Detecting Visual Text

Jesse Dodge[1], **Amit Goyal**[2], **Xufeng Han**[3], **Alyssa Mensch**[4], **Margaret Mitchell**[5], **Karl Stratos**[6]
Kota Yamaguchi[3], **Yejin Choi**[3], **Hal Daumé III**[2], **Alexander C. Berg**[3] and **Tamara L. Berg**[3]
[1]University of Washington, [2]University of Maryland, [3]Stony Brook University
[4]MIT, [5]Oregon Health & Science University, [6]Columbia University
dodgejesse@gmail.com, amit@umiacs.umd.edu, xufhan@cs.stonybrook.edu
acmensch@mit.edu, mitchmar@ohsu.edu, stratos@cs.columbia.edu
kyamagu@cs.stonybrook.edu, ychoi@cs.stonybrook.edu
me@hal3.name, aberg@cs.stonybrook.edu, tlberg@cs.stonybrook.edu

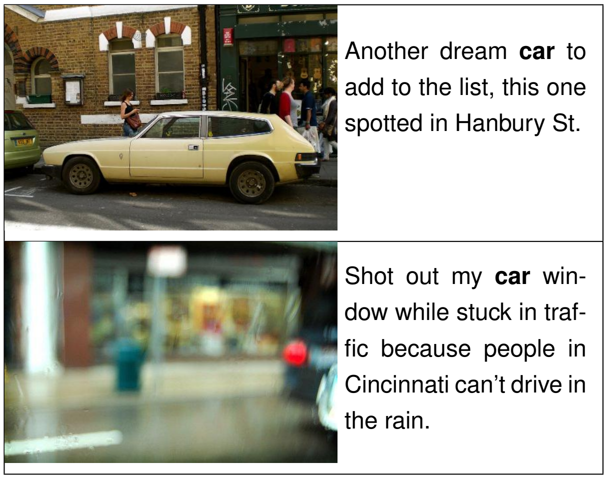
Abstract

When people describe a scene, they often include information that is not visually apparent; sometimes based on background knowledge, sometimes to tell a story. We aim to separate *visual text*—descriptions of what is being seen—from non-visual text in natural images and their descriptions. To do so, we first concretely define what it means to be visual, annotate visual text and then develop algorithms to automatically classify noun phrases as visual or non-visual. We find that using text alone, we are able to achieve high accuracies at this task, and that incorporating features derived from computer vision algorithms improves performance. Finally, we show that we can reliably mine visual nouns and adjectives from large corpora and that we can use these effectively in the classification task.

1 Introduction

People use language to describe the visual world. Our goal is to: *formalize* what "visual text" is (Section 2.2); *analyze* naturally occurring written language for occurrences of visual text (Section 2); and *build* models that can detect visual descriptions from raw text or from image/text pairs (Section 3). This is a challenging problem. One challenge is demonstrated in Figure 1, which contains two images that contain the noun "car" in their human-written captions. In one case (the top image), there actually *is* a car in the image; in the other case, there is not: the car refers to the state of the speaker.

The ability to automatically identify visual text is practically useful in a number of scenarios. One can imagine automatically mining image/caption data (like that in Figure 1) to train object recognition systems. However, in order to do so reliably, one must know whether the "car" actually appears or not. When building image search engines, it is common to use text near an image as features; this is more useful when this text is actually visual. Or when training systems to automatically generate captions of images (e.g., for visually impaired users), we need good language models for visual text.


Another dream **car** to add to the list, this one spotted in Hanbury St.

Shot out my **car** window while stuck in traffic because people in Cincinnati can't drive in the rain.

Figure 1: Two image/caption pairs, both containing the noun "car" but only the top one in a visual context.

One of our goals is to define what it means for a bit of text to be visual. As inspiration, we consider image/description pairs automatically crawled from Flickr (Ordonez et al., 2011). A first pass attempt might be to say "a phrase in the description of an image is *visual* if you can see it in the corresponding image." Unfortunately, this is too vague to be useful; the biggest issues are discussed in Section 2.2.

2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 762–772,
Montréal, Canada, June 3-8, 2012. ©2012 Association for Computational Linguistics

Based on our analysis, we settled on the following **definition:** A piece of text is visual (with respect to a corresponding image) *if* you can cut out a part of that image, paste it into any other image, and a third party could describe that cut-out part in the same way. In the car example, the claim is that I could cut out the car, put it in the middle of any other image, and someone else might still refer to that car as "dream car." The car in the bottom image in Figure 1 is *not* visual because there's nothing you could cut out that would retain car-ness.

