer vision "anchors". Recognized objects serve as head noun anchors in a lexicalized, generative grammar which we call a *tree growth grammar*. This grammar fleshes out the tree details around head noun anchors by utilizing lexical dependencies between closed-class words (prepositions, determiners) and open-class words (nouns, verbs, adjectives, adverbs).

The generation system was developed over the workshop as a detailed, robust approach to generation. It is linguistically sophisticated and produces many correct descriptions of our development data. One main missing piece is that *it is not yet using vision scores*; this is a vital component that is likely to improve the system overall. At the time of the workshop end, there was still not a clear vision input with usable vision scores. We therefore developed the system using training images for the object detectors, and at the end, generated descriptions for the PASCAL images[5], which have been used in earlier work [45].

Object detections from a computer vision system tend to be more reliable than scene or attribute detections, and action detections rely on object detection. We exploit this aspect of the vision system by using recognized objects as head noun building blocks for the rest of the tree. Although surrounding tree structure can be generated around these nouns, new noun heads beyond the set returned from the computer vision system cannot be generated. This allows the system to generate several varied descriptions for items in the image, while avoiding the explosion of possible tree structures inherent in generating unlimited open-class words.

This approach has commonalities with a generative lexicalized PCFG approach, but we remove some of the assumptions of structural context-freeness: Most tree growth rules are conditioned on the positions of head nouns. This is a simple approach that generates many well-formed descriptive structures.

The anchor head nouns thus help to expand and constrain descriptions in the following ways:

1. Set the number of head nouns in the sentence (or phrase).
2. Limit the kinds of branching structures that may be used to expand the tree upwards.

The anchor constraint given in (2) we will call a *directionality constraint*. This establishes which direction the structure may "grow" in, embedding a given phrase within a larger phrase whose head is to the right or to the left of the anchor noun. Using a notion of directionality has similarities to earlier work on generative parsing, such as in the SPATTER system [54], but the system retains the simpler, more computationally efficient approach of later work in parsing [15, 13] by using a language model to determine what can be generated to the right and left of a given head.

We follow a three-tiered generation process [70], utilizing *content determination* to first order the object nouns, *microplanning* to construct a tree around these nouns, and *surface realization* to order selected modifiers and, in the future, rank and select outputs. Informing these processes is a knowledge base that provides commonsense knowledge about the world.

5.1.1 Knowledge Base

Having knowledge about the world helps us to understand what to look for, to know what kinds of things are unexpected or that the system may be detecting incorrectly, and provides a way to backoff in the face of uncertainty. We discuss each of these aspects below.

---

1. What to look for. Using semantic similarity can help guide what the computer vision system tries to detect. For example, knowing that *building* and *car* are both in the scene, we can look for the semantically similar *street*, and iteratively move back and forth between what the knowledge base

[5] http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2010/

this, that, those, these, one, here, there, some, now

Figure 30: Non-NN noun heads.

JJ, JJR, JJS, VBG, VBN, VBD, RB, RBR, RBS, NN, NNS

Figure 31: Permitted prenominal modifiers describing an object.

tells us to expect and what the computer vision sees before deciding on a set of recognized objects to talk about.

Knowing what is typical of each recognized object can also help to sanity-check the vision system: if horses tend to have fur, and we recognize a horse, we can also check that the fur detector fires in approximately the same region. The *what to look for* aspect of the knowledge base is not currently employed.

2. What's unexpected. If a feature detector fires that is unexpected given our world knowledge, we will not include the feature unless its score is quite high. Currently, we are discarding any unexpected attribute value if another value of the same attribute also fires. For example, if both *feathered* and *furry* fires within the *horse* bounding box, the knowledge base tells us that p(furry|horse) > p(feathered|horse), and we will therefore not use the *feathered* detection. This falls out from the *Mutual Exclusivity Hypothesis*, which we discuss at the bottom of Section 5.1.3.

3. Backing off. When more than one object detector is firing in the same general region, for example, detecting *cat* and *dog* with relatively high scores, we can use the knowledge base to collect the properties that both of these objects have in common (e.g., SUPERCLASS: `animal`; ANIMAL-COVERING: `fur`) and backoff our description to describe the object by the given properties (*a furry animal*).

The knowledge base of our system has two key components: a *hand-coded* component, and a *data-driven* component, built from word co-occurrence statistics in the Flickr data.

The hand-coded component is still preliminary, and was put into place to inform and constrain the computer vision outputs as work on detection progresses. This part of the knowledge base uses the Scenario-Based Lexical Resource (SBLR), which underlies the text-to-graphics system WordsEye [16], in order to know typical parts of different query objects and typical objects within a scene. We are also working to incorporate McRae's norms [60], which list typical visual attribute values of concrete objects.

The current version of the generation system utilizes the *data-driven* component, using word co-occurrence statistics. For each noun phrase in the parsed data, we extract the head noun (rightmost nominal, including some not tagged as NN, listed in Figure 30) and all words (tagged as adjectives, gerunds, participles, nouns, etc., see Figure 31) that occur prenominally. These counts are used to assign probabilities p(modifier|head noun), p(head noun|modifier) and the nPMI between the noun and modifier. This information is stored in the knowledge base, indexed by <modifier, modifier tag, noun head> triples. This lets us know what tends to modify a given noun.

A large subset of the prenominal modifiers are adjectives, and we further associate these adjectives (tagged as JJ) to the attribute clusters defined in **AMIT'S BIT** within the knowledge base. This provides information on the expected attribute values a given object can have. With this in place, each object is described using a series of <attribute:value> pairs, and the system compares values within each attribute during generation. We return to this process in Section 5.1.3.

Building commonsense semantic knowledge about the world is an active area of research in NLP, and we have explored using other kinds of knowledge bases built automatically, such as NELL [63] and some of the projects developed at the University of Trento [44, 11]. However, the information these systems provide tend to be noisy, and which aspects are visual and which are not is not clearly defined. The final version of the knowledge base will therefore likely employ the SBLR, McRae's norms, and the data-driven component developed as part of the workshop.

5.1.2 Content Determination

Using the database discussed in Section 3.1.6, we learn the closest hypernyms of each of the anchor nouns. This determines the order in which the nouns will appear in the sentence: Given a set of anchor nouns *A* of size *n* – the object detections that have fired for an image – we use WordNet to associate each anchor noun $a \in A$ to its closest hypernym h_a that is also in our model. Then for each possible position of the anchor noun in the sentence, $p_i \in P_{1...n}$, we maximize the $p(p_i|n, h_a)$. We solve this problem greedily, solving for p_1 first, followed by p_2, etc., until all nouns have been ordered.

This is a novel contribution to generating captions for images, and is a simple solution that works quite well. By ordering nouns before generation begins and placing them in fixed positions, we constrain and inform the full phrase structures that may be built from them. In future work, we would like to see how well such an approach works for generation in other tasks.

We explored using nPMI to cluster sets of nouns returned from the computer vision system, grouping the nouns into sets of 2 or 3 depending on whether the nPMI values between each pair was above a certain threshold; in the case of 4 nouns, each with a pairwise nPMI above threshold, we grouped the nouns into a set of 3 and then 1, which would form a new sentence. This follows the preference for 2 to 3 nouns per main phrase discussed in Section 3.1.4, and the preference for no more than 1 or 2 main phrases per caption discussed in Section 3.1.2. As we showed in these sections, 5 or more concrete nouns is extremely rare in the Flickr captions, and so we did not allow the system to generate captions with more than 4 concrete nouns in total. However, in the interest of time, we moved away from further developing this aspect of the system. Currently, generation proceeds given a computer vision output of 1, 2, or 3 nouns.

As the ordered nouns are passed to the Microplanning stage, they are tagged (using Penn Treebank notation, NN) and given a *directionality constraint*, defined by their position in the sequence just determined by content determination. A rightward directionality constraint means that the noun will function as a subject when another noun combines with it in a tree; a leftward directionality constraint means that the noun will function as an object when another noun combines with it in a tree.

The leftmost noun in the sequence is given a rightward directionality constraint, which means it will be the subject of the sentence, and will be used to build trees that expand to the right. The rightmost noun is given a leftward directionality constraint, which means it will be an object, and build trees that expand to the left. The noun in the middle, if there is one, can have both a rightward and leftward directionality constraint, which give rise to two separate kinds of sentence structures. With a rightward directionality constraint, the middle noun will combine in a local subtree with the noun to its right, and function as the head noun in that structure; with a leftward directionality constraint, the noun will combine in a local subtree with the noun to its left, and function as a complement in that structure. In the current implementation, we only explore a rightward directionality constraint for the middle noun.

5.1.3 Microplanning

The core idea behind microplanning in our generation system is to generate syntactically well-formed trees following a lexicalized grammar, with the semantic information provided by our knowledge base and the word co-occurrence statistics.

Because we have nouns serving as our sentence anchors, we begin with the idea that the relationship between any two head nouns can be characterized as one of the following:

1. verbal (a boy *cleans* the table)
2. prepositional (a boy *on* the table)
3. verb with preposition (a boy *sits on* the table)

As mentioned above, each noun anchor has a directionality constraint associated to it. We use the notation $noun_l$ to mark a noun with a leftward directionality constraint and $noun_r$ to mark a noun with a rightward directionality constraint. The noun, along with its directionality constraint, is passed upwards through the tree to further determine the prepositions and verbs that it combines with, using the listed rules. Nouns are passed through constituents headed by closed-class words (prepositional phrases), to be made available when forming higher subtrees. We walk through this implementation below.

Essentially, the generation system finds the probable relationships between nouns, and uses this to form a syntactic tree. In more detail, syntactic structures are built between nouns that have complementary directionality constraints (pointing in the same direction), based on probabilities learned from the Flickr data.

Probabilities are conditioned only on open-class words (specifically, nouns and verbs). This means that a closed-class word is never used to generate an open-class word. The intuition behind this idea is that the system has more fine-grained control over the kinds of structures that may be generated when it utilizes the rich distributional information provided by open-class words to choose among the small, enumerable set of closed-class words than when it uses the small set of closed-class words to choose among a vast amount of semantically dissimilar open-class words. In other words, the probability space of (closed class|open class) has many less points, with much more data per point, than the probability space of (open class|closed class). ** Does that make any sense? sigh. **

Although this is a generative model of syntax, we do not start with a single START symbol: Instead, we start with a sequence of nouns, and project local subtrees based on each. This amounts to trying to predict the parent nodes given knowledge of the child nodes.

