Using Visual Information to Predict Lexical Preference

Shane Bergsma
Dept. of Computer Science and HLTCOE
Johns Hopkins University
sbergsma@jhu.edu

Randy Goebel
Dept. of Computing Science
University of Alberta
goebel@cs.ualberta.ca

Abstract

Most NLP systems make predictions based solely on linguistic (textual or spoken) input. We show how to use *visual* information to make better *linguistic* predictions. We focus on selectional preference; specifically, determining the plausible noun arguments for particular verb predicates. For each argument noun, we extract visual features from corresponding images on the web. For each verb predicate, we train a classifier to select the visual features that are indicative of its preferred arguments. We show that for certain verbs, using visual information can significantly improve performance over a baseline. For the successful cases, visual information is useful even in the presence of co-occurrence information derived from web-scale text. We assess a variety of training configurations, which vary over classes of visual features, methods of image acquisition, and numbers of images.

1 Introduction

Selectional preferences quantify the plausibility of predicate-argument pairs. We focus on predicting the plausibility of a noun argument (e.g. *pasta*) occurring as the direct object of a verb predicate (e.g. *eat*). Such knowledge is useful since many NLP tasks require determining the actual argument from the alternatives that arise because of syntactic, semantic or anaphoric ambiguity. Previous uses of selectional preferences include prepositional-phrase attachment (Hindle and Rooth, 1993), word-sense disambiguation (Resnik, 1997), pronoun resolution (Dagan and Itai, 1990), and semantic role labeling (Erk, 2007).

The compatibility of a predicate and an argument can be quantified by counting how often they occur together in a large text corpus (Hindle and Rooth, 1993), but many plausible pairs are absent even from web-scale text (Bergsma et al., 2008). We therefore seek to *generalize* from observed pairs in order to make inferences for unseen combinations. Some approaches back off to counts over argument classes (Resnik, 1996; Rooth et al., 1999; Clark and Weir, 2002; Ó Séaghdha, 2010; Ritter et al., 2010), Others interpolate over similar words (Dagan et al., 1999; Erk, 2007). Text-based approaches work best for arguments that are *frequent* in text, but, paradoxically, frequent arguments are the arguments for which generalization is least needed. This provides motivation to look beyond text in order to make better predictions for infrequent or out-of-vocabulary arguments.

We propose using *visual* features to identify a verb's preferred arguments. Visual information may play a role in the human acquisition of word meaning (Feng and Lapata, 2010b). For computers, there is a massive amount of visual data to exploit. Billions of images are added to websites like Facebook and Flickr every month. The challenge of associating words and images is reduced because many users label their images as they post them online, providing an explicit link between a word and its visual depiction. Bergsma and Van Durme (2011) used these explicit word-image connections in order to find words in different languages having the same meaning (translations); pairs of words are proposed as translations if their visual depictions are visually similar.

In this paper, we use online images to help predict a predicate's selectional preferences. For each verb-noun pair, (v, n), we retrieve labeled images of n from the web, and apply computer vision techniques to extract visual features from the images. We then use the DSP model of Bergsma et al. (2008) to combine the visual features collected for n into a single plausibility score for (v, n). In the original DSP model, each verb has a corre-

Proceedings of Recent Advances in Natural Language Processing, pages 399–405,
Hissar, Bulgaria, 12-14 September 2011.