2 Data Analysis

Before embarking on the road to building models of visual text, it is useful to obtain a better understanding of what visual text is like, and how it compares to the more standard corpora that we are used to working with. We describe the two large data sets that we use (one visual, one non-visual), then describe the quantitative differences between them, and finally discuss our annotation effort for labeling visual text.

2.1 Data sets

We use the SBU Captioned Photo Dataset (Ordonez et al., 2011) as our primary source of image/caption data. This dataset contains 1 million images with user associated captions, collected *in the wild* by intelligent filtering of a huge number of Flickr photos. Past work has made use of this dataset to retrieve whole captions for association with a query image (Ordonez et al., 2011). Their method first used global image descriptors to retrieve an initial matched set, and then applied more local estimates of content to re-rank this (relatively small) set (Ordonez et al., 2011). This means that content based matching was relatively constrained by the bottleneck of global descriptors, and local content (e.g., objects) had relatively small effect on accuracy.

As an auxiliary source of information for (largely) non-visual text, we consider a large corpus of text obtained by concatenating ukWaC[1] and the New York Times Newswire Service (NYT) section of the Gigaword (Graff, 2003) Corpus. The Web-derived ukWaC is already tokenized and POS-tagged with the TreeTagger (Schmid, 1995). NYT is tokenized, and POS-tagged using TagChunk (Daumé III and Marcu, 2005). This consists of 171 million sentences (4 billion words). We refer to this *generic* text corpus as **Large-Data**.

2.2 Formalizing *visual text*

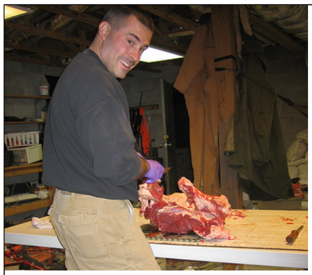
We begin our analysis by revisiting the definition of visual text from the introduction, and justifying this particular definition. In order to arrive at a sufficiently specific definition of "visual text," we focused on the applications of visual text that we care about. As discussed in the introduction, these are: training object detectors, building image search engines and automatically generating captions for images. Our definition is based on access to image/text pairs, but later we discuss how to talk about it purely based on text. To make things concrete, consider an image/text pair like that in the top of Figure 1. And then consider a phrase in the text, like "dream car." The question is: is "dream car" visual or not?

One of the challenges in arriving at such a definition is that the description of an image in Flickr is almost always written by the photographer of that image. This means the descriptions often contain information that is *not* actually pictured in the image, or contain references that are only relevant to the photographer (referring to a person/pet by name).

One might think that this is an artifact of this particular dataset, but it appears to be generic to all captions, even those written by a viewer (rather than the photographer). Figure 2 shows an image from the Pascal dataset (Everingham et al., 2010), together with captions written by random people collected via crowd-sourcing (Rashtchian et al., 2010). There is much in this caption that is clearly made-up by the author, presumably to make the caption more interesting (e.g., meta-references like "the camera" or "A photo" as well as "guesses" about the image, such as "garage" and "venison").

Second, there is a question of how much inference you are allowed to do when you say that you "see" something. For example, in the top image in Figure 1, the street *is* pictured, but does that mean that "Hanbury St." is visual? What if there were a street sign that clearly read "Hanbury St." in the image? This problem comes up all the time, when people say things like "in London" or "in France" in their captions. If it's just a portrait of people "in France,"

[1] ukWaC is a freely available Wikipedia-derived corpus from 2009; see http://wacky.sslmit.unibo.it/doku.php.



1. A distorted photo of **a man** cutting up **a large cut of meat** in a garage.
2. **A man** smiling at the camera while carving up **meat**.
3. **A man** smiling while **he** cuts up **a piece of meat**.
4. **A smiling man** is standing next to **a table** dressing a piece of venison.
5. **The man** is smiling into the camera as **he** cuts **meat**.

Figure 2: An image from the Pascal data with five captions collected via crowd-sourcing. Measurements on the SMALL and LARGE dataset show that approximately 70% of noun phrases are visual (bolded), while the rest are non-visual (underlined). See Section 2.4 for details.

it's hard to say that this is visual. If you see the Eiffel tower in the background, this is perhaps better (though it could be Las Vegas!), but how does this compare to a photo taken out of an airplane window in which you actually do see France-the-country?