The generation of subtrees depends on whether the probability of a noun combining with a given node is above a specified probability threshold α. In development, we found that a threshold of α = .025 produced reasonable results.

We now walk through the generation process in greater detail. Subtrees are built upwards from noun anchors following the rules listed below.

1. DT JJ* NN → NP
2. IN NP[$noun_l$] → PP[$noun_l$]
3. VBG|VBN NP[$noun_l$] → VP_VBG
4. VBG|VBN Ø → VP_VBG
5. is VBG|VBN NP[$noun_l$] → VP_VBZ
6. is VBG|VBN Ø → VP_VBZ
7. VBZ NP[$noun_l$] → VP_VBZ
8. NP[$noun_r$] (PP|VP_VBG)+ → NP
9. NP[$noun_r$] VP_VBZ → S

Rule 1:

- Generate all DT[dt] JJ[adj]* NN[$noun$] constructs where:
  - p(JJ[adj]|NN[$noun$]) $> \alpha$; $adj =$ any adjective.
  - p(DT[dt]|NN[$noun$]) $> \alpha$; $dt =$ indefinite/definite/Ø, and the indefinite article is then realized as *a* or *an* depending on whether the first character of the immediately following word is a vowel or a consonant.

Any number of adjectives (including none) may be generated, given the further <attribute:value> constraints from the Mutual Exclusivity discussion below.

The motivation behind the determiner (DT) constraint was to learn whether to treat the given noun as a mass or count noun (not taking a determiner or taking a determiner, respectively) or as a given or new noun (phrases like *a sky* sound unnatural because *sky* is taken to be given knowledge, requiring the definite article *the*). However, the selection of determiner is not independent of the selection of adjective; *a sky* may sound unnatural, but *a blue sky* is fine. Further development of the model should take the dependency between determiner and adjective into account.

Rule 2:

- For each object NP with a leftward directionality constraint (NP[$noun_l$]), find the nearest NP with a rightward directionality constraint (NP[$noun_r$]), and generate all permitted IN[$prep$] NP[$noun_l$] trees, such that:

- p(IN$[prep]$|NP$[noun_r]$=SUBJ) $> \alpha$
- p(IN$[prep]$|NP$[noun_l]$=OBJ) $> \alpha$

Rules 3, 5, and 7:

- For each object NP with a leftward directionality constraint (NP$[noun_l]$), find the nearest NP with a rightward directionality constraint (NP$[noun_r]$), and generate all permitted subtrees:
  - VBG$[verb]$ NP$[noun_l]$
  - is VBG$[verb]$ NP$[noun_l]$
  - VBN$[verb]$ NP$[noun_l]$
  - is VBN$[verb]$ NP$[noun_l]$
  - VBZ$[verb]$ NP$[noun_l]$

  where:

  - p(VP$[verb]$|NP$[noun_r]$=SUBJ) $> \alpha$
  - p(VP$[verb]$|NP$[noun_l]$=OBJ) $> \alpha$

  These rules are only used when an action detection is requested – either because we're in "hallucination" mode (deciding on probable verbs given the nouns in the sentence), or because an action/pose detection has fired. When action or pose detections fire, they are indexed with a specific object (noun); the thing doing the action. In these cases, the set of possible verbs is limited to the set of action detections.
- Tree insertion comes into play at this rule: For a selected verb, the model determines whether it should take a prepositional complement PP$[noun_l]$, where $noun_l$ is in a noun phrase complement of the preposition, or a nominal complement NP$[noun_l]$. This is checked based on the following probabilities:
  - (p(VP$[verb]$|NP$[noun_l]$=OBJ) $> \alpha$ AND p(PP$[prep]$|VP$[verb]$=HEAD)) $> \alpha$
  - p(PP$[prep]$|NP$[noun_l]$=OBJ) $> \alpha$

  This is the only place in the model where probabilities are conditioned on something other than a noun. Because we are still generating closed-class words based on open-class words, we avoid an explosion of nonsensical forms.

Rules 4 and 6:

- Allow for the generation of intransitive verbs, where:
  - p(Ø|VBG$[verb]$) $> \alpha$
  - p(Ø|is VBG$[verb]$) $> \alpha$
  - p(Ø|VBN$[verb]$) $> \alpha$
  - p(Ø|is VBN$[verb]$) $> \alpha$
- These rules are only used when there is a single object detection, when a tree is being built upwards from Rules 8 and 9.

Rule 8:

- For all NP$[noun_r]$ subtrees, form all larger NPs with the subtree to the right – either a PP headed by $prep$ or a VP_VBG headed by $verb$, where:
  - p(PP$[prep]$|NP$[noun_r]$) $> \alpha$, or
  - p(VP_VBG$[verb]$|NP$[noun_r]$) $> \alpha$.
- Note in this case the system can generate intransitive forms of verbs.

Rule 9:

- For a subject NP$[noun_r]$, generate trees of the form NP$[noun_r]$ VP_VBZ$[verb]$, where:
  - p(VP_VBZ$[verb]$|NP$[noun_r]$=SUBJ) $> \alpha$

- Note in this case the system can generate intransitive forms of verbs.

With the exception of the copula rule, the given rules roughly correspond to the more likely grammar rules from a PCFG induced over the ImageClef data, discussed in **KARL'S SECTION**. We aim to incorporate a PCFG in future work. This will also help to expand the kinds of structures the system can generate.

Each of the listed rules interact with the vision system and the knowledge base of word co-occurrence statistics, returning those items provided by the vision system that are most probable (above the threshold α) based on the knowledge base.

The Mutual Exclusivity Hypothesis. For the selection of adjectives, we utilize the idea that for a given attribute (COLOR, SIZE, etc.), only a single value will be generated in the final description. Although this is clearly not always true (`red`, `white` and `blue` are three values of COLOR that may appear in a single description), it is true most of the time, and serves as a reasonable working assumption while we are focusing on generating descriptions of whole objects, and not their parts. Table 39 illustrates how well this hypothesis captures the data, listing how often an attribute of the same type occurs more than once in a noun phrase.

Attribute	COLOR	SIZE	MATERIAL	DIRECTION	PATTERN	SHAPE
Count	6388	190	111	66	15	6
Relative Frequency	0.016	0.001	<0.000	<0.000	<0.000	<0.000

Table 39: Relative frequency of different attributes occurring more than once in multi-modifier noun phrases.

Different attributes appear to be more mutually exclusive than others. For example, SHAPE values basically never co-occur in a phrase, but COLOR values may. Such features may be useful in further work on identifying and classifying visually descriptive text.

Using this hypothesis, we leverage the knowledge base to find the attributes of detected values. Values that belong to the same attribute can then be compared. This allows us to filter the noise of the vision output by selecting only the value that is of the highest probability for the given noun. For example, the probability of *blue* given *lake* is higher than the probability of *black* given *lake*. Both *blue* and *black* may fire within an object detection for *lake*, but because the knowledge base indicates that both are values for the same attribute COLOR, the model can select one over the other during generation. Currently, we select the more probable adjective for a given attribute based on the word co-occurrence statistics from the knowledge base, but we may able to use the vision scores as well in future work. With this in place, we can now begin to generate more complex noun phrases.

Hallucination. Verbs are not always provided from detections by the vision system; often, no verb forms (action, pose) fire. In these cases, and in the cases where only one object detection has fired, we can *hallucinate*, generating the more probable verbs above a threshold and continuing with the construction of the tree as explained in rules 3–7 above. In future work, hallucination may also be applied to other kinds of open-class forms, such as adjectives.

Prepositions. A subset of possible prepositions can be derived by comparing the x and y axes of the bounding boxes of recognized objects. This comparison allows the system to know whether an object is *over* another, *by* another, of *in* another. By the end of the workshop, the generation system had not yet been provided with the preposition function developed in [45], but presumably such functionality may be easily added.

Within these three coarse-grained spatial relations, we define a subset of possible values, listed in Table 40 below. The system then uses the word co-occurrence statistics to generate only those prepositions within the chosen subset above the probability threshold, discussed in Rule 2 above. This approach was used in the final presentation of the generation system, generating descriptions of PASCAL images based on the input used by the Baby Talk system [45].

a over b	a over b	a above b	a on b	b under a	b underneath a	b beneath a
	b below a	a upon b				
a by b	a against b	b against a	a on b	b on a	a near b	b near a
	a by b	a next to b	a beside b	a with b	b by a	b next to a
	b beside a	b with a	b around a	a around b		
a in b	a in b	a within b	b outside of a	a inside of b		

Table 40: Possible prepositions from bounding boxes.

5.1.4 Surface Realization

In the surface realization stage, which has not yet been implemented, we would like to select a subset of the generated structures based on ngram ranking. For now, the system returns the full set of descriptions: all descriptions that are generated from the above pipeline. The full set can also be used to capture speaker variation, e.g., to generate a different caption for different speaker profiles, or to generate a specific kind of construction depending on other types of constraints.

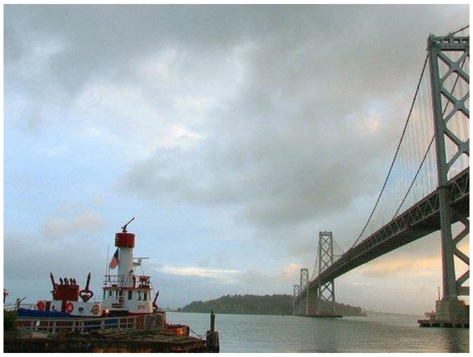
5.1.5 Results

Clearly, the effectiveness of the generation system is dependent on the effectiveness of the computer vision system. When we know the objects in the image, and thus what detectors to run, the generation system does quite well. For the PASCAL images with reasonable detections, we also do quite well. Examples are listed below, along with the full set of generated phrases from the system. Each generated sentence has a vector of probabilities associated to it (the probabilities of each combination made during tree growth), as well as part of speech tags for each word. These sentences can be used to further prune and optimize based on ngram ranking, or to tune different outputs to capture speaker variation.