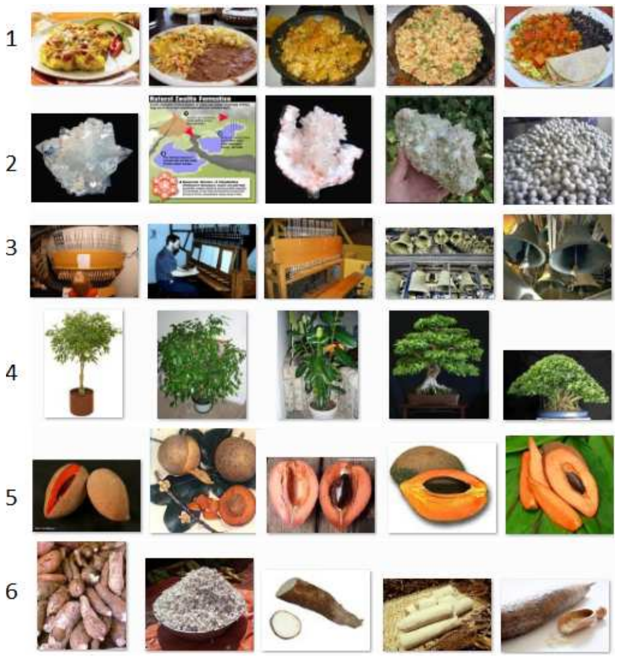

Figure 1: Which out-of-vocabulary nouns are plausible direct objects for the verb *eat*? Each row corresponds to a noun: 1. *migas*, 2. *zeolite*, 3. *carillon*, 4. *ficus*, 5. *mamey* and 6. *manioc*.

sponding classifier that scores noun arguments on the basis of various *textual* features. We use this discriminative framework to incorporate the visual information as new, *visual* features.

Our experiments evaluate the ability of these classifiers to correctly predict the selectional preferences of a small set of verbs. We evaluate two cases: 1) the case where the nouns are all assumed to be out-of-vocabulary, and the classifiers must make predictions without any corpus-based co-occurrence information, and 2) the case where we assume access to noun-verb co-occurrence information derived from web-scale N-gram data.

We show that visual features are useful for some verbs, but not for others. For verbs taking abstract arguments without definitive visual features, the classifier can often learn to disregard the visual data. On the other hand, for verbs taking *physical* arguments (such as food, animals, or people), the classifier can make accurate predictions using the nouns' visual properties. In these cases, visual information remains useful even after incorporating the web-scale statistics.

2 Visual Selectional Preference

Consider determining whether the nouns *carillon*, *migas* and *mamey* are plausible arguments for the verb *eat*. Existing systems are unlikely to have such words in their training data, let alone information about their edibility. However, after inspecting a few images returned by a Google search for these words (Figure 1), a human might reasonably predict which words are edible. Humans make this determination by observing both intrinsic visual properties (pits, skins, rounded shapes and fruity colors) and extrinsic visual context (circular plates, bowls, and other food-related tools) (Oliva and Torralba, 2007).

We propose using similar information to predict the plausibility of arbitrary verb-noun pairs. That is, we aim to learn the distinguishing visual features of all nouns that are plausible arguments for a given verb. This differs from work that has aimed to recognize, annotate and retrieve objects defined by a single phrase, such as *tree* or *wrist watch* (Feng and Lapata, 2010a). These approaches learn from labeled images during training in order to assign words to unlabeled images during testing. In contrast, we analyze labeled images (during training and testing) in order to determine their visual compatibility with a given predicate. Our approach does not need labeled training images for a *specific* noun in order to assess that noun during testing; e.g. we can make a reasonable prediction for the plausibility of *eat mamey* even if we've never encountered *mamey* before.

We now specify how we automatically 1) download a set of images for each noun, 2) extract visual features from each image, and 3) combine the visual features from multiple images into plausibility scores. Scripts, code and data are available at: `www.clsp.jhu.edu/~sbergsma/ImageSP/`.

2.1 Mining noun images from the web

To obtain a set of images for a particular noun argument, we submit the noun as a query to either the Flickr photo-sharing website (`www.flickr.com`), or Google's image search (`www.google.com/imghp`). In both cases, we download the thumbnails on the results page directly rather than downloading the source images. Flickr returns images by matching the query against user-provided tags and accompanying text. Google retrieves images based on the image caption, file-name, and surrounding text (Feng and Lapata, 2010a). Images obtained from Google are known to be competitive with "hand prepared datasets" for training object recognizers (Fergus et al., 2005).

2.2 Extracting visual features from images

A range of features have been developed in the vision community, typically with the aim of improving content-based image retrieval (Deselaers et al., 2008). We follow previous work in using features in a *bag-of-words* representation that ignores the spacial relationship between image components.

