Journal of Machine Learning Research 3 (2003) 1107–1135 Submitted 4/02; Published 2/03

Matching Words and Pictures

Kobus Barnard KOBUS@CS.ARIZONA.EDU
Computer Science Department
University of Arizona
Tucson, AZ 85721-0077, USA

Pinar Duygulu DUYGULU@CENG.METU.EDU.TR
Department of Computer Engineering
Middle East Technical University
Ankara, Turkey

David Forsyth DAF@CS.BERKELEY.EDU
Computer Science Division
University of California
Berkeley, CA 94720-1776, USA

Nando de Freitas NANDO@CS.UBC.CA
Department of Computer Science
University of British Columbia
Vancouver, B.C. V6T 1Z4, Canada

David M. Blei BLEI@CS.BERKELEY.EDU
Computer Science Division
University of California
Berkeley, CA 94720-1776, USA

Michael I. Jordan JORDAN@CS.BERKELEY.EDU
Computer Science Division and Department of Statistics
University of California
Berkeley, CA 94720-1776, USA

Editors: Jaz Kandola, Thomas Hofmann, Tomaso Poggio and John Shawe-Taylor

Abstract

We present a new approach for modeling multi-modal data sets, focusing on the specific case of segmented images with associated text. Learning the joint distribution of image regions and words has many applications. We consider in detail predicting words associated with whole images (auto-annotation) and corresponding to particular image regions (region naming). Auto-annotation might help organize and access large collections of images. Region naming is a model of object recognition as a process of translating image regions to words, much as one might translate from one language to another. Learning the relationships between image regions and semantic correlates (words) is an interesting example of multi-modal data mining, particularly because it is typically hard to apply data mining techniques to collections of images. We develop a number of models for the joint distribution of image regions and words, including several which explicitly learn the correspondence between regions and words. We study multi-modal and correspondence extensions to Hofmann's hierarchical clustering/aspect model, a translation model adapted from statistical machine translation (Brown et al.), and a multi-modal extension to mixture of latent Dirichlet allocation (MoM-LDA). All models are assessed using a large collection of annotated images of real scenes. We study in depth the difficult problem of measuring performance. For the annotation task, we look at prediction performance on held out data. We present three alternative measures, oriented toward different types of task. Measuring the performance of correspondence methods is harder, because one must determine whether a word has been placed on the right region of an image. We can use annotation performance as a proxy measure, but accurate measurement requires hand labeled data, and thus must occur on a smaller scale. We show results using both an annotation proxy, and manually labeled data.

1. Introduction

It is a remarkable fact that, while text and images are separately ambiguous, jointly they tend not to be; this is probably because the writers of text descriptions of images tend to leave out what is visually obvious (the color of flowers, etc.) and to mention properties that are very difficult to infer using vision (the species of the flower, say). There are a wide variety of data sets that consist of very large numbers of annotated images. Examples include the Corel data set, most museum image collections (for example, http://www.thinker.org/fam/ thinker.html), the web archive (http://www.archive.org), and most collections of news photographs on the web (which come with captions). Typically, these annotations refer to the content of the annotated image, more or less specifically and more or less comprehensively. For example, the Corel annotations describe specific image content, but not all of it; museum collections are often annotated with some specific material—the artist, date of acquisition, etc.—but often contain some rather abstract material as well. In this paper, we describe a series of models that link images and text in various ways.

1.1 Practical Applications

Very large collections of images are widespread and users would like to be able to browse and to search these collections. A broad range of computer vision methods have been used to search collections of images. Typically, images are matched based on features computed from the entire image or from image regions. The literature is too broad to review here; there are reviews in Forsyth (1999), Forsyth and Ponce (2002). With the exception of systems that can identify faces (Schneiderman and Kanade, 2000), naked people (Fleck et al., 1996), pedestrians (Oren et al., 1997) or cars (Schneiderman and Kanade, 2000), matching is not usually directed toward object semantics. However, user studies show a large disparity between user needs and what technology supplies (Armitage and Enser, 1997, Enser, 1993, 1995). This work makes hair-raising reading—an example is a request to a stock photo library for "Pretty girl doing something active, sporty in a summery setting, beach - not wearing lycra, exercise clothes - more relaxed in tee-shirt. Feature is about deodorant so girl should look active - not sweaty but happy, healthy, carefree - nothing too posed or set up - nice and natural looking."