Generation from Flickr Images with Objects Predefined.

This is the data the generation system was developed on, a set of images where detectors were run based on a set of predefined objects. The task of defining the set of object detectors that should be run on an image is still an active area of research.

Each set of sentences is generated with α set to .025 and an observation cutoff (how many times a given combination must be seen) of 10.



Input:

```
- label: "boat",
  attrs: {’black’:5.1553e-09,’blue’:0.080288,’brown’:1.0436e-09,’green’:0.0072417,
’orange’:0.0088653,’pink’:0.012285,’red’:0.016118,’white’:0.16355,’yellow’:0.02126,
’gray’:0.14643,’golden’:0.094969,’colorful’:4.8442e-07,’clear’:0.32307,’dirty’:0.0097018,
’feathered’:0.020314,’furry’:0.040699,’rectangular’:1.5298e-08,’rusty’:0.0052295,
’shiny’:0.010719,’striped’:9.4852e-06,’wooden’:0.031852}
  id: 717, type: 1, label: "boat", score: -0.90447, post_id: 52
  bbox: [37.377200,195.011700,145.508800,254.640400]
  img_size: [500,375]
```

```
- label: "boat",
  attrs: {’black’:0.011771,’blue’:0.037625,’brown’:6.4371e-08,’green’:0.0063443,
’orange’:0.015943,’pink’:7.1085e-06,’red’:0.0031089,’white’:0.1829,
’yellow’:0.01426,’gray’:0.37588,’golden’:0.39624,’colorful’:0.04802,
’clear’:0.0026245,’dirty’:0.029647,’feathered’:0.04233,’furry’:0.025376,
’rectangular’:1.7385e-05,’rusty’:0.0073565,’shiny’:0.05733,
’striped’:0.037917,’wooden’:0.069839}
  id: 718, type: 1, label: "boat", score: -0.9067, post_id: 52
  bbox: [92.895900,221.550100,202.170900,367.583500]
  img_size: [500,375]

- label: "boat",
  attrs: {’black’:0.013378,’blue’:0.11247,’brown’:0.066426,’green’:0.016305,
’orange’:0.0097203,’pink’:6.5082e-09,’red’:0.0044914,’white’:0.075547,
’yellow’:0.0060668,’gray’:0.37513,’golden’:0.03373,’colorful’:0.017711,
’clear’:0.0030455,’dirty’:0.038692,’feathered’:0.045265,’furry’:0.030119,
’rectangular’:0.012487,’rusty’:0.022829,’shiny’:0.018442,
’striped’:0.0227,’wooden’:0.009238}
  id: 719, type: 1, label: "boat", score: -0.93214, post_id: 52
  bbox: [421.135200,335.393300,497.302900,375.000000]
  img_size: [500,375]

- label: "boat",
  attrs: {’black’:0.0082378,’blue’:0.046733,’brown’:7.1628e-07,’green’:1.8228e-05,
’orange’:0.16988,’pink’:1.7385e-07,’red’:8.1592e-06,’white’:0.52339,
’yellow’:8.2312e-06,’gray’:0.23086,’golden’:0.011526,’colorful’:0.021428,
’clear’:8.6288e-07,’dirty’:0.025727,’feathered’:0.075628,’furry’:0.049814,
’rectangular’:0.0030995,’rusty’:0.019861,’shiny’:0.11319,
’striped’:0.11566,’wooden’:0.027919}
  id: 720, type: 1, label: "boat", score: -0.93956, post_id: 52
  bbox: [395.412600,69.593500,497.302900,205.780500]
  img_size: [500,375]

- label: "boat",
  attrs: {’black’:7.0335e-07,’blue’:0.16061,’brown’:3.0079e-10,’green’:0.0084284,
’orange’:0.025592,’pink’:0.0046916,’red’:0.01032,’white’:0.066322,
’yellow’:0.018796,’gray’:0.23895,’golden’:0.11488,’colorful’:0.0030058,
’clear’:0.029104,’dirty’:0.013877,’feathered’:0.0247,’furry’:0.031513,
’rectangular’:5.887e-07,’rusty’:0.0054022,’shiny’:0.017322,
’striped’:0.010807,’wooden’:0.02951}
  id: 721, type: 1, label: "boat", score: -0.95362, post_id: 52
  bbox: [1.000000,159.391900,135.764500,339.411300]
  img_size: [500,375]

- label: "boat",
  attrs: {’black’:0.03392,’blue’:0.034742,’brown’:7.1475e-06,’green’:0.0035864,
’orange’:0.41119,’pink’:3.2541e-08,’red’:3.4253e-06,’white’:0.20538,
’yellow’:4.4664e-06,’gray’:0.62888,’golden’:0.02206,’colorful’:0.024072,
’clear’:2.5962e-06,’dirty’:0.058585,’feathered’:0.1178,’furry’:0.030533,
’rectangular’:0.0089058,’rusty’:0.030297,’shiny’:0.12033,
’striped’:0.098637,’wooden’:0.024063}
  id: 722, type: 1, label: "boat", score: -0.95777, post_id: 52
  bbox: [338.794000,43.224300,500.000000,375.000000]
  img_size: [500,375]

- label: "boat",
  attrs: {’black’:0.010862,’blue’:0.022072,’brown’:1.6589e-07,’green’:0.0055498,
’orange’:0.027182,’pink’:4.6359e-06,’red’:0.0026992,’white’:0.11829,
’yellow’:0.011899,’gray’:0.35218,’golden’:0.52388,’colorful’:0.043474,
’clear’:0.0026679,’dirty’:0.031999,’feathered’:0.046301,’furry’:0.018008,
’rectangular’:0.0037991,’rusty’:0.0098905,’shiny’:0.093787,
’striped’:0.042076,’wooden’:0.04176}
  id: 723, type: 1, score: -0.97337, post_id: 52
  bbox: [60.094900,257.078000,295.474700,334.871300]
  img_size: [500,375]
```

```
- label: "bridge"
  attrs: {’black’:0.014172,’blue’:0.017965,’brown’:3.5358e-06,’green’:0.0039058,
’orange’:0.12258,’pink’:3.3388e-06,’red’:1.7742e-05,’white’:0.29989,
’yellow’:0.007551,’gray’:0.32111,’golden’:0.59479,’colorful’:0.013931,
’clear’:0.0048013,’dirty’:0.034278,’feathered’:0.049823,’furry’:0.022883,
’rectangular’:0.0033548,’rusty’:0.014698,’shiny’:0.047594,
’striped’:0.044412,’wooden’:0.030621}
  id: 724, type: 0, score: -0.87932, post_id: 52
  bbox: [1.000000,46.254800,497.803200,362.038700]
  img_size: [500,375]
```

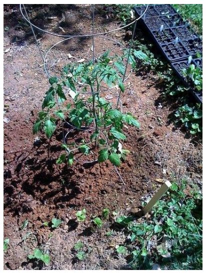
For the sake of brevity, we show outputs with ‘a’ and ‘the’ on one line.

Output:

a/the boat on a/the wooden golden bridge
a/the boat by a/the wooden golden bridge
a/the boat going over a/the wooden golden bridge
a/the boat going by a/the wooden golden bridge
a/the boat with a/the wooden golden bridge
a/the boat going under a/the wooden golden bridge
a/the boat near a/the wooden golden bridge
a/the boat under a/the wooden golden bridge
a/the boat going on a/the wooden golden bridge
a/the boat going a/the wooden golden bridge
a/the boat on a/the bridge
a/the boat by a/the bridge
a/the boat going over a/the bridge
a/the boat going by a/the bridge
a/the boat with a/the bridge
a/the boat going under a/the bridge
a/the boat near a/the bridge
a/the boat under a/the bridge
a/the boat going on a/the bridge
a/the boat going a/the bridge
a/the boat on a/the golden bridge
a/the boat by a/the golden bridge
a/the boat going over a/the golden bridge
a/the boat going by a/the golden bridge
a/the boat with a/the golden bridge
a/the boat going under a/the golden bridge
a/the boat near a/the golden bridge
a/the boat under a/the golden bridge
a/the boat going on a/the golden bridge
a/the boat going a/the golden bridge
a/the wooden boat on a/the wooden golden bridge
a/the wooden boat by a/the wooden golden bridge
a/the wooden boat going over a/the wooden golden bridge
a/the wooden boat going by a/the wooden golden bridge
a/the wooden boat with a/the wooden golden bridge
a/the wooden boat going under a/the wooden golden bridge
a/the wooden boat near a/the wooden golden bridge
a/the wooden boat under a/the wooden golden bridge
a/the wooden boat going on a/the wooden golden bridge
a/the wooden boat going a/the wooden golden bridge
a/the wooden boat on a/the bridge
a/the wooden boat by a/the bridge
a/the wooden boat going over a/the bridge
a/the wooden boat going by a/the bridge
a/the wooden boat with a/the bridge
a/the wooden boat going under a/the bridge
a/the wooden boat near a/the bridge
a/the wooden boat under a/the bridge
a/the wooden boat going on a/the bridge
a/the wooden boat going a/the bridge
a/the wooden boat on a/the golden bridge
a/the wooden boat by a/the golden bridge
a/the wooden boat going over a/the golden bridge
a/the wooden boat going by a/the golden bridge
a/the wooden boat with a/the golden bridge
a/the wooden boat going under a/the golden bridge
a/the wooden boat near a/the golden bridge
a/the wooden boat under a/the golden bridge
a/the wooden boat going on a/the golden bridge
a/the wooden boat going a/the golden bridge

Figure 32: Example input and output using Flickr images, 2 objects predefined.

Input: Similar to Figure 32, but with 25 detections for ‘plant’.