Color Histogram Our first set of features are extracted from the color histogram of the image. We partition the color space by dividing the R, G, and B values of the pixel colors into equal-sized bins. For a given image, we count the number of pixels that occur within each RGB bin. Each color bin and its count is used as a feature dimension and its value, respectively. We describe how we choose the number of bins in Section 3.

SIFT Keypoints Additional features are derived from the image's SIFT (scale-invariant feature transform) keypoints (Lowe, 2004). SIFT keypoints are detected at visually-distinct image locations. Each keypoint has a corresponding *descriptor vector* that identifies a location's unique visual properties. SIFT keypoints are conceptually similar to local features identified by so-called *corner detectors*. Corner detectors find image locations that have "large gradients in all directions at a predetermined scale" (Lowe, 2004). Unlike typical corner detectors, SIFT keypoints are invariant to scaling and rotation. They are also robust to illumination, noise and distortion. We identify SIFT keypoints using David Lowe's software: `www.cs.ubc.ca/~lowe/keypoints/`. SIFT keypoints are taken from images converted to grayscale.

Since each keypoint is itself a vector, we quantize the keypoints by mapping them to a set of K discrete visual words. This set of words forms the visual vocabulary of our bag-of-words representation. The set of words is obtained by clustering a random selection of keypoints into K cluster centroids using the K-means algorithm. The final feature representation for an image consists of a feature dimension for each visual word; each feature value is the number of keypoints in the image that have that word as their nearest centroid.

We generate different clusterings (and thus different vocabularies) separately for each verb predicate. For each verb, we randomly sample 500,000 keypoints from the set of downloaded images for that verb's potential argument nouns, and run the clustering over these keypoints. Section 3 describes how we choose the number of clusters, K.

2.3 Combining features with the DSP model

We use DSP (Bergsma et al., 2008) to generate a plausibility score for a verb-noun pair, (v, n). Let Φ be a function that generates features for nouns, $\Phi : n \rightarrow (\phi_1 ... \phi_k)$. We explain below how, for each n, we aggregate visual features across multiple images to create features in $\Phi(n)$. DSP determines whether n is a plausible argument of v by scoring $\Phi(n)$ using a verb-specific set of learned weights, $\boldsymbol{w}_v$=$(w_1 ... w_k)$. The weights are trained for each v in order to distinguish the verb's positive nouns from its negatives in training data (the generation of training data is also explained below). The weights can be learned using any binary classification algorithm; we use logistic regression. At test time, we generate a final compatibility score (prediction) via the logistic function:

$$\text{Score}(v, n) = \frac{exp(\boldsymbol{w}_v \cdot \Phi(n))}{1 + exp(\boldsymbol{w}_v \cdot \Phi(n))} \qquad (1)$$

Our discriminative model differs from a recent generative model over words and visual features by Feng and Lapata (2010b). In that work, including visual features resulted in better topic clusters, which indirectly improved (topic-derived) word-word associations. In our work, visual features are directly exploited by a discriminative model, allowing us to use arbitrary and potentially interdependent visual attributes in our representation.

Generating Examples We follow Bergsma et al. (2008)'s approach by first calculating the pointwise mutual information (PMI) between predicate verbs and (direct object) argument nouns in a large parsed corpus. For each verb predicate, v, we create positive examples, (v, n), by pairing v with *all* nouns, n, such that v and n have a positive PMI, i.e. PMI$(v, n) > 0$. For each of these positives pairs (e.g. *eat pasta*), we generate two *pseudo*-negative examples, (v, n'), by randomly pairing v with some nouns n' that either did not occur with v (and hence PMI is undefined) or have PMI$(v, n') \leq 0$ (e.g., *eat distribution*, *eat wheelchair*). As in Bergsma et al. (2008), pseudo-negatives n' are chosen to have similar corpus frequency to the original positive noun, n.

We use this approach to generate both training examples for learning the DSP classifier and also separate test examples for evaluating the model's predictions. We train and evaluate a classifier for each v separately from all other verbs. For each v, we take 85% of examples for training, 7.5% for development, and 7.5% for final testing.