This problem becomes even *more* challenging when you consider things *other than nouns*. For instance, when is a *verb* visual? For instance, the most common non-copula verb in our data is "sitting," which appears in roughly two usages: (1) "Took this shot, sitting in a bar and enjoying a Portugese beer." and (2) "Lexy sitting in a basket on top of her cat tree." The first one is clearly not visual; the second probably is. A more nuanced case is for "playing," as in: "Girls playing in a boat on the river bank" (probably visual) versus "Tuckered out from playing in Nannie's yard." The corresponding image for the latter description shows a sleeping cat.

Our final definition, based on cutting out the potentially visual part of the image, allows us to say that: (1) "venison" is not visual (because you cannot actually tell); (2) "Hanbury St." and "Lexy" are not visual (you can infer them, in the first case because there is only one street and in the second case because there is only one cat); (3) that seeing the *real* Eiffel tower in the background does not mean that "France" is visual (but again, may be inferred); etc.

2.3 Most Pronounced Differences

To get an intuitive sense of how Flickr captions (expected to be predominantly visual) and generic text (expected not to be so) differ, we computed some simple statistics on sentences from these. In general, the generic text had twice as many main verbs as the Flickr data, four times as many auxiliaries or light verbs, and about 50% more prepositions.

Flickr captions tended to have far more references to *physical* objects (versus *abstract* objects) than the generic text, according to the WordNet hierarchy. Approximately 64% of the objects in Flickr were physical (about 22% abstract and 14% unknown). Whereas in the generic text, only 30% of the objects were physical, 53% were abstract (17% unknown).

A third major difference between the corpora is in terms of noun modifiers. In both corpora, nouns tend not to have any modifiers, but modifiers are still more prevalent in Flickr than in generic text. In particular, 60% of nouns in Flickr have zero modifiers, but 70% of nouns in generic text have zero modifiers. In Flickr, 30% of nouns have exactly one modifier, as compared to only 22% for generic text.

The breakdown of what those modifiers look like is even more pronounced, even when restricted just to physical objects (modifier types are obtained through the bootstrapping process discussed in Section 3.1). Almost 50% of nominal modifiers in the Flickr data are color modifiers, whereas color accounts for less than 5% of nominal modifiers in generic text. In Flickr, 10% of modifiers talk about beauty, in comparison to less than 5% in generic text. On the other hand, less than 3% of modifiers in Flickr reference ethnicity, as compared to almost 20% in generic text; and 20% of Flickr modifiers reference size, versus 50% in generic text.

2.4 Annotating Visual Text

In order to obtain ground truth data, we rely on crowdsourcing (via Amazon's Mechanical Turk). Each *instance* is an image, a paired caption, and a highlighted noun phrase in that caption. The annotation for this instance is a label of "visual," "non-visual" or "error," where the error category is reserved for cases where the noun phrase segmentation was erroneous. Each worker is given five instances to label and paid one cent per annotation.[2]

For a small amount of data (803 images containing 2339 instances), we obtained annotations from three separate workers per instance to obtain higher quality data. For a large amount of data (48k images), we obtained annotations from only a single worker. Subsequently, we will refer to these two data sets as the SMALL and LARGE data sets. In both data sets, approximately 70% of the noun phrases were visual, 28% were non-visual and 2% were erroneous. For simplicity, we group erroneous and non-visual for all learning and evaluation.

In the SMALL data set, the rate of disagreement between annotators was relatively low. In 74% of the annotations, there was *no* disagreement at all. We reconciled the annotations using the quality management technique of Ipeirotis et al. (2010); only 14% of the annotations need to be changed in order to obtain a gold standard.

One immediate question raised in this process is whether one needs to actually see the image to perform the annotation. In particular, if we expect an NLP system to be able to classify noun phrases as visual or non-visual, we need to know whether people can do this task *sans* image. We therefore performed the *same* annotation on the SMALL data set, but where the workers were *not* shown the image. Their task was to *imagine* an image for this caption and then annotate the noun phrase based on whether they thought it would be pictured or not. We obtained three annotations as before and reconciled them (Ipeirotis et al., 2010). The accuracy of this reconciled version against the gold standard (produced by people who *did* see the image) was 91%. This suggests that while people are able to do this task with some reliability, seeing the image is very important (recall that always guessing "visual" leads to an accuracy of 70%).

3 Visual Features from Raw Text

Our first goal is to attempt to obtain relatively large knowledge bases of terms that are (predominantly) visual. This is potentially useful in its own right (for instance, in the context of search, to determine which query terms are likely to be pictured). We have explored two techniques for performing this task, the first based on bootstrapping (Section 3.1) and the second based on label propagation (Section 3.2). We then use these lists to generate features for a classifier that predicts whether a noun phrase—in context—is visual or not (Section 4).