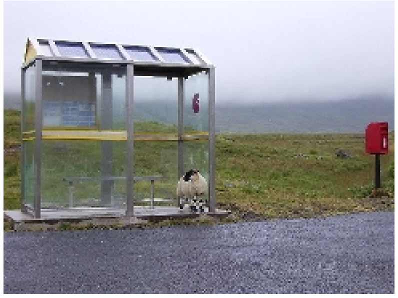
Output:

a plant growing
a plant is growing
a green plant growing
a green plant is growing
the plant growing
the plant is growing
the green plant growing
the green plant is growing

Figure 33: Example input and output using Flickr images, 1 object predefined.

Generation from PASCAL data.

Each set of sentences is generated with α set to .01 and an observation cutoff (how many times a given combination must be seen) of 2.



Input: <<gray,sky>,over,<gray,road>>,<<gray,sheep>,by,<gray,road>>
For the sake of brevity, we show outputs starting with 'a' and 'the' on one line.

Output:

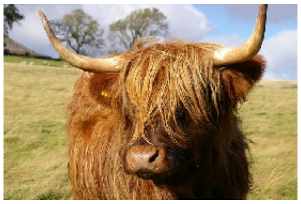
a/the sheep on a road with a gray sky
a/the sheep near a road with a gray sky
a/the sheep beside a road with a gray sky
a/the sheep by a road with a gray sky
a/the sheep on a road with the sky
a/the sheep near a road with the sky
a/the sheep beside a road with the sky
a/the sheep by a road with the sky
a/the sheep on a road with a sky
a/the sheep near a road with a sky
a/the sheep beside a road with a sky
a/the sheep by a road with a sky
a/the sheep on a road with the gray sky
a/the sheep near a road with the gray sky
a/the sheep beside a road with the gray sky
a/the sheep by a road with the gray sky
a/the sheep on the road with a gray sky
a/the sheep near the road with a gray sky
a/the sheep beside the road with a gray sky
a/the sheep by the road with a gray sky
a/the sheep on the road with the sky
a/the sheep near the road with the sky
a/the sheep beside the road with the sky
a/the sheep by the road with the sky
a/the sheep on the road with a sky
a/the sheep near the road with a sky
a/the sheep beside the road with a sky
a/the sheep by the road with a sky
a/the sheep on the road with the gray sky
a/the sheep near the road with the gray sky
a/the sheep beside the road with the gray sky
a/the sheep by the road with the gray sky
a/the sheep on that road with a gray sky
a/the sheep near that road with a gray sky
a/the sheep beside that road with a gray sky
a/the sheep by that road with a gray sky
a/the sheep on that road with the sky
a/the sheep near that road with the sky
a/the sheep beside that road with the sky
a/the sheep by that road with the sky
a/the sheep on that road with a sky
a/the sheep near that road with a sky
a/the sheep beside that road with a sky
a/the sheep by that road with a sky
a/the sheep on that road with the gray sky
a/the sheep near that road with the gray sky
a/the sheep beside that road with the gray sky
a/the sheep by that road with the gray sky
a/the sheep on this road with a gray sky
a/the sheep near this road with a gray sky
a/the sheep beside this road with a gray sky
a/the sheep by this road with a gray sky
a/the sheep on this road with the sky
a/the sheep near this road with the sky
a/the sheep beside this road with the sky
a/the sheep by this road with the sky
a/the sheep on this road with a sky
a/the sheep near this road with a sky
a/the sheep beside this road with a sky
a/the sheep by this road with a sky
a/the sheep on this road with the gray sky
a/the sheep near this road with the gray sky
a/the sheep beside this road with the gray sky
a/the sheep by this road with the gray sky

Figure 34: Example input and output using PASCAL images, 3 objects recognized.

In Figure 34, the generation system limits the *gray* detection for *sheep*, generating references to the sheep without an adjective; this reflects the fact that sheep, although visually gray, tend to be described as *white*. The language model does not have a high probability of ($gray|sheep$), and so does not generate this descriptor – but it does have a high probability of ($white|sheep$), which was not detected. Further work may link the vision output to these kinds of object-based color lexicalization preferences.

Input:

<<golden,cow>,by,<blue,sky>>



Output:

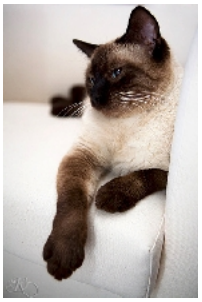
the golden cow against the sky
the golden cow with the sky
the golden cow against a blue sky
the golden cow with a blue sky
the golden cow against a sky
the golden cow with a sky
the golden cow against the blue sky
the golden cow with the blue sky
a cow against the sky
a cow with the sky
a cow against a blue sky
a cow with a blue sky
a cow against a sky
a cow with a sky
a cow against the blue sky
a cow with the blue sky

a golden cow against the sky
a golden cow with the sky
a golden cow against a blue sky
a golden cow with a blue sky
a golden cow against a sky
a golden cow with a sky
a golden cow against the blue sky
a golden cow with the blue sky
the cow against the sky
the cow with the sky
the cow against a blue sky
the cow with a blue sky
the cow against a sky
the cow with a sky
the cow against the blue sky
the cow with the blue sky

Figure 35: Example input and output using PASCAL images, 2 objects recognized.

Input:

<<cat>„<>>



Output:

the cat lives
the cat sitting
the cat is sitting
the cat sleeping
the cat is sleeping
a cat lives
a cat sitting
a cat is sitting
a cat sleeping
a cat is sleeping

Figure 36: Example input and output using PASCAL images, 1 object recognized.

In Figure 35, we begin to see sensible, poetic descriptions, such as *the golden cow against the blue sky*. As in the figure above, *sky* takes the indefinite article *a*, because the model learns article preferences without taking into account an intervening adjective; p($a|sky, blue$) is high, but we are marginalizing over adjectives to get a high p($a|sky$). It is straightforward to develop the system further to take the joint probability into account, which will further refine these outputs.

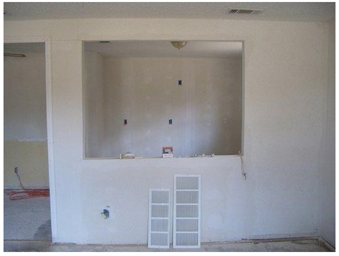
In Figure 36, a single detection is turned into a reasonable (albeit somewhat existential) set of descriptive captions. This is done using the *hallucination* aspect of the generation system, which returns probable verbs when only a single detection fires.

Note that the input for the generation system in these descriptions is significantly different from the input for the generation system as it was developed on the raw visual output. Instead of bounding boxes and detections with scores, it is a series of <<adj,obj>,prep<adj,obj>> triples, returned from the system described in [45]. The actual vision processing is therefore a 'black box' for the generation system, and the generation system does not evaluate competing detections. Both kinds of inputs are accepted by the system. However, some of the generation mechanisms do not get used when given this pre-processed input, e.g., values within a single attribute class are not compared.

Generation from All Detectors on Random Image.

We also ran all object detectors on the Flickr data, and checked how well the system did with the more raw data. This is an end-to-end system from novel images to image descriptions. In these cases, we use the nPMI approach discussed in 5.1.2 to determine which objects to describe. Vision scores are not used.

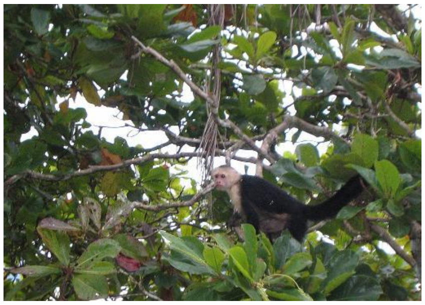
Input: 84 detections for trains, dogs, flowers, what have you.



Output:

a laptop on a chair
a laptop in a chair
a laptop on the chair
a laptop in the chair
the laptop on a chair
the laptop in a chair
the laptop on the chair
the laptop in the chair

Figure 37: Unconstrained vision-to-language generation.



a) monkey playing in the tree canopy, Monte Verde in the rain forest

b) capuchin monkey in front of my window

c) monkey spotted in Apenheul Netherlands under the tree

d) a white-faced or capuchin in the tree in the garden

e) the monkey sitting in a tree, posing for his picture

Figure 38: **Grab 'N Mash Generation:** an example query photograph with 4 automatically generated captions and the original caption associated with the image by the photo owner. Could you guess which is the true image caption – b? Generated captions are often quite realistic and relevant to the photo content.

As can be seen in Figure 37...There may be some more work to do.

5.1.6 Future Work

Future work has been mentioned throughout this section.

A further possibility is to convert the hand-coded rules into features within a log-linear model. The process of building a hand-coded algorithm can bring useful features to light for a statistical model [62], which may perform even better than the hand-coded approach. We can then compare the statistical method to the current hand-coded method and see how each performs.

Another area for work includes learning different probability thresholds for different rules; currently, subtrees are created based on combinations of nodes above a single blanket threshold α.

5.2 grab & mash

6 Grab 'N Mash Caption Generation

Producing a relevant and accurate caption for an arbitrary image is an extremely challenging problem, perhaps nearly as difficult as the underlying general image understanding task. However, there are many images with relevant associated descriptive text available in the noisy vastness of the web. The key is to find the right images and make use of them in the right way!

Man sits in a rusted car buried in the sand on Waitarere beach

Little girl and her dog in northern Thailand. They both seemed interested in what we were doing

Interior design of modern white and brown living room furniture against white wall with a lamp hanging.

The Egyptian cat statue by the floor clock and perpetual motion machine in the pantheon

Our dog Zoe in her bed

Emma in her hat looking super cute

Figure 39: **SBU Caption Dataset:** example photographs with user-associated captions from the SBU Caption dataset, a web-scale collection containing 1 million captioned photographs.

In this project, we present a method for captioning photographs that makes use of the enormous number of images on the web with associated visually descriptive text. We follow in the footsteps of past work on internet vision that has demonstrated that big data can often make big problems – like image localization [41], retrieving photos with specific content [77], or image parsing [76] – much more bite size and approachable by very simple matching methods. In our case, with a large enough captioned photo collection we can automatically generate relevant and human-sounding image descriptions surprisingly well. Figure 38 shows 5 captions, 4 that were automatically generated by our method and 1 that was written by the owner of the photograph. Our resulting generated captions are often quite realistic and relevant to the photo content.