Generating Features The DSP model allows us to use any information that might indicate a noun's compatibility with a verb; we simply encode this information as features in the noun's feature representation, $\Phi(n)$. Bergsma et al. used DSP's flexibility to include novel string-based features of the noun argument (e.g., the verb *become* prefers lower-case direct objects; *accuse* prefers capitalized ones). We augment $\Phi(n)$ with visual features.

Since we download multiple images for each noun, n, we have multiple color histograms and multiple bags of SIFT keypoints. To generate a single feature representation, $\Phi(n)$, we first sum the color and SIFT-keypoint feature vectors, respectively, across all the images in n's image set. We then normalize each sum vector to unit length, and include all of the resulting normalized features as additional features in $\Phi(n)$.

In summary, we can produce a score for a (v, n) pair at test time as follows: 1) select the appropriate weights, $\boldsymbol{w}_v$, for verb v, 2) generate the composite (normalized) feature vector, $\Phi(n)$, for noun n, and 3) score the features with the weights using the formula for Score(v, n) (Equation (1) above). In practice, this score is exactly what is returned by our logistic regression software package. We can use this score directly, or, for hard classifications, predict positive if the returned probability is greater than 0.5 and otherwise predict negative.

3 Experimental Set-up

Task and Data The task is to predict whether a particular verb-noun pair, previously unseen during training of the DSP classifier, is a positive or a negative example, as defined in Section 2.3 above. We evaluate using *Accuracy*: the proportion of examples correctly classified on test data. We calculate significance using *McNemar's test.*

Since the negatives are pseudo-negatives, this kind of evaluation is also known as a pseudo-disambiguation evaluation. While the set-up of pseudo-disambiguation evaluations has varied in NLP (Chambers and Jurafsky, 2010), we use an identical set-up to Bergsma et al. (2008): we generate positive and negative examples for DSP from a parsed and processed copy of the AQUAINT corpus, and use the same PMI-threshold (i.e. 0) and positive-to-negative ratio (i.e. 1:2).

We evaluate on nouns in the direct object position of seven verbs: *eat*, *inform*, *hit*, *kill*, *park*, *hunt* and *shoot down*. The total number of training examples for these verbs varies from roughly 500 to 10,000 instances, while the number of test instances varies from roughly 50 to 1000 instances.

We chose these seven verbs as test cases because we speculated they might benefit from visual information to different degrees (e.g. we expected indicative food-features for *eat*, but perhaps less helpful human-features for *inform*, etc.). Ideally one would like to automatically categorize all the verbs for which visual features might be helpful, but it is natural to first demonstrate the benefits of visual information in certain cases in order to motivate further study. Importantly, note that while we hand-selected a set of verb predicates, our evaluation data is based on real observed arguments of these predicates, and in particular not on nouns for which we would *a priori* expect visual information to be predictive. Our evaluation is thus focused, but realistic.

Classifier In all cases, we use an L2-regularized logistic regression model for DSP's base classifier, and train it via LIBLINEAR (Fan et al., 2008). We optimize the regularization parameter on the development data.

Visual Features For each noun,[1] we take the first six images returned from both Google and Flickr, and extract the corresponding visual features as described above. While we later discovered that the more images we have, the better the results (Figure 2), we initially decided to use only six images mainly for computational reasons; downloading and processing images is space and time-intensive.

Rather than selecting fixed values for the size of the color bins and the number of SIFT centroids, we take advantage of our model's flexibility to use features over different granularities: we use separate features with both 64 and 512 color bins, and with both 100 and 1000 SIFT centroids. The flexibility to include visual information at different levels of granularity is one of the chief advantages of the discriminative model.

Test Configurations We are primarily interested in whether visual information can lead to

[1] For a given verb in our corpus, DSP actually provides plausibility scores for both nouns and multi-word noun phrases; we refer to both of these as 'nouns' for convenience.