Other user studies include the work of Ornager (1996), who studied practice at a manually operated newspaper photo archive and Markkula and Sormunen (2000), who study practice at a Finnish newspaper's digital photo archive. Keister studied requests received by the National Library of Medicine's Archive (Keister, 1994). In this literature, authors break out the semantics of the images requested in different ways, but from our perspective the important points are:

- that users request images both by object kinds (a princess) and identities (the princess of Wales);

- that users request images both by what they depict (things visible in the picture) and by what they are about (concepts evoked by what is visible in the picture);
- that queries based on image histograms, texture, overall appearance, etc. are vanishingly uncommon;
- and that text associated with images is extremely useful in practice—for example, newspaper archivists index largely on captions (Markkula and Sormunen, 2000).

There are several practical applications for methods that can link text and images, however imperfectly:

- Automated image annotation: Numerous organizations manage collections of images for internal use. A typical workflow is described by the work of Markkula and Sormunen (2000), who studied the image archive of a Finnish newspaper.[1] Archivists receive pictures and annotate them with words that are likely to be useful keys for retrieving the pictures; journalists then search the collection using these keywords. Annotation is often difficult and uncertain; it would be attractive to have a procedure that annotated images automatically. One might auto-annotate by predicting words with high posterior probability given an image. Examples of automated annotation appear in Barnard et al. (2001), Barnard and Forsyth (2001) and below.
- Browsing support: Museums release parts of their collections onto the web to attract visitors by giving them a sense of what they would see if they visited. Typically users who know a collection well wish to search it, and those who don't, prefer to browse (Frost et al., 2000). This means it would be attractive to organize the collection in a way that made sense to visitors, and so supported browsing. Collecting together images that looked similar and were similarly annotated would be a good start. Fitting a probability model with an appropriate structure yields quite useful clusters, as described in Barnard et al. (2001).
- Auto-illustrate: Commercial image collections can't supply an attractive service to a casual user, because searching the collection is typically difficult and expensive. A tool that could automatically suggest images to illustrate blocks of text might expose value in the collection by making it possible for casual users to get reasonable results cheaply. Auto-illustration is possible if one can obtain images with high probability given text (Barnard et al., 2001, Barnard and Forsyth, 2001).

1.2 Annotation, Correspondence and Recognition

One can currently use words to search for pictures (it is often productive to use a sequence of terms and then 'jpg' or 'jpeg' as a query to Google). There are a variety of ways to use words and pictures simultaneously. The most straightforward is to search using a simple conjunction of keywords and image region features, a facility provided in Blobworld (Carson et al., 2002). Webseer (Swain et al., 1996) uses similar ideas for query of images on the web, but also indexes the results of a few automatically estimated image features. These include whether the image is a photograph or a sketch and notably the output of a face finder. Going further, Cascia et al. integrate some

1. Also see Enser's work (Armitage and Enser, 1997, Enser, 1993, 1995) on various image archives, which use roughly the same procedure.

text and histogram data in the indexing (La Cascia et al., 1998). Others have also experimented with using image features as part of a query refinement process (Chen et al., 1999, 2000). Srihari and others have used text information to disambiguate image features, particularly in face finding applications (Srihari et al., 1994, Srihari, 1991, Srihari and Burhans, 1994).

We are not aware of general probability models that link text and images, however. Such a model would offer the usual benefits of probability models over boolean queries—one doesn't need to know exactly the right search terms to get useful results—but might offer more. One is that one might predict text given images. There are two ways to do this. Firstly, one might attempt to predict annotations of entire images using all information present. We refer to this task as **annotation**. Secondly, one might attempt to associate particular words with particular image substructures—that is, to infer **correspondence**.