In addition, we consider the task of separating adjectives into different visual categories (Section 3.3). We have already used the results of this in Section 2.3 to understand the differences between our two corpora. It is also potentially useful for the purpose of building new object detection systems or even attribute detection systems, to get a vocabulary of target detections.

3.1 Bootstrapping for Visual Text

In this section, we learn visual and non-visual nouns and adjectives automatically based on bootstrapping techniques. First, we construct a graph between adjectives by computing distributional similarity (Turney and Pantel, 2010) 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 exact, we use only those adjectives as modifiers which appear adjacent to a noun (that is, in a JJ NN construction). For example, in "small red apple," we consider only *red* as a modifier for noun. We use Pointwise Mutual Information (PMI) (Church and Hanks, 1989) to weight the contexts, and select the top 1000 PMI contexts for each adjective.[3]

Next, we apply cosine similarity to find the top 10 distributionally similar adjectives with respect to each target adjective based on our large generic corpus (**Large-Data** from Section 2.1). This creates a graph with adjectives as nodes and cosine similarity as weight on the edges. Analogously, we construct a graph with nouns as nodes (here, adjectives are used as contexts for nouns).

We then apply bootstrapping (Kozareva et al., 2008) on the noun and adjective graphs by selecting 10 seeds for visual and non-visual nouns and adjectives (see Table 1). We use in-degree (sum of weights of incoming edges) to compute the score for

[2] Data available at http://hal3.name/dvt/, with direct links back to the SBU Captioned Photo Dataset.

[3] We are interested in *descriptive* adjectives, which "typically ascribe to a noun a value of an attribute" (Miller, 1998).

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

Table 1: Example seeds for bootstrapping.

each node that has connections with known (seeds) or automatically labeled nodes, previously exploited to learn hyponymy relations from the web (Kozareva et al., 2008). Intuitively, in-degree captures the popularity of new instances among instances that have already been identified as good instances. We learn visual and non-visual words together (known as the mutual exclusion principle in bootstrapping (Thelen and Riloff, 2002; McIntosh and Curran, 2008)): each word (node) is assigned to only one class. Moreover, after each iteration, we harmonically decrease the weight of the in-degree associated with instances learned in later iterations. We added 25 new instances at each iteration and ran 500 iterations of bootstrapping, yielding 11955 visual and 11978 non-visual nouns, and 7746 visual and 7464 non-visual adjectives.

Based on manual inspection, the learned visual and non-visual lists look great. In the future, we would 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 the Flickr data-set: Visual nouns: 53.71%; Non-visual nouns: 14.25%; Visual adjectives: 51.79%; Non-visual adjectives: 14.40%. Overall, we find more visual nouns and adjectives are covered in the Flickr data-set, which makes sense, since the Flickr data-set is largely visual.

Second, we show the coverage of these classes on the large text corpora (**Large-Data** from Section 2.1): Visual nouns: 26.05%; Non-visual nouns: 41.16%; Visual adjectives: 20.02%; Non-visual adjectives: 40.00%. Overall, more non-visual nouns and adjectives cover text data, since **Large-Data** is a non-visual data-set.

Visual: attend, buy, clean, comb, cook, drink, eat, fry, pack, paint, photograph, smash, spill, steal, taste, tie, touch, watch, wear, wipe
Non-visual: achieve, admire, admit, advocate, alleviate, appreciate, arrange, criticize, eradicate, induce, investigate, minimize, overcome, promote, protest, relieve, resolve, review, support, tolerate

Table 2: Predicates that are visual and non-visual.

Visual: water, cotton, food, pumpkin, chicken, ring, hair, mouth, meeting, kind, filter, game, oil, show, tear, online, face, class, car
Non-visual: problem, poverty, pain, issue, use, symptom, goal, effect, thought, government, share, stress, work, risk, impact, concern, obstacle, change, disease, dispute

Table 3: Learned visual/non-visual nouns.

3.2 Label Propagation for Visual Text

To propagate visual labels, we construct a bipartite graph between visually descriptive predicates and their arguments. Let V_P be the set of nodes that corresponds to predicates, and let V_A be the set of nodes that corresponds to arguments. To learn the visually descriptive words, we set V_P to 20 visually descriptive predicates shown in the top of Table 2, and V_A to all nouns that appear in the object argument position with respect to the seed predicates. We approximate this by taking nouns on the right hand side of the predicates within a window of 4 words using the Web 1T Google N-gram data (Brants and Franz., 2006). For edge weights, we use conditional probabilities between predicates and arguments so that $w(p \rightarrow a) := pr(a|p)$ and $w(a \rightarrow p) := pr(p|a)$.