In particular, for a query image, our method:

- detects or classifies local and global image content
- finds similar content within a large data base of captioned photographs
- transfers phrases from the matched photographs to the query photo
- merges the transferred phrases into a set of complete image descriptions

In the remainder of this section we describe: previous related work (sec 6.1) on image captioning, the data we use for generation (sec 6.2), our methods for finding similar content and *grabbing phrases* from their associated captions (sec 6.3), and finally we present methods for *mashing* these phrases into complete resulting image captions (sec 6.4).

6.1 Related Work

Studying the association between words with pictures has been explored in a variety of tasks, including: labeling faces in news photographs with associated captions [6, 5], finding a correspondence between keywords and image regions [3, 20], or for moving beyond objects to mid-level recognition elements such as attribute [46, 24, 48, 32] or prepositions [40].

Image description generation in particular has been studied in a few recent papers [25, 30, 45, 89]. Kulkarni et al [45] generate descriptions from scratch based on detected object, attribute, and prepositional relationships. This results in descriptions for images that are usually closely related to image content, but that are also often quite verbose and non-humanlike. Yao et al [89] look at the problem of generating text using various hierarchical knowledge ontologies and with a human in the loop for image parsing (except in specialized circumstances). Feng and Lapata [30] generate captions for images using extractive and abstractive generation methods, but assume relevant documents are provided as input; Aker et al [2] rely on GPS meta data to access relevant text documents, whereas our generation method requires only an image as input.

A recent approach from Farhadi et al [25] produces image descriptions via a retrieval method, by translating both images and text descriptions to a shared meaning space represented by a single $< object, action, scene >$ tuple. A description for a query image is produced by retrieving whole image descriptions via this meaning space from a set of image descriptions (the UIUC Pascal Sentence data set). This results in descriptions that are very human – since they were written by humans – but which may not be relevant to the specific image content. This limited relevancy often occurs because of problems of sparsity, both in the data collection – 1000 images is too few to guarantee similar image matches – and in the representation – only a few categories for 3 types of image content are considered.

Compared to these previous approaches, we attack the caption generation problem for much more general images (images found via thousands of Flickr queries compared to 1000 images from Pascal) and a larger set of object categories (89 vs 20). In addition to using a much larger set of object categories than previous approaches, we also include a wider variety of image content aspects, including: non-part based stuff categories, attributes of objects, person specific action models, and a larger number of common scene classes. Our descriptions also have access to via an extractive method with access to much larger and more general set of captioned photographs from the web (1 million vs 1k). This project extends our previous approach for non-parametric caption generation [65] which transferred whole captions from matching images by exploring techniques to transfer and merge bits of text (phrases) from matching images to a query image.

6.2 Data

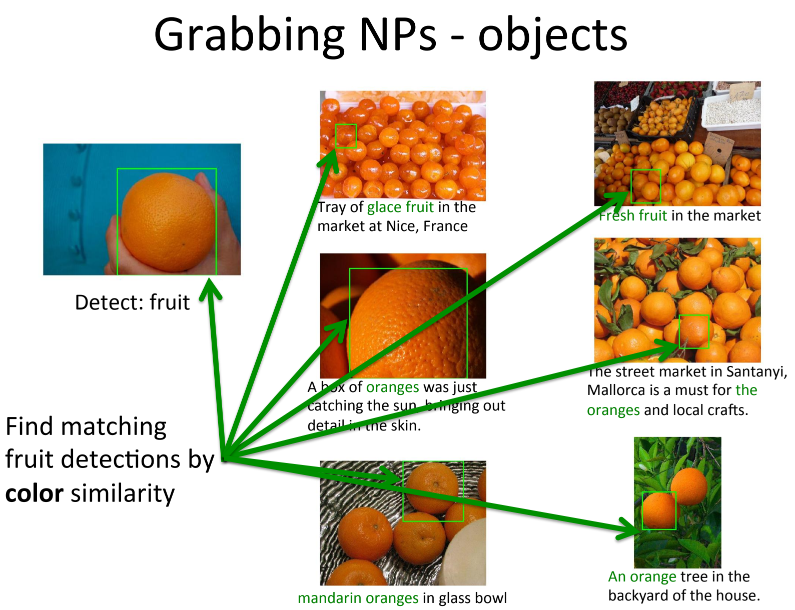
For this approach we make use of the SBU captioned photo data set [65], recently collected by Ordonez, Kulkarni, and (team-leader) T. Berg. To be useful for non-parametric caption generation, this database must be good in two ways: 1) It must be large so that image based matches to a query are reasonably similar, 2) The captions associated with the data base photographs must be relevant so that transferring captions between pictures is useful. To achieve the first requirement Ordonez *et al.* queried Flickr using a huge number of pairs of query terms (objects, attributes, actions, stuff, and scenes). This produces a very large initial set of photographs with associated text. To achieve our second requirement they then filtered this set of photos so that the descriptions attached to a picture are relevant and visually descriptive. To encourage visual descriptiveness in our collection, they selected only those images with descriptions of satisfactory length based on observed lengths in visual descriptions. They also enforced that retained descriptions contain at least 2 words belonging to our term lists or that are prepositional words, e.g. “on”, “under” which often indicate spatial relationships.

The resulting data set contains 1 million photographs with associated descriptions, most of which are quite relevant to the attached photo. Some examples are shown in fig 39. An additional useful property of this data set is that the captions associated with images were written by people. This means that if we generate novel captions from this text, it is likely to sound natural and human-like – because the original captions were written by people.

On this large data collection we have run many different visual recognition algorithms to detect or classify content elements (sec 4). For this part of the project we currently make use of a subset of these results: object category (e.g. horse or person) detection, stuff category (e.g. grass or road) detection, and scene type (e.g. market or living room) classification.

6.3 Grabbing Relevant Phrases

For a query image, we would like to retrieve phrases that are relevant to the visual content of the image. We do this by matching the content the query to our large data base of captioned photographs (sec 6.2) and finding similar images. Three kinds of content elements are used for similar image retrieval: local object (e.g. fruit or ball), or stuff (e.g. sky or water) detections, and global scene (e.g. kitchen or pasture) classifications. Given a query, we detect or classify the content elements



Grabbing NPs - objects

Detect: fruit

Find matching fruit detections by **color** similarity

Tray of glace fruit in the market at Nice, France

Fresh fruit in the market

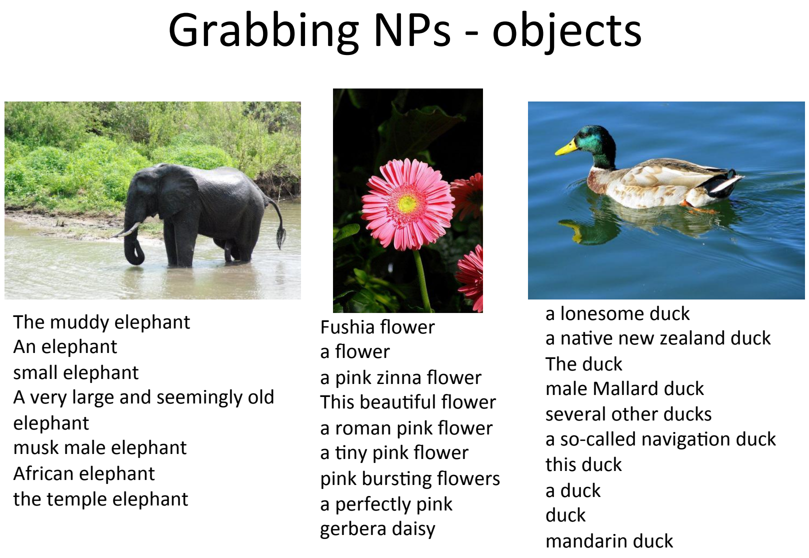
A box of oranges was just catching the sun, bringing out detail in the skin.

The street market in Santanyi, Mallorca is a must for the oranges and local crafts.

mandarin oranges in glass bowl

An orange tree in the backyard of the house.

Figure 40: **Grabbing Noun Phrases from Object Matches:** an example object detection (fruit) and retrieved matching images using color similarity. Noun phrases (green) from matched images can be transferred to the query image.



Grabbing NPs - objects

The muddy elephant
An elephant
small elephant
A very large and seemingly old elephant
musk male elephant
African elephant
the temple elephant

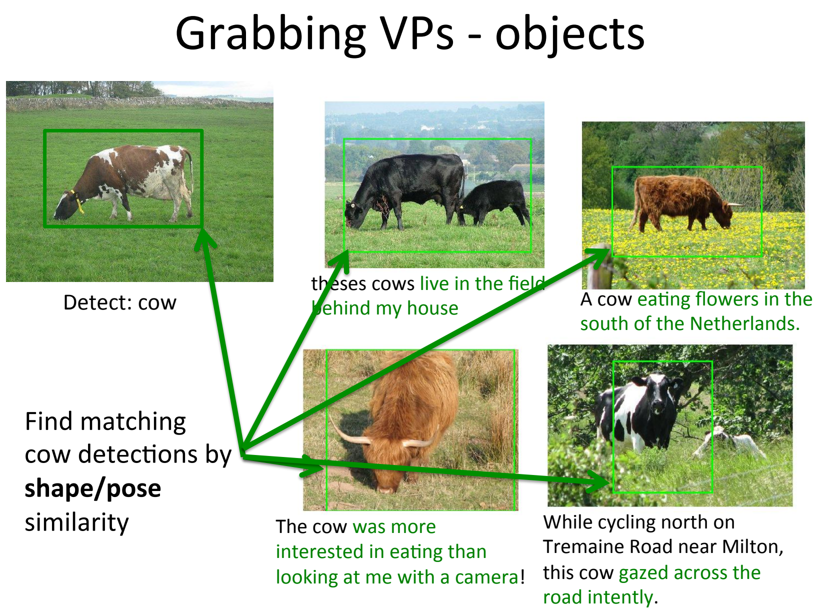
Fushia flower
a flower
a pink zinna flower
This beautiful flower
a roman pink flower
a tiny pink flower
pink bursting flowers
a perfectly pink gerbera daisy

a lonesome duck
a native new zealand duck
The duck
male Mallard duck
several other ducks
a so-called navigation duck
this duck
a duck
duck
mandarin duck

Figure 41: **Grabbing Noun Phrases from Object Matches:** some example noun phrase transfer results.

present in the image, retrieve images from the data base that are visually similar relative to one of the content elements, and then "grab" phrases from the retrieved photos' captions.