Few data sets contain correspondence information, probably because it is difficult to create such data sets. Normally, one has a collection of images, each of which has a collection of associated words. This can be seen as a form of classification problem, where instead of having labeled examples one has labeled bags of examples—an image is "positive" if it contains a tiger somewhere amongst all the other stuff and "negative" if it doesn't. Maron and Ratan (1998) and Maron (1998) used multiple-instance learning to train classifiers to identify particular keywords from image data using such bags. Rather than attempt to sort out all correspondences between image structures and words directly, they build classifiers for each word separately. Satoh and Kanade (1997) used co-occurrence models for automatically associating faces with names in video. Finally, perhaps closest to our work on predicting words for regions is the work of Mori et al. (1999), where co-occurrence statistics are collected for words and image areas defined by a fixed grid.

Correspondence is a peculiar feature of object recognition. Current theories of object recognition reason either in terms of geometric correspondence and pose consistency; in terms of template matching via classifiers; or by search to establish the presence of suggestive relations between templates. A detailed review of these strategies appears in Forsyth and Ponce (2002). There has been little work to address object recognition at a broad scale. For example, not much is known on how to recognize thousands of different objects from data sets that are practically available. Little can be said about what is easy and what is hard to recognize using a particular set of features. It is reasonable to hope that these questions can be discussed if one sees object recognition as a process by which one uses a huge data set to learn to put image structures and words in correspondence.

This paper explores a variety of latent variable models that can be used for auto-illustration, annotation and correspondence. The first step is to represent image information; Section 2 describes the representation we have adopted. In Section 3 we describe a series of models that link words and image data, without explicit encoding of correspondence between words and regions. These models are appropriate for auto-illustration and annotation.

There is an analogy between learning a correspondence model that can associate words with image regions and learning a lexicon, which suggests it is possible to build a process that uses rather little supervisory input. In effect, one builds a model using unsupervised methods, marks up the model's output, and refits. This is a standard process in the machine translation literature (a good guide is Melamed's thesis, 2001; see also Jurafsky and Martin, 2000, and Manning and Schütze, 1999). In Section 4 we describe a model that uses a vector quantized representation of image regions to yield a problem exactly analogous with lexicon learning. This is the simplest model that learns correspondence explicitly. It has the disadvantage that the representation of image regions is obtained independent of the text annotation. This ignores potentially important data about what image differences are important. More sophisticated correspondence models use text data while simultaneously clustering representations of image regions (Section 5).

Part of this paper's role is to compare numerous models, meaning that there are several different variants of each type of model. To keep track of them all, each is allocated an acronym, which appears in bold the first time it is used. Annotation performance is fairly straightforward to evaluate automatically and so large data sets can be used, but correspondence performance is rather more difficult to evaluate. Since large data sets giving correspondence don't exist, we are obliged to evaluate correspondence manually; this means evaluation can't be done on a large scale. Section 6 describes our methods of evaluation, and Section 7 gives extensive comparison of the methods.

2. Input Representation and Preprocessing

Each image is segmented using normalized cuts (Shi and Malik, 2000). This segmenter has the occasional tendency to produce small, typically unstable regions. We represent the 8 largest regions in each image by computing, for each region, a set of 40 features. The features represent, rather roughly, major visual properties:

- Size is represented by the portion of the image covered by the region
- Position is represented using the coordinates of the region center of mass normalized by the image dimensions
- Color is (redundantly) represented using the average and standard deviation of (R,G,B), (L,a,b) and (r=R/(R+G+B), g=G/(R+G+B)) over the region.
- Texture is represented using the average and variance of 16 filter responses. We use 4 difference of Gaussian filters with different sigmas, and 12 oriented filters, aligned in 30 degree increments. See Shi and Malik (2000) for additional details and references on this approach to texture.
- Shape is represented by the ratio of the area to the perimeter squared, the moment of inertia (about the center of mass), and the ratio of the region area to that of its convex hull.

We will refer to a region, together with the features, as a blob. We make no claim that the image features adopted are canonical. They are chosen to be computable for any image region, and be independent of any recognition hypothesis. We expect that better or worse behavior would be available using different sets of image features. It remains an interesting open question to construct feature sets that (a) offer very good performance for a particular vision task and (b) can depend on an emerging object hypothesis in an interesting and efficient way.