In order to collectively induce the visually descriptive words from this graph, we apply the graph propagation algorithm of Velikovich et al. (2010), a variant of label propagation algorithms (Zhu and Ghahramani, 2002) that has been shown to be effective for inducing a web-scale polarity lexicon based on word co-occurrence statistics. This algo-

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 4: Attribute Classes with their seed values

rithm iteratively updates the semantic distance between each pair of nodes in the graph, then produces a score for each node that represents how visually descriptive each word is. To learn the words that are *not* visually descriptive, we use the predicates shown in the bottom of Table 2 as V_P instead. Table 3 shows the top ranked nouns that are visually descriptive and *not* visually descriptive.

3.3 Bootstrapping Visual Adjectives

Our goal in this section is to automatically generate comprehensive lists of adjectives for different attributes, such as color, material, shape, etc. To our knowledge, this is the first significant effort of this type for adjectives: most bootstrapping techniques focus exclusively on nouns, although Almuhareb and Poesio (2005) populated lists of attributes using web-based similarity measures. We found that in some ways adjectives are easier than nouns, but require slightly different representations.

One might conjecture that listing attributes by hand is difficult. Colors names are well known to be quite varied. For instance, our bootstrapping approach is able to discover colors like "grayish," "chestnut," "emerald," and "rufous" that would be hard to list manually (the last is a reddish-brown color, somewhat like rust). Although perhaps not easy to create, the Wikipedia list of colors (http://en.wikipedia.org/wiki/List_of_colors) includes all of these except "grayish". On the other hand, it includes color terms that might be difficult to make *use of* as colors, such as "bisque," "bone" and "bubbles" (the last is a very light cyan), which might over-generate hits. For shape, we find "oblong," "hemispherical," "quadrangular" and, our favorite, "convex".

We use essentially the same bootstrapping process as described earlier in Section 3.1, but on a slightly different data representation. The only difference is that instead of linking adjectives to their 10 most similar neighbors, we link them only to 25 neighbors to attempt to improve recall.

We begin with seeds for each attribute class from Table 4. 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 an attribute class or not. The annotators assigned "1" if a learned value belongs to a class, otherwise "0".

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. Therefore, 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 obtain the correct output of several systems from the total output produced by these systems.

First, we measured the agreement on whether each learned value belongs to a semantic class or not. We computed κ to measure inter-annotator agreement for each pair of annotators. We focus our evaluation on 4 classes: age, beauty, color, and direction; between Human 2 and Human 3 and between Human 1 and Human 3, the κ value was 0.48; between Human 1 and Human 2 it was 0.45. These numbers are somewhat lower than we would like, but not terrible. If we evaluate the classes individually, we find that age has the lowest κ. If we remove "age," the pairwise κs rise to 0.59, 0.57 and 0.55.

Second, we compute Precision (Pr), Recall (Rec) and F-measure (F1) for different bootstrapping systems (based on the number of iterations and the number of new words added in each iteration). Two parameter settings performed consistently better than others (10 iterations with 25 items, and 5 iterations with 50 items). The former system achieves a precision/recall/F1 of 0.53, 0.71, 0.60 against Human 2; the latter achieves scores of 0.54, 0.72, 0.62.

4 Recognizing Visual Text

We train a logistic regression (aka maximum entropy) model (Daumé III, 2004) to classify text as visual or non-visual. The features we use fall into the following categories: WORDS (the actual lexical items and stems); BIGRAMS (lexical bigrams); SPELL (lexical features such as capitalization pattern, and word prefixes and suffixes); WORDNET (set of hypernyms according to WordNet); and BOOTSTRAP (features derived from bootstrapping or label propagation).

For each of these feature categories, we compute features *inside* the phrase being categorized (e.g., "the car"), *before* the phrase (two words to the left) and *after* the phrase (two words to the right). We additionally add a feature that computes the number of words in a phrase, and a feature that computes the position of the phrase in the caption (first fifth through last fifth of the description). This leads to seventeen feature templates that are computed for each example. In the SMALL data set, there are $25k$ features ($10k$ non-singletons); in the LARGE data set, there are $191k$ features ($79k$ non-singletons).

To train models on the SMALL data set, we use 1500 instances as training, 200 as development and the remaining 639 as test data. To train models on the LARGE data set, we use 45000 instances as training and the remaining 4401 as development. We always test on the 639 instances from the SMALL data, since it has been redundantly annotated. The development data is used only to choose the regularization parameter for a Gaussian prior on the logistic regression model; this parameter is chosen in the range $\{0.01, 0.05, 0.1, 0.5, 1, 2, 4, 8, 16, 32, 64\}$.