System	*eat*	*inform*	*hit*	*kill*	*park*	*hunt*	*shoot down*	Average
Baseline	68.3	68.0	68.7	67.7	69.9	67.6	70.0	68.6
+ Visual Features via Flickr	75.8	68.0	**68.8**	67.2	69.9	69.6	70.0	69.9
+ Visual Features via Google	***79.5***	**68.2**	68.7	***68.5***	69.9	***76.5***	**72.0**	**71.9**

Table 1: Using visual features from Google significantly improves accuracy (%) over the baseline system on *eat* ($p<0.001$), *kill* ($p<0.1$) and *hunt* ($p<0.1$).

better predictions on out-of-vocabulary (OOV) nouns, but obtaining a sufficiently-large test set of labeled OOV instances is difficult. We therefore first provide results on *simulated* OOV arguments (Section 4.1), where we assume no corpus-based knowledge is available to the DSP classifier. That is, we initially exclude corpus-based features from our models. We compare visual models to ones that only use features for the noun string (such features are always available). Our string features are binary features that indicate the 'shape' of the noun via the regular expression maps: [A-Z]+ → A, and [a-z]+ → a. E.g., *Al Unser Jr.* will have the one feature '*Aa Aa Aa.*'.

In the second part of our results (Section 4.2), we test whether visual information can help even in the presence of high-quality corpus-based features. We use Keller and Lapata (2003)'s approach to obtain web-scale co-occurrence frequencies for the verb-noun pair. That is, we retrieve counts for the pattern "V Det N" from a web-scale Google N-gram corpus (Lin et al., 2010). Here, V is any inflection of the verb, Det is *the*, *a*, *an*, or the empty string, and N is the noun. We include the log-count of this pattern as a feature, and also include separate features for the log-counts of the noun and verb themselves. By multiplying these features by appropriate weights, a classifier can generate a (web-based) PMI score.

4 Results

4.1 Results on OOV nouns

We now compare the use of visual features to string-based features alone (Baseline), simulating out-of-vocabulary arguments by assuming no corpus-based knowledge is available for the noun features. For these verbs, we actually found the Baseline with only string features to be no better than picking the majority-class.

Visual features significantly improve performance for 3 of the verbs (Table 1). Visual features do not improve (but also do not impair) accuracy on the verbs that have mostly abstract or general arguments. For example, one can "*hit turbulence*," "*hit record*," or "*hit the slopes*," but there are no visual features that can help select these nouns. Macro-averaged accuracy across all verbs increases from a baseline of 68.6% to 71.9% using Google-derived visual features.

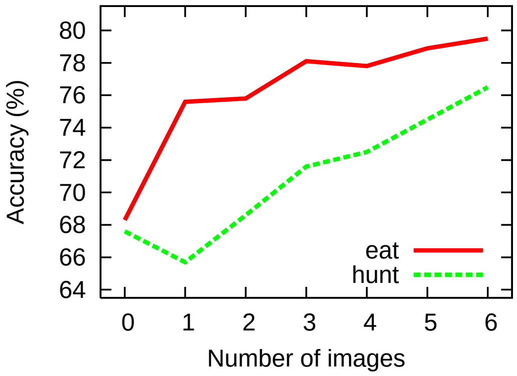

Figure 2: The more images, the more accurate: Performance on the verbs *eat* and *hunt* as features are extracted from a varying number of images.

The features obtained from Google images perform better than features from Flickr (Table 1). Inspecting the retrieved image sets, we observe that compared to Flickr, Google tends to retrieve more consistent, more canonical images for a particular noun. For example, Google's top results for the query "buffalo" are exclusively images of buffalo animals. On Flickr, "buffalo" returns images of the city of Buffalo, buffalo hides, and pictures of buffalo animals alongside people, cars, birds, etc. For our purposes, the consistency of the Google images is better; it makes learning and predicting easier for the visual classifier.

We provide further analysis using Google images only. Figure 2 shows that, as we use more images, accuracy on the verbs *eat* and *hunt* improves and is not yet leveling off. With computation only linear in the number of images, adding even more images is one possible way to improve accuracy.