3. Annotation Models

We present two classes of models for the joint distribution of text and blobs, and show how they are applied to annotate images.

3.1 Multi-Modal Hierarchical Aspect Models

Our first model is a multi-modal extension of Hofmann's hierarchical model for text (Hofmann, 1998, Hofmann and Puzicha, 1998). This model combines the aspect model with a soft clustering

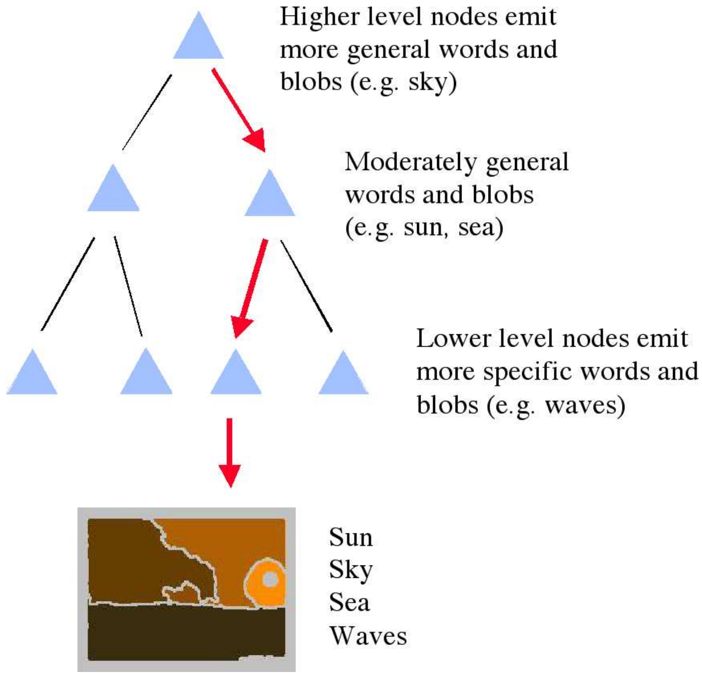

Figure 1: Multi-modal extension of a hierarchical model for text.

model. As shown in Figure 1, images and co-occurring text are generated by nodes arranged in a tree structure. The nodes generate both image regions using a Gaussian distribution, and words using a multinomial distribution. Each cluster is associated with a path from a leaf to the root. Nodes close to the root are shared by many clusters, and nodes closer to leaves are shared by few clusters. This means that in a properly fitted model, nodes closer to the root tend to emit items (words or regions) shared by a large number of data elements, and the nodes closer to the leaves each emit items more specific to small numbers of data elements.

Potentially, both the vertical structure (aspects), and horizontal structure (clusters), can help model the distributions of interest. Our implementation supports all tree topologies, including the degenerate "linear" topology where there are no branches and therefore only one cluster, as well as topologies which have no vertical structure and thus all modeling is through clustering. In the experiments we consider a binary tree, and the linear case with a comparable number of nodes.

To the extent that an image is in a given cluster, it is generated by the nodes on that path. Taking all clusters into consideration, a document is modeled by a sum over the clusters, weighted by the probability that the document is in the cluster. The process for generating the set of observations D associated with a document, d, can be described by:

$$p(D|d) = \sum_c p(c) \prod_{w \in W} \left[\sum_l p(w|l,c) p(l|d) \right]^{\frac{N_w}{N_{w,d}}} \prod_{b \in B} \left[\sum_l p(b|l,c) p(l|d) \right]^{\frac{N_b}{N_{b,d}}}, \tag{1}$$

where c indexes clusters, w indexes the words in document d, b indexes the image regions in document d, and l indexes levels. D is the set of observations for the document, W is the set of words for the document, B is the set of blobs for the document, with $D = W \cup B$. The exponents $\frac{N_w}{N_{w,d}}$

and $\frac{N_b}{N_{b,d}}$ are introduced to normalize for differing numbers of words and blobs in each image. $N_{w,d}$ denotes the number of words in document d, while N_w denotes the maximum number of words in any document. This normalization works fine for the Corel data because there is not much variance in $N_{w,d}$. The same applies for blobs. This choice essentially makes an image with only the word "sun" comparable with one with the words "sun" and "clouds," by duplicating the word "sun" for the first image.