Object Phrase Retrieval: We use object detections to retrieve 2 kinds of phrases – noun phrases (NP) describing the visual appearance of an object, and verb phrases (VP) describing the actions undertaken by the object. Noun phrases are retrieved by matching the color appearance of object detections. This is accomplished by computing color histograms of detected objects within the query image and comparing them to color histograms computed on detected objects of the same category



Grabbing VPs - objects

Detect: cow

theses cows live in the field behind my house

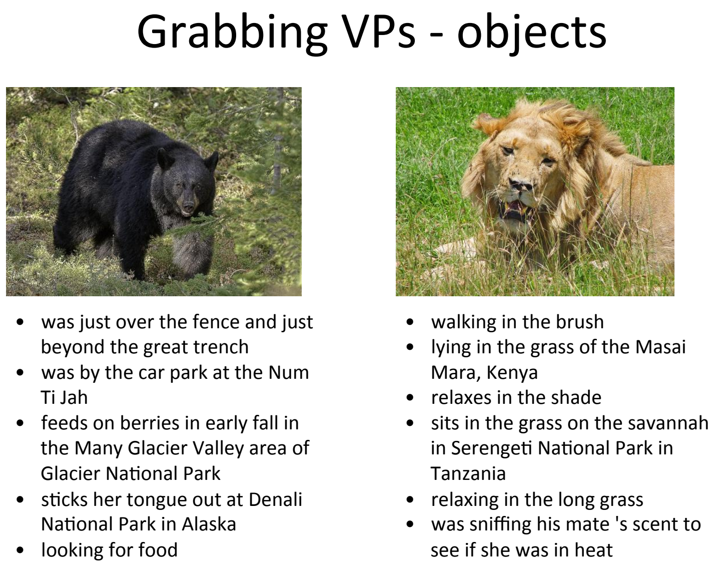
A cow eating flowers in the south of the Netherlands.

Find matching cow detections by **shape/pose** similarity

The cow was more interested in eating than looking at me with a camera!

While cycling north on Tremaine Road near Milton, this cow gazed across the road intently.

Figure 42: **Grabbing Verb Phrases from Object Matches:** an example object detection (cow) and retrieved matching images using shape/pose similarity. Verb phrases (green) from matched images can be transferred to the query image.



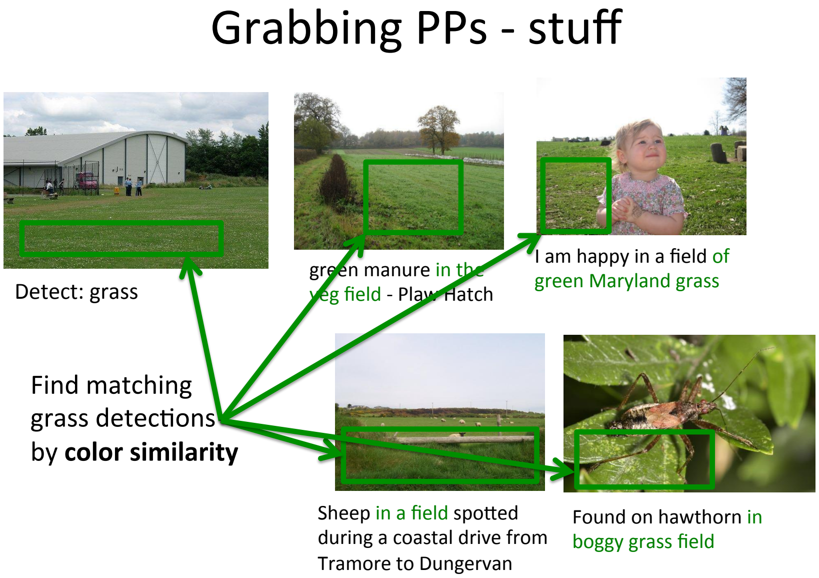
Grabbing VPs - objects

- was just over the fence and just beyond the great trench
- was by the car park at the Num Ti Jah
- feeds on berries in early fall in the Many Glacier Valley area of Glacier National Park
- sticks her tongue out at Denali National Park in Alaska
- looking for food

- walking in the brush
- lying in the grass of the Masai Mara, Kenya
- relaxes in the shade
- sits in the grass on the savannah in Serengeti National Park in Tanzania
- relaxing in the long grass
- was sniffing his mate 's scent to see if she was in heat

Figure 43: **Grabbing Verb Phrases from Object Matches:** some example verb phrase transfer results.

in the data base. An example for a fruit detection is shown in figure 40. In the query image, a fruit has been detected. Retrieved images match well in color and their referring noun phrases (NPs that have the query category as their head noun) can be transferred to the query. Note, we make use of hyponyms (e.g. orange) to find relevant object detections and phrases. More noun phrase transfer results are shown in fig 41. Similarly, verb phrases are transferred to the query by matching shape descriptors (e.g. visual word SIFT histograms [52]) between a query object detection and object detections in the data base. Figure 42 shows an example cow detection, retrieved matching images


Grabbing PPs - stuff

Detect: grass

green manure in the veg field - Plaw Hatch

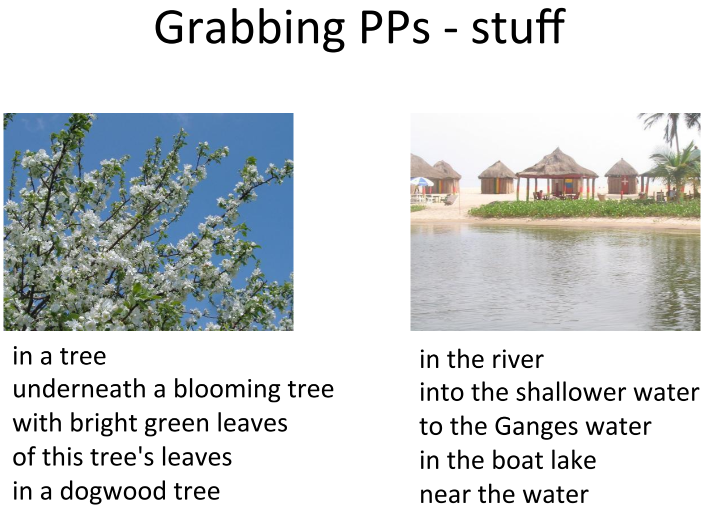
I am happy in a field of green Maryland grass

Find matching grass detections by **color similarity**

Sheep in a field spotted during a coastal drive from Tramore to Dungervan

Found on hawthorn in boggy grass field

Figure 44: **Grabbing Prepositional Phrases from Stuff Matches:** an example stuff detection (grass) and retrieved matching images using color similarity. Prepositional phrases (green) from matched images can be transferred to the query image.


Grabbing PPs - stuff

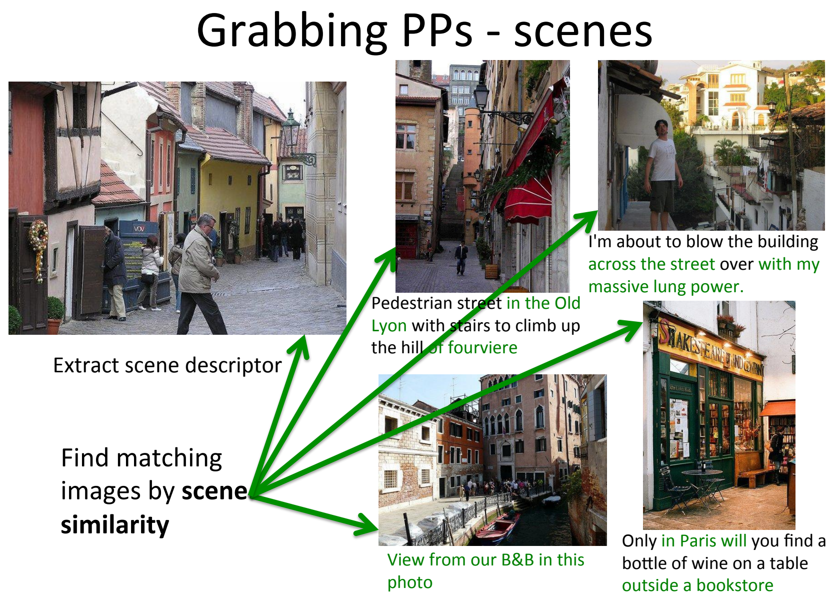
in a tree
underneath a blooming tree
with bright green leaves
of this tree's leaves
in a dogwood tree

in the river
into the shallower water
to the Ganges water
in the boat lake
near the water

Figure 45: **Grabbing Prepositional Phrases from Stuff Matches:** some example prepositional (stuff) phrase transfer results.

using object shape similarity, and transferable verb phrases (green). Additional examples are shown in figure 43.

Stuff Phrase Retrieval: We use stuff detections to retrieve prepositional phrases (PP) describing the prepositional relationship of an object with a stuff (e.g. water) detection. These denote things like “in the green field” and provide spatial context or relationships between object detections and their surroundings. For this retrieval we also use color histogram similarity between stuff detections in the query image and stuff detections of the same category in the data base. An example prepositional



Grabbing PPs - scenes

Extract scene descriptor

Find matching images by **scene similarity**

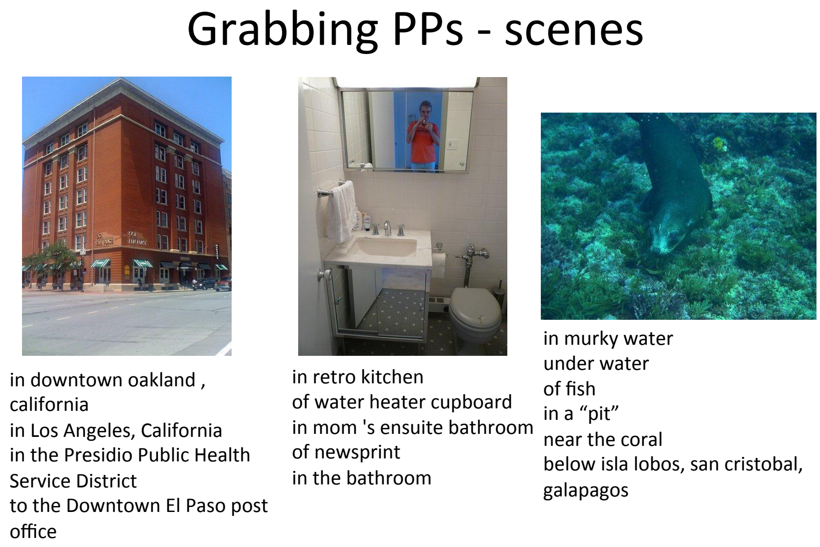
Pedestrian street in the Old Lyon with stairs to climb up the hill of fourviere

I'm about to blow the building across the street over with my massive lung power.