Table 2 shows the contribution of the two visual feature types for classifying arguments involving

Features	Accuracy
All Features	79.5
-Color Histogram	78.4
-SIFT Keypoint	78.1
-Color & -SIFT	68.3

Table 2: Accuracy on *eat* as different feature classes are removed.

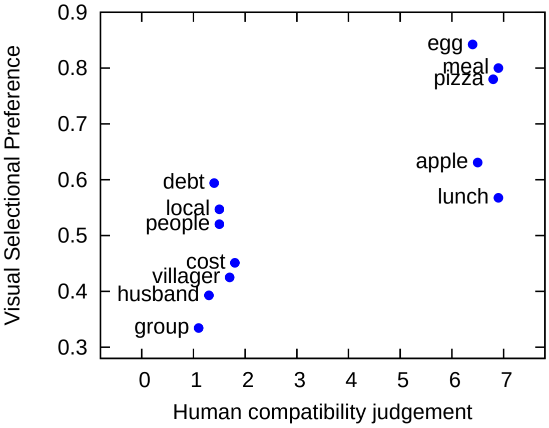

Figure 3: Visual selectional preference correlates well with human judgments: arguments of the verb *eat* are plotted using visual and average human compatibility scores.

the verb *eat*. Either visual feature type helps a lot on its own; together they further improve accuracy.

We also tried replacing our logistic classifier with kernelized SVMs, which have previously proved useful for object recognition (Chapelle et al., 1999). While kernel-SVMs can implicitly consider all combinations of features (resulting in the encoding of richer visual information), we found the resulting gains over linear classifiers to be minimal. The kernelized SVMs also took much longer to train and apply. The further development of effective while still efficient visual features remains an important direction for future work.

Figure 3 compares the scores of the visual system (computed via Equation (1)) to human plausibility judgments (described by Padó et al. (2006)).[2] The human scores are the average judgments for the question, “how common is it to *eat* X?” where X is a given noun. Participants responded with scores from 1 (very uncommon) to 7 (very common). These average judgments have a high correlation with our predicted scores; the Pearson correlation coefficient is 0.803. The visual system does a good job on the nouns *egg*, *meal*, *pizza* and *apple*, but ranks *debt* above (the somewhat abstract) *lunch*. Looking at the Google images for *lunch*, we note that clearer pictures of food occur beyond the top 6 images, and hence using more images would likely improve scoring.

System	*eat*	*kill*	*hunt*
Baseline	68.3	67.7	67.6
+ Visual Features alone	79.5	68.5	76.5
+ Web Co-occ alone	85.1	74.0	76.5
+ Web Co-occ & Visual	**85.7**	**74.3**	**78.4**

Table 3: Visual features improve accuracy (%) even when web co-occurrence information is used.

Finally, we note that for *eat*, we found the visual system’s accuracy was consistent across nouns of different frequencies. This contrasts with systems using text-based features; these perform much better on more frequent nouns (Bergsma et al., 2008).

4.2 Results with web-scale statistics

We have shown that visual information can result in significantly improved performance in cases where no corpus-based information is available. Do these gains hold up when high-quality corpus-based information is available?

On those verbs where visual information helped in the OOV setting, visual information remains helpful even with features encoding web-scale co-occurrence statistics (Table 3).[3] Note the gains from adding visual features are consistent in all three cases, but not statistically significant, as the proportion of nouns where the visual features can help is now much smaller.

These final results are somewhat sobering. Visual information is not helpful for every verb, and even in the positive cases, it is not very helpful when combined with existing text-based features. However, the exploitation of visual information is still in its infancy in NLP. Using search engines to obtain images for NLP today is perhaps similar to how search engines were also used to obtain web-scale *text* statistics for NLP a decade ago. While we leveraged a relatively small number of visual features from a relatively small number of images,

[2] Available online at `http://www.nlpado.de/~ulrike/data/pado_plausibility.tgz`

[3] Not surprisingly, on the verbs where visual features were not effective earlier, visual features remains ineffective here; these features tend to actually impair performance when added to the web-scale co-occurrence features.

future advances in computer vision and large-scale data processing will allow richer visual information to be extracted and applied to NLP problems.