For word emission probabilities, $p(w|l,c)$, we use frequency tables, and for blob emission probabilities, $p(b|l,c)$, we use a Gaussian distribution with diagonal covariance over the features for the regions. To stabilize training, we translate and scale the region feature data to have zero mean and unit variance. The model parameters are converted into the original data space once training is complete. We also limit the variance of the Gaussian distribution to be at least 0.001 in the training data space. Similarly, the word frequency is forced to be at least a small value greater than zero (0.01 / vocabulary size), but here the goal is simply to avoid the need to do certain computations in log space. $p(l|d)$ is a training document specific prior over the nodes on the path from the leaf to the root (vertical weights).

We will refer to this as **model I-0** in the results (I for "independent"). Following Barnard et al. (2001) and Barnard and Forsyth (2001), we also experiment with allowing a cluster dependent level structure. Here $p(l|d)$ is replaced with $p(l|c,d)$ (**model I-1**). Notice that these models are not true generative models because the joint probability distribution of the image items is described in terms of $p(l|d)$ or $p(l|c,d)$ which are specific to the documents in the training set. This makes the model powerful for search applications. However, prediction is difficult for documents not in the training set. One can marginalize out the training data, as in Blei et al. (2002). An alternative is to estimate the mixing weights using a cluster specific average computed during training. This latter strategy appears to work well, suggesting that the set of vertical nodes used to model a document is more significant than the mixing weights.

This observation led to the alternative model—first described in Barnard et al. (2001)—which is generative. This model—which we call **model I-2**—gives

$$p(D) = \sum_c p(c) \prod_{w \in W} \left[\sum_l p(w|l,c)p(l|c) \right]^{\frac{N_w}{N_{w,d}}} \prod_{b \in B} \left[\sum_l p(b|l,c)p(l|c) \right]^{\frac{N_b}{N_{b,d}}}. \tag{2}$$

Model Fitting. All of these models are fit using the expectation maximization algorithm (Dempster et al. (1977)). The update equations are very similar to those in Hofmann and Puzicha (1998), except of course, the probability expressions now include parts both for word and region occurrences. For model I-0, we estimate the vertical mixing weights for each document as in Hofmann and Puzicha (1998). For model I-1, we estimate the vertical mixing weights for each document given *each* cluster. For model I-2, the update equations for the vertical mixing weights are simpler, as we just need to compute a cluster dependent average, rather than estimating quantities for each document. This can be a significant memory saving if the number of documents is large.

Image Based Word Prediction. To predict words from images we assume that we have a new document with a set of observed blobs, B. We wish to compute $p(w|B) \propto p(w,B)$ for each word, w, in our vocabulary:

$$
\begin{aligned}
p(w|B) &\propto \sum_c p(c)p(w|c)p(B|c) \\
&= \sum_c p(c)\left[\sum_l p(w|l,c)p(l|c)\right]\prod_{b\in B}\left[\sum_l p(b|l,c)p(l|c)\right]^{\frac{N_b}{N_{b,d}}}.
\end{aligned}
\tag{3}
$$

In the case of models I-0 and I-1 we drop the document index, d, from the vertical weights, as we are normally interested in applying (3) to documents outside the training set. Further note that for model I-0, $p(l|c)$ is replaced by $p(l)$. For the vertical weights we either use cluster specific average mixing weights (labeled "**ave-vert**" in the results), or estimate $p(l|d)$ by $p(l|B)$ (labeled "**doc-vert**"). In previous work we instead estimated $p(l|d)$ by refitting the model based on B, but this is more expensive and does not give better results. Marginalizing out the training data worked at least as well as "ave-vert" on small data sets, but it is very expensive to compute on data of the scale of interest, and thus we do not report results here.

3.2 Mixture of Multi-Modal Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) (Blei et al., 2002) is a generative probabilistic model for independent *collections* of data where each collection is modeled by a randomly generated mixture over latent factors. For example, in text modeling, a collection of words makes up a document and LDA provides a model of independently generated documents (a corpus).