Because of the imbalanced data problem, evaluating according to accuracy is not appropriate for this task. Even evaluating by precision/recall is not appropriate, because a baseline system that guesses that everything is visual obtains 100% recall and 70% precision. Due to these issues, we instead evaluate according to the area under the ROC curve (AUC). To check statistical significance, we compute standard deviations using bootstrap resampling, and consider there to be a significant difference if a result falls outside of two standard deviations of the baseline (95% confidence).

Figure 3 shows learning curves for the two data sets. The SMALL data achieves an AUC score of 71.3 in the full data setting (1700 examples); the LARGE data needs $12k$ examples to achieve similar accuracy due to noise. However, with $49k$ examples, we are able to achieve a AUC score of 75.3 using the large data set. By pooling the data (and weighting the small data), this boosts results to 76.1. The confidence range on these data is approximately ±1.9, meaning that this boost is likely not significant.

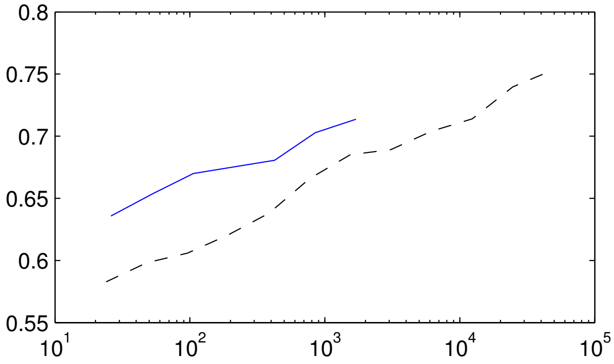


Figure 3: Learning curves for training on SMALL data (blue solid) and LARGE data (black dashed). X-axis (in log-scale) is number of training examples; Y-axis is AUC.

4.1 Using Image Features

As discussed previously, humans are only able to achieve 90% accuracy on the visual/non-visual task when they are not allowed to view the image. This potentially upper-bounds the performance of a learned system that can only look at text. In order to attempt to overcome this, we augment our basic system with a number of features computed from the corresponding images. These features are derived from the output of state of the art vision algorithms to detect 121 different objects, stuff and scenes.

As our object detectors, we use standard state of the art deformable part-based models (Felzenszwalb et al., 2010) for 89 common object categories, including: the original 20 objects from Pascal, 49 objects from Object Bank (Li-Jia Li and Fei-Fei, 2010), and 20 from Im2Text (Ordonez et al., 2011). We additionally use coarse image parsing to estimate background elements in each database image. Six possible background (stuff) categories are considered: sky, water, grass, road, tree, and building. For this we use detectors (Ordonez et al., 2011) which compute color, texton, HoG (Dalal and Triggs, 2005) and Geometric Context (Hoiem et al., 2005) as input features to a sliding window based SVM classifier. These detectors are run on all database images, creating a large pool of background elements for retrieval. Finally, we ob-

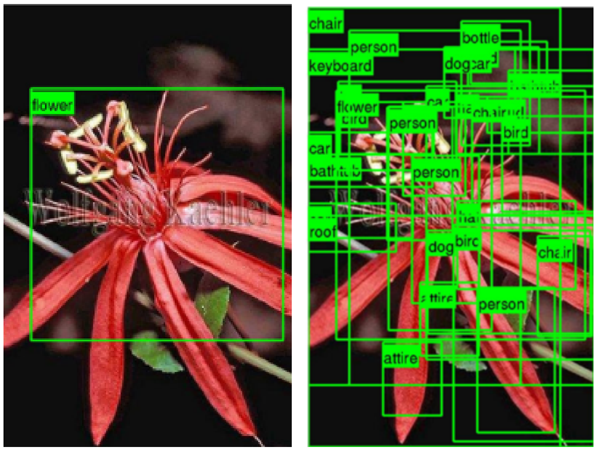


Figure 4: (Left) Highest confidence flower detected in an image; (Right) All detections in the same image.

	CATEGORY	POSITION	AUC
	Bootstrap	Phrase	65.2
+	Spell	Phrase	68.6
+	Image	-	69.2
+	Words	Phrase	70.0
+	Length	-	69.8
+	Wordnet	Phrase	70.4
+	Wordnet	Before	*70.6*
+	Spell	Before	*71.8*
+	Words	Before	*72.2*
+	Bootstrap	Before	**72.4**
+	Spell	After	*71.5*

Table 5: Results of feature ablation on SMALL data set. Best result is in bold; results that are not statistically significantly worse are italicized.

tain scene descriptors for each image by computing scene classification scores for 26 common scene categories, using the features, methods and training data from the SUN dataset (Xiao et al., 2010).