5 Conclusion

We have shown that it is possible to predict verb-noun selectional preference purely on the basis of visual information. For a given noun, web images are downloaded, processed, and then analyzed by classifiers corresponding to different verbs. Each verb classifier is trained to identify the visual properties that distinguish the verb's preferred arguments. Statistically-significant improvements were obtained on three verbs and visual data remains helpful even in the presence of high-quality web-scale co-occurrence information.

These results give us a good basis for moving forward. We know where we should get our images (Google), which features are useful (both color and SIFT) and how many images to use (as many as possible). It remains to be seen which other predicates, which other predicate-argument relationships, and which other NLP problems can benefit from visual information.

References

S. Bergsma and B. Van Durme. 2011. Learning bilingual lexicons using the visual similarity of labeled web images. In *Proc. IJCAI*.

S. Bergsma, D. Lin, and R. Goebel. 2008. Discriminative learning of selectional preference from unlabeled text. In *Proc. EMNLP*, pages 59–68.

N. Chambers and D. Jurafsky. 2010. Improving the use of pseudo-words for evaluating selectional preferences. In *Proc. ACL*, pages 445–453.

O. Chapelle, P. Haffner, and V. Vapnik. 1999. Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks*, 10(5):1055–1064.

S. Clark and D. Weir. 2002. Class-based probability estimation using a semantic hierarchy. *Computational Linguistics*, 28(2):187–206.

I. Dagan and A. Itai. 1990. Automatic processing of large corpora for the resolution of anaphora references. In *Proc. COLING*, pages 330–332.

I. Dagan, L. Lee, and F. C. N. Pereira. 1999. Similarity-based models of word cooccurrence probabilities. *Mach. Learn.*, 34(1-3):43–69.

T. Deselaers, D. Keysers, and H. Ney. 2008. Features for image retrieval: an experimental comparison. *Information Retrieval*, 11:77–107.

K. Erk. 2007. A simple, similarity-based model for selectional preference. In *Proc. ACL*, pages 216–223.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Y. Feng and M. Lapata. 2010a. Topic models for image annotation and text illustration. In *Proc. HLT-NAACL*, pages 831–839.

Y. Feng and M. Lapata. 2010b. Visual information in semantic representation. In *Proc. HLT-NAACL*, pages 91–99.

R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. 2005. Learning object categories from Google's Image Search. In *Proc. ICCV*, pages 1816–1823.

D. Hindle and M. Rooth. 1993. Structural ambiguity and lexical relations. *Computational Linguistics*, 19(1):103–120.

F. Keller and M. Lapata. 2003. Using the web to obtain frequencies for unseen bigrams. *Computational Linguistics*, 29(3):459–484.

D. Lin, K. Church, H. Ji, S. Sekine, D. Yarowsky, S. Bergsma, K. Patil, E. Pitler, R. Lathbury, V. Rao, K. Dalwani, and S. Narsale. 2010. New tools for web-scale N-grams. In *Proc. LREC*, pages 2221–2227.

D. G. Lowe. 2004. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110.

D. Ó Séaghdha. 2010. Latent variable models of selectional preference. In *Proc. ACL*, pages 435–444.

A. Oliva and A. Torralba. 2007. The role of context in object recognition. *Trends in Cognitive Sciences*, 11(12):520–527.

U. Padó, F. Keller, and M. Crocker. 2006. Combining syntax and thematic fit in a probabilistic model of sentence processing. In *Proc. CogSci*, pages 657–662.

P. Resnik. 1996. Selectional constraints: An information-theoretic model and its computational realization. *Cognition*, 61:127–159.

P. Resnik. 1997. Selectional preference and sense disambiguation. In *Proc. ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*

A. Ritter, Mausam, and O. Etzioni. 2010. A latent dirichlet allocation method for selectional preferences. In *Proc. ACL*, pages 424–434.

M. Rooth, S. Riezler, D. Prescher, G. Carroll, and F. Beil. 1999. Inducing a semantically annotated lexicon via EM-based clustering. In *Proc. ACL*, pages 104–111.