As a generative model, LDA can readily be used as a module in a mixture model. Furthermore, LDA is extendable to multi-modal data. In particular, the mixture of multi-modal LDA model (**MoM-LDA**) assumes that each image and corresponding words were generated by the following process:

1. Choose one of J mixture components $c \sim \text{Multinomial}(\eta)$.
2. Conditioned on the mixture component, choose a mixture over J factors, $\theta \sim \text{Dir}(\alpha_c)$.
3. For each of the N words:
   (a) Choose one of K factors $z_n \sim \text{Multinomial}(\theta)$.
   (b) Choose one of V words w_n from $p(w_n|z_n, c, \beta)$, the conditional probability of w_n given the mixture component and latent factor.
4. For each of the M blobs:
   (a) Choose a factor $s_m \sim \text{Multinomial}(\theta)$.
   (b) Choose a blob b_m from $p(b_m|s_m, c, \mu, \Sigma)$, a multivariate Gaussian distribution with diagonal covariance, conditioned on the factor s_m and the mixture component c.

This is depicted as a graphical model in Figure 2. The parameters to MoM-LDA are

- A J-dimensional multinomial parameter η.
- A $J \times K$ matrix α where α_c is is the J-dimensional Dirichlet parameter conditioned on mixture component.

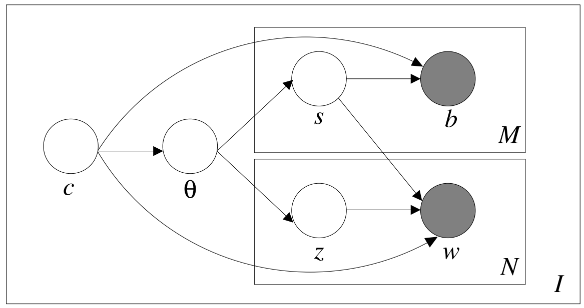

Figure 2: The mixture of multi-modal latent Dirichlet allocation model. The outer plate represents the repetition of I images. Each image has M blobs and N words. The parameters of the model are not depicted, for simplicity.

- A $J \times K \times V$ matrix β where β_{cz} is the distribution over words conditioned on the mixture component and hidden factor.
- A $J \times K \times D$ matrix μ and a $J \times K \times D$ matrix Σ where μ_{cs} and Σ_{cs} are parameters to the D-dimensional multivariate Gaussian distribution over blobs, conditioned on the mixture component and hidden factor.

Maximum likelihood estimates of the Dirichlet, word multinomials, and Gaussian parameters can be obtained by the EM algorithm with a variational E step.

Given an image and a MoM-LDA, we can compute both an approximate posterior over mixture components and, for each mixture component, an approximate posterior Dirichlet over factors. Using these parameters, we perform image based word-prediction by finding the corresponding distribution over words. Let ϕ denote the approximate posterior over mixture components, and γ_c denote the corresponding approximate posterior Dirichlet. The distribution over words given an image (that is, a collection of blobs) is:

$$p(w|b) = \sum_{c=1}^{J} p(c|\phi) \sum_{z=1}^{K} p(w|z) \int p(z|\theta) p(\theta|\gamma_c) d\theta.$$

The integral over θ is easily computed; it is the expectation of the zth component of $\theta \sim \mathrm{Dir}(\gamma_c)$:

$$\int p(z|\theta) p(\theta|\gamma_c) d\theta = \frac{\gamma_{cz}}{\sum_{y=1}^{K} \gamma_{cy}}.$$

MoM-LDA is similar to the I-0 model in that it derives its predictive abilities from the higher level mixture component c. The underlying LDA is useful as a joint model but, on its own, cannot accurately predict words from images. This is due to the implicit assumption that words and blobs are exchangeable and thus can be generated in any order. This issue is described in Blei and Jordan (2002), where the authors derive a LDA-based model of annotated data that is based on *partial exchangeability*. Images are generated first, and words are subsequently generated from the images. The resulting model can predict words from images without resorting to higher level multinomial mixture components.