Figure 4 shows an example image on which several detectors have been run. From each image, we extract the following features: which object detectors fired; how many times they fired; the confidence of the most-likely firing; the percentage of the image (in pixels) that the bounding box corresponding to this object occupies; and the percentage of the width (and height) of the image that it occupies.

Unfortunately, object detection is a highly noisy process. The right image in Figure 4 shows all detections for that image, which includes, for instance, a chair detection that spans nearly the entire image, and a person detection in the bottom-right corner. For an average image, if a single detector (e.g., the flower detector) fires once, it actually fires 40 times ($\pm\sigma = 1.8$). Moreover, of the 120 detectors, on an average image over 22 ($\pm\sigma = 5.6$) of them fire at least once (though certainly in an average image only a few objects are actually present). Exacerbating this problem, although the confidence scores for a single detector can be compared, the scores between different detectors are not at all comparable. In order to attenuate this problem, we include duplicate copies of all the above features restricted to the *most confident* object for each object type.

On the SMALL data set, this adds 400 new features (300 of which are non-singletons[4]); on the LARGE data set, this adds 500 new features (480 non-singletons). Overall, the AUC scores trained on the small data set increase from 71.3 to 73.9 (a significant improvement). On the large data set, the increase is only from 76.1 to 76.8, which is not likely to be significant. In general, the improvement obtained by adding image features is most pronounced in the setting of small training data, perhaps because these features are more generic than the highly lexicalized features used in the textual model. But once there is a substantial amount of text data, the noisy image features become less useful.

4.2 Feature Ablations

In order to ascertain the degree to which each feature template is useful, we perform an ablation study. We first perform feature selection at the template level using the information gain criteria, and then train models using the corresponding subset of features.

The results on the SMALL data set are shown in Table 5. Here, the bootstrapping features computed on words within the phrase to be classified were judged as the most useful, followed by spelling features. Image features were judged third most useful. In general, features in the phrase were most useful (not surprisingly), and then features before the phrase (presumably to give context, for instance as in "*out of* the window"). Features from after the phrase were not useful.

[4] Non-singleton features appear more than once in the data.

	Category	Position	AUC
	Words	Phrase	74.7
+	Image	-	74.4
+	Bootstrap	Phrase	74.3
+	Spell	Phrase	75.3
+	Length	-	74.7
+	Words	Before	*76.2*
+	Wordnet	Phrase	*76.1*
+	Spell	After	*76.0*
+	Spell	Before	*76.8*
+	Wordnet	Before	**77.0**
+	Wordnet	After	75.6

Table 6: Results of feature ablation on Large data set.

Corresponding results on the Large data set are shown in Table 6. Note that the order of features selected is different because the training data is different. Here, the most useful features are simply the words in the phrase to be classified, which alone already gives an AUC score of 74.7, only a few points off from the best performance of 77.0 once image features, bootstrap features and spelling features are added. As before, these features are rated as very useful for classification performance.

Finally, we consider the effect of using Bootstrap-based features or label-propagation-based features. In all the above experiments, the features used are based on the *union* of word lists created by these two techniques. We perform three experiments. Beginning with the system that contains *all* features (Small=73.9, Large=76.8), we first remove the bootstrap-based features (Small→71.8, Large→75.5) or remove the label-propagation-based features (Small→71.2, Large→74.9) or remove both (Small→70.7, Large→74.2). From these results, we can see that these techniques are useful, but somewhat redundant: if you had to choose one, you should choose label-propagation.

5 Discussion

As connections between language and vision become stronger, for instance in the contexts of object detection (Hou and Zhang, 2007; Kim and Torralba, 2009; Sivic et al., 2008; Alexe et al., 2010; Gu et al., 2009), attribute detection (Ferrari and Zisserman, 2007; Farhadi et al., 2009; Kumar et al., 2009; Berg et al., 2010), visual phrases (Farhadi and Sadeghi, 2011), and automatic caption generation (Farhadi et al., 2010; Feng and Lapata, 2010; Ordonez et al., 2011; Kulkarni et al., 2011; Yang et al., 2011; Li et al., 2011; Mitchell et al., 2012), it becomes increasingly important to understand, and to be able to detect, text that *actually* refers to observed phenomena. Our results suggest that while this is a hard problem, it is possible to leverage large text resources and state-of-the-art computer vision algorithms to address it with high accuracy.