View from our B&B in this photo

Only in Paris will you find a bottle of wine on a table outside a bookstore

Figure 46: **Grabbing Prepositional Phrases using Scene Similarity:** an example retrieved image matches using global scene descriptors. Prepositional phrases (green) from matched images can be transferred to the query image.



Grabbing PPs - scenes

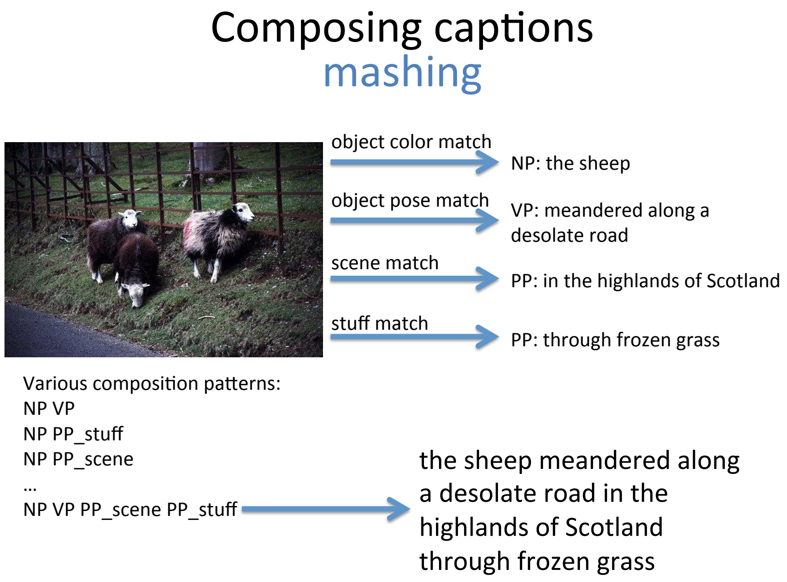
in downtown oakland , california
in Los Angeles, California
in the Presidio Public Health Service District
to the Downtown El Paso post office

in retro kitchen
of water heater cupboard
in mom 's ensuite bathroom
of newsprint
in the bathroom

in murky water
under water
of fish
in a "pit"
near the coral
below isla lobos, san cristobal, galapagos

Figure 47: **Grabbing Prepositional Phrases using Scene Similarity:** some example prepositional (scene) phrase transfer results.

phrase transfer is shown in figure 44, where a grass detection in a query image is matched to similar looking grass detections in the data set. We could transfer prepositional phrases (green) from the matched image to the query. Additional examples are shown in figure 45. Note, ideally prepositional phrase transfer should reflect spatial relationships between an object and a stuff detection. Currently we only use color similarity to find matches within the data base. We did explore some use of spatial similarity at the workshop, but leave further explorations for future work.



Composing captions
mashing

object color match → NP: the sheep

object pose match → VP: meandered along a desolate road

scene match → PP: in the highlands of Scotland

stuff match → PP: through frozen grass

Various composition patterns:
NP VP
NP PP_stuff
NP PP_scene
...
NP VP PP_scene PP_stuff → the sheep meandered along a desolate road in the highlands of Scotland through frozen grass

Figure 48: **Mashing Phrases to Generate Captions:** Given a query image and retrieved relevant phrases. We can compose captions through a set of simple compositional rules. For example a retrieved NP-VP pair can produce a reasonable caption. We produce a query image caption set using a variety of reasonable composition rules.

Good results

A duck was having a bath in the harbor at whitehaven, cumbria, england in the water near Camley St

A female Monarch butterfly was visiting the plant in my front yard in Devon 17/10/10

Stained glass window depicting Christ and numerous saints in Washington National Cathedral in the Eglise

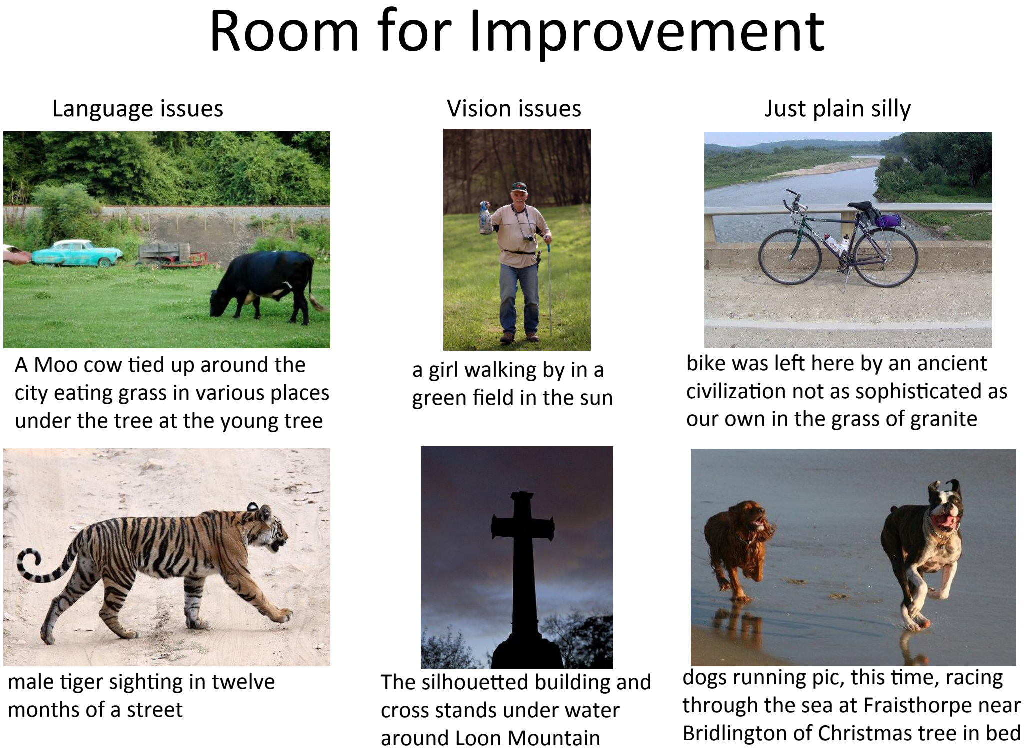
her flower girl dress designed by Mainbocher in the house

A double-decker bus under some spreading shade trees

cat enjoys hiding under the tree

Figure 49: **Mashing Phrases to Generate Captions:** some example good caption generation results.

Scene Phrase Retrieval: We use scene classifiers to retrieve general prepositional phrases describing an image. These should denote general characteristics that might be relevant to an image, for



Room for Improvement

Language issues

Vision issues

Just plain silly

A Moo cow tied up around the city eating grass in various places under the tree at the young tree

a girl walking by in a green field in the sun

bike was left here by an ancient civilization not as sophisticated as our own in the grass of granite

male tiger sighting in twelve months of a street

The silhouetted building and cross stands under water around Loon Mountain

dogs running pic, this time, racing through the sea at Fraisthorpe near Bridlington of Christmas tree in bed

Figure 50: **Mashing Phrases to Generate Captions:** some example caption generation results that are less than ideal. Our simple grab 'n mash generation method doesn't always work and can fail in a number of ways from language issues, to incorrect visual recognition detections, to just plain silly captions.

example "at the market" or "in Paris". We use a global scene descriptor for this similarity measure. For a query image, a number of scene classifiers are used to extract a scene descriptor, a vector of probabilities for a number of common scene categories. This scene descriptor is compared to scene descriptors calculated on our image data base and the most similar images are retrieved. Similarly to our stuff based retrieval, this is also used to attach relevant prepopositional phrases to a query image. Figure 46 shows an example query image, matched images according to our global scene similarity measure, and prepositional phrases that could be transferred to the query (green). Figure 47 shows some additional prepositional phrase transfers.

6.4 Mashing Phrases into Captions

We use a very simple method to compose image captions from our retrieved relevant phrases. This generation method uses a small set of compositional rules to generate captions. For example, captions can be generated using a rule that concatenates a retrieved NP with a retrieved VP. We use a number of such composition rules to generate a proposed caption set for query images. One example result using the (NP - VP - PP stuff - PP scene) rule is shown in figure 48. Often this produces quite nice captions for images (fig 49 shows some reasonable results). Image captioning is an extremely challenging problem. Sometimes our simple grab 'n mash strategy fails for a number of different reasons (fig 50 shows some failure cases), including language generation issues, incorrect visual recognition detections, to producing somewhat silly captions.

However, often our captioning method performs quite well and can capture some appearance characteristics of objects in the image (e.g. double-decker bus or Monarch butterfly) despite only having knowledge of object categories. It also produces quite human-sounding captions (e.g. "A duck was having a bath in the harbor at whitehaven, cumbria, england in the water near Camley St"). Of course sometimes these captions will not be completely correct. For example, the stained glass window is probably not in the “Washington National Cathedral in the Eglise”, but allowing such “incorrect” information in the resulting captions produces pleasing and informative captions (it probably is in a church somewhere). Future work could allow or disallow phrases like these from being produced in the final caption according to the desires and needs of the user or application.

7 Conclusion

In summary we had a productive summer, making progress on all aspects of the proposed work. The above notes describe the summer activities, and the paper submissions attached show the directions we have been following to extend the work into publishable units.