Acknowledgments

T.L. Berg and K. Yamaguchi were supported in part by NSF Faculty Early Career Development (CAREER) Award #1054133; A.C. Berg and Y. Choi were partially supported by the Stony Brook University Office of the Vice President for Research; H. Daumé III and A. Goyal were partially supported by NSF Award IIS-1139909; all authors were partially supported by a 2011 JHU Summer Workshop.

References

B. Alexe, T. Deselaers, and V. Ferrari. 2010. What is an object? In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 73 –80.

A. Almuhareb and M. Poesio. 2005. Finding concept attributes in the web. In *Corpus Linguistics Conference*.

Tamara L. Berg, Alexander C. Berg, and Jonathan Shih. 2010. Automatic attribute discovery and characterization from noisy web data. In *European Conference on Computer Vision (ECCV)*.

Thorsten Brants and Alex Franz. 2006. Web 1t 5-gram version 1. In *Linguistic Data Consortium, ISBN: 1-58563-397-6, Philadelphia.*

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

N. Dalal and B. Triggs. 2005. Histograms of oriented gradients for human detection. In *CVPR*.

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

Hal Daumé III. 2004. Notes on CG and LM-BFGS optimization of logistic regression. Paper available at http://pub.hal3.name/#daume04cg-bfgs, implementation available at http://hal3.name/megam/, August.

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

Ali Farhadi and Amin Sadeghi. 2011. Recognition using visual phrases. In *Computer Vision and Pattern Recognition (CVPR)*.

A. Farhadi, I. Endres, D. Hoiem, and D.A. Forsyth. 2009. Describing objects by their attributes. In *Computer Vision and Pattern Recognition (CVPR)*.

A. Farhadi, M. Hejrati, M.A. Sadeghi, P. Young, C. Rashtchian1, J. Hockenmaier, and D.A. Forsyth. 2010. Every picture tells a story: Generating sentences from images. In *ECCV*.

P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. 2010. Discriminatively trained deformable part models, release 4. http://people.cs.uchicago.edu/~pff/latent-release4/.

Y. Feng and M. Lapata. 2010. How many words is a picture worth? automatic caption generation for news images. In *ACL*.

V. Ferrari and A. Zisserman. 2007. Learning visual attributes. In *Advances in Neural Information Processing Systems (NIPS)*.

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

Chunhui Gu, J.J. Lim, P. Arbelaez, and J. Malik. 2009. Recognition using regions. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1030 –1037.

Derek Hoiem, Alexei A. Efros, and Martial Hebert. 2005. Geometric context from a single image. In *ICCV*.

Xiaodi Hou and Liqing Zhang. 2007. Saliency detection: A spectral residual approach. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1 –8.

P. Ipeirotis, F. Provost, and J. Wang. 2010. Quality management on amazon mechanical turk. In *Proceedings of the Second Human Computation Workshop (KDD-HCOMP)*.

Gunhee Kim and Antonio Torralba. 2009. Unsupervised Detection of Regions of Interest using Iterative Link Analysis. In *Annual Conference on Neural Information Processing Systems (NIPS 2009)*.

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

G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C Berg, and T. L Berg. 2011. Babytalk: Understanding and generating simple image descriptions. In *CVPR*.

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

Siming Li, Girish Kulkarni, Tamara L. Berg, Alexander C. Berg, and Yejin Choi. 2011. Composing simple image descriptions using web-scale n-grams. In *CONLL*.

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

Tara McIntosh and James R Curran. 2008. 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.

K.J. Miller. 1998. Modifiers in WordNet. In C. Fellbaum, editor, *WordNet*, chapter 2. MIT Press.

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

Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. 2011. Im2Text: Describing Images Using 1 Million Captioned Photographs. In *NIPS*.

Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. 2010. Collecting image annotations using amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*. Association for Computational Linguistics.

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

J. Sivic, B.C. Russell, A. Zisserman, W.T. Freeman, and A.A. Efros. 2008. Unsupervised discovery of visual object class hierarchies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –8.

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

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research (JAIR)*, 37:141.

Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. 2010. 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. Association for Computational Linguistics.

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

Yezhou Yang, Ching Lik Teo, Hal Daumé III, and Yiannis Aloimonos. 2011. Corpus-guided sentence generation of natural images. In *EMNLP*.

Xiaojin Zhu and Zoubin Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation. In *Technical Report CMU-CALD-02-107*. CarnegieMellon University.