References

[1] ImageNet. http://image-net.org.

[2] A. Aker and R. Gaizauskas. Generating image descriptions using dependency relational patterns. In *Proc. of Assoc. for Computational Linguistics*, pages 1250–1258, 2010.

[3] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[4] T. Berg and D. Forsyth. Animals on the web. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[5] T. L. Berg, A. C. Berg, J. Edwards, and D.A. Forsyth. Who’s in the picture? In *NIPS*, December 2004.

[6] T. L. Berg, A. C. Berg, J. Edwards, M. Maire, R. White, E. Learned-Miller, Y. W. Teh, and D. A. Forsyth. Names and faces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[7] Tamara L. Berg, Alexander C. Berg, and Jonathan Shih. Automatic attribute discovery and characterization from noisy web data. *European Conference on Computer Vision*, 2010.

[8] T.L. Berg, A.C. Berg, J. Edwards, and D.A. Forsyth. Who’s in the picture? In *NIPS*, 2004.

[9] T.L. Berg, A.C. Berg, and J. Shih. Automatic attribute discovery and characterization. In *Proceedings of the European Conference on Computer Vision*, 2010.

[10] Lubomir Bourdev, Subhransu Maji, Thomas Brox, and Jitendra Malik. Detecting people using mutually consistent poselet activations. In *ECCV*, 2010.

[11] Elia Bruni, Giang Binh Tran, and Marco Baroni. Distributional semantics from text and images. *Proceedings of the EMNLP 2011 Geometrical Models for Natural Language Semantics (GEMS 2011) Workshop*, pages 22–32, 2011.

[12] C.-C. Chang and C.-J. Lin. Libsvm : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011.

[13] Eugene Charniak. Statistical parsing with a context-free grammar and word statistics. *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI’97)*, pages 598–603, 1997.

[14] K. Church and P. Hanks. Word Association Norms, Mutual Information and Lexicography. In *Proceedings of ACL*, pages 76–83, Vancouver, Canada, June 1989.

[15] Michael John Collins. Three generative, lexicalised models for statistical parsing. *ACL 35/EACL 8*, pages 16–23, 1997.

[16] Bob Coyne and Richard Sproat. Wordseye: an automatic text-to-scene conversion system. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH ’01, pages 487–496, New York, NY, USA, 2001. ACM.

[17] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[18] Hal Daumé III and Daniel Marcu. Learning as search optimization: Approximate large margin methods for structured prediction. In *International Conference on Machine Learning (ICML)*, 2005.

[19] 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.

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

[21] H. J. Escalante, C. Hernandez, J. Gonzalez, A. Lopez, M. Montes, E. Morales, L. E. Sucar, and M. Grubinger. The segmented and annotated iapr tc-12 benchmark. In *Computer Vision and Image Understanding*, 2009.

[22] M. Everingham, L. Van Gool, C. Williams, and A. Zisserman. Pascal visual object classes challenge results. Technical report, 2005. unpublished manuscript circulated on the web, URL is http://www.pascal-network.org/challenges/VOC/voc/index.html.

[23] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

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

[25] 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.

[26] Ali Farhadi, Ian Endres, Derek Hoiem, and David A. Forsyth. Describing objects by their attributes. *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[27] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 28, 2006.

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

[29] Pedro F. Felzenszwalb, Ross B. Girshick, David Mcallester, and Deva Ramanan. Object detection with discriminatively trained part based models. *to Appear in IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[30] Y. Feng and M. Lapata. How many words is a picture worth? automatic caption generation for news images. In *Proc. of the Assoc. for Computational Linguistics*, ACL '10, pages 1239–1249, 2010.

[31] V. Ferrari and A Zisserman. Learning visual attributes. *Proceedings of Advances in Neural Information Processing Systems*, 2007.

[32] V. Ferrari and A. Zisserman. Learning visual attributes. In *Advances in Neural Information Processing Systems*, 2007.

[33] A. Frome, Y. Singer, and J. Malik. Image retrieval and classification using local distance functions. In *Advances in Neural Information Processing Systems*, pages 417–424, 2006.

[34] A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *Proceedings of the International Conference on Computer Vision*, 2007.

[35] D. Graff. English Gigaword. Linguistic Data Consortium, Philadelphia, PA, January 2003.

[36] David Graff and Christopher Cieri. *English Gigaword*. Linguistic Data Consortium, Philadelphia, PA, 2003.

[37] Paul H. Grice. Logic and conversation. *Syntax and Semantics*, 3:41–58, 1975.

[38] G. Griffin, A.D. Holub, and P. Perona. The caltech-256. In *Caltech Technical Report*, 2006.

[39] Michael Grubinger, Paul D. Clough, Henning Mller, and Thomas Deselaers. The iapr benchmark: A new evaluation resource for visual information systems. In *International Conference on Language Resources and Evaluation*, 2006.

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

[41] J. Hays and A. Efros. im2gps: estimating geographic information from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[42] D. Hoiem, A.A. Efros, and M. Hebert. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1), October 2007.

[43] Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proceedings of ACL-08: HLT*, pages 1048–1056, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[44] Gerhard Kremer and Marco Baroni. Predicting cognitively salient modifiers of the constitutive parts of concepts. *Proceedings of the Cognitive Modeling and Computational Linguistics Workshop at ACL 2010*, pages 54–62, 2010.

[45] Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Babytalk: Understanding and generating simple image descriptions. In *CVPR*, 2011.

[46] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *Proceedings of the International Conference on Computer Vision*, 2009.

[47] N. Kumar, A.C. Berg, P.N. Belhumeur, and S.K. Nayar. Attribute and simile classifiers for face verification. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 365 –372, 29 2009-oct. 2 2009.

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

[49] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. *IEEE Computer Vision and Pattern Recognition*, 2009.

[50] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 2169–2178, 2006.

[51] Li-Jia Li, Hao Su, Eric P. Xing, and Li Fei-Fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *NIPS*, 2010.

[52] D. G. Lowe. Distinctive image features from scale invariant keypoints. *IJCV*, 2004.

[53] DG Lowe. Distinctive image features from scale-invariant keypoints. In *CVPR*, 2004.

[54] David M. Magerman. Statistical decision-tree models for parsing. *ACL 33*, pages 276–283, 1995.

[55] S. Maji and A.C Berg. Max-margin additive classifiers for detection. In *Proceedings of the International Conference on Computer Vision*, 2009.

[56] S. Maji, A.C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[57] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. In *International Journal of Computer Vision*, 2001.

[58] Tara McIntosh and James R Curran. Weighted mutual exclusion bootstrapping for domain independent lexicon and template acquisition. In *Proceedings of the Australasian Language Technology Association Workshop 2008*, pages 97–105, December 2008.

[59] Tara McIntosh and James R. Curran. Reducing semantic drift with bagging and distributional similarity. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 396–404, Suntec, Singapore, August 2009. Association for Computational Linguistics.

[60] Ken McRae. Mcrae's norms, 2011. url: http://amdrae.ssc.uwo.ca/McRaeLab/norms.php. Accessed 5.Oct.2011.

[61] G. A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[62] Margaret Mitchell, Kees van Deemter, and Ehud Reiter. Two approaches for generating size modifiers. *ENLG*, 2011.

[63] Tom Mitchell. Never-ending language learning, 2011. http://rtw.ml.cmu.edu/rtw/.

[64] Peter Norvig. How to write a spelling corrector. http://norvig.com/spell-correct.html.

[65] Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, 2011.

[66] S. Petrov, A. Faria, P. Michaillat, A.C. Berg, A. Stolcke, D. Klein, and J. Malik. Detecting categories in news video using acoustic, speech and image features. In *Proceedings of (VIDEO) TREC (TrecVid 2006)*, 2006.

[67] Slav Petrov. Berkeley parser, 2010. GNU General Public License v.2.

[68] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[69] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier. Collecting image annotations using amazon's mechanical turk. In *NAACL Workshop Creating Speech and Language Data With Amazon's Mechanical Turk*, 2010.

[70] E. Reiter and R. Dale. *Building Natural Language Generation Systems*. Cambridge University Press, 2000.

[71] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77:157–173, may 2008.

[72] H. Schmid. Improvements in part–of–speech tagging with an application to german. In *Proceedings of the EACL SIGDAT Workshop*, 1995.

[73] Merrielle Spain and Pietro Perona. Measuring and predicting object importance. In *ECCV*, 2008.

[74] R. Sproat, A. Black, S. Chen, S. Kumar, M Ostendorf, and C. Richards. Normalization of "non-standard" words. *CLSP Workshop 99 Final Report*, 1999.

[75] M. Thelen and E. Riloff. A Bootstrapping Method for Learning Semantic Lexicons Using Extraction Pattern Contexts. In *Proceedings of the Empirical Methods in Natural Language Processing*, pages 214–221, 2002.

[76] J. Tighe and S. Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. In *ECCV*, 2010.

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

[78] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1958–1970, 2008.

[79] Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *JOURNAL OF ARTIFICIAL INTELLIGENCE RESEARCH*, 37:141, 2010.

[80] `http://www.image-net.org/challenges/LSVRC/2010/`.

[81] J. van de Weijer, C. Schmid, and J. Verbeek. Learning color names from real-world images. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1 –8, june 2007.

[82] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *ICCV*, 2007.

[83] Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. The viability of web-derived polarity lexicons. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 777–785, Los Angeles, California, June 2010. Association for Computational Linguistics.

[84] Gang Wang and D. Forsyth. Joint learning of visual attributes, object classes and visual saliency. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 537 –544, 29 2009-oct. 2 2009.

[85] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[86] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[87] Keiji Yanai and Kobus Barnard. Image region entropy: a measure of ”visualness” of web images associated with one concept. In *Proceedings of the 13th annual ACM international conference on Multimedia*, MULTIMEDIA ’05, pages 419–422, New York, NY, USA, 2005. ACM.

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

[89] B.Z. Yao, Xiong Yang, Liang Lin, Mun Wai Lee, and Song-Chun Zhu. I2t: Image parsing to text description. *Proc. IEEE*, 98(8), 2010.

[90] H. Zhang, A.C. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.