4. Simple Correspondence Models

It is natural to want to build models that can predict words for specific image regions, rather than for a whole image. There are several simple ways to do so. The simplest is to vector-quantize representations of image regions, and then directly exploit the analogy with statistical lexicon learning. In fact, hierarchical aspect models and MoM-LDA can yield correspondence information, too.

4.1 Discrete Data Translation

In machine translation a lexicon links discrete objects (words in one language) to discrete objects (words in the other language). We must come up with a lexicon given an aligned bitext, which consists of many small blocks of text in both languages, which are known to correspond in meaning. A traditional example is Hansard for the Canadian parliament, where each speaker's remarks in French and in English correspond in meaning. Assuming an unknown one-one correspondence between words, coming up with a joint probability distribution linking words in the two languages is a missing data problem (Brown et al., 1993). It is straightforward to create analogous image data. We use K-means to vector-quantize the set of features representing an image region. Each region then gets a single label (blob token).

We now have an aligned bitext consisting of the blobs and the words for each image. We must construct a joint probability table linking word tokens (the abstract model of a word, as opposed to an instance) to blob tokens. In the current work, we use all keywords associated with each image. Because the data set does not provide explicit correspondences, we have a missing data problem which is easily dealt with as an application of EM (see Duygulu et al., 2002 for details). We label this approach as **discrete-translation**.

4.2 Correspondence from a Hierarchical Clustering Model

Our hierarchical clustering models do not model the relationships between specific image regions and words explicitly. However, they do encode this correspondence to some extent through co-occurrence because there is a advantage to having "topics" collect at the nodes. For example, if the word "tiger" always co-occurs with an orange stripy region and never otherwise, then these items will likely be generated by a shared node, as there are far fewer nodes than observations. Thus we have the following simple word prediction method for a single blob, b:

$$p(w|b) \propto \sum_{c} p(c) \sum_{l} p(l) p(w|l,c) p(b|l,c). \tag{4}$$

We can further consider the effect of the other regions through what they say about the cluster membership by replacing the cluster prior, $p(c)$ with $p(c|B)$. We label these strategies as "region-only" and "region-cluster" in the results. Notice that using (4) is not quite the same as replacing the set of blobs, B, in (3) with the single blob of interest, b. Here we insist that the word and the region come from the same node, whereas in (3) they come from the same cluster, but possibly from different nodes associated with that cluster. Thus to get correspondence from I-0, I-1, and I-2, we use the models differently from how they are trained.

5. Integrating Correspondence and Hierarchical Clustering

None of the methods described above are wholly satisfactory for learning correspondence. By vector-quantizing image regions independent of words, we are perversely ignoring potentially useful training data. We should not expect too much from the other methods, because they are being compelled to reveal information they were not trained to represent. However, there is a relationship between clustering and correspondence. For example, in translating from English to French, the word "sun" might be translated to "soleil," but if the surrounding words are computer related, then the word "sun" should remain unchanged, as it is likely a brand of computer. Similarly, a gray patch is more likely to be pavement in a city scene than a jungle scene. This suggests building explicit correspondence information into our existing hierarchical clustering models. Building correspondence models involves strengthening the relationship between words and image regions.

5.1 Linking Word Emission and Region Emission Probabilities with Mixture Weights

In this approach, image regions are modeled as in the independent model, but the words are not emitted conditioned on the regions. To implement this strategy by having the vertical mixture weights for the regions carries over to that for the words. Thus if a node contributes little to the region emission for an image, then that node will also contribute little to the word emission. Correspondence is still implicit, and is calculated using (4) above, and thus this is not a true correspondence method. More formally, we consider the words W and the regions S in $D = B \cup W$ to be handled asymmetrically.

$$p(D|d) = \sum_{c} p(c) \prod_{w \in W} \left[\sum_{l} p(w|l,c) p(l|B,c,d) \right]^{\frac{N_w}{N_{w,d}}} \prod_{b \in B} \left[\sum_{l} p(b|l,c) p(l|d) \right]^{\frac{N_b}{N_{b,d}}}, \qquad (5)$$

where we stipulate that

$$p(l|B,c,d) \propto \sum_{b \in B} p(l|b,c,d).$$

Notice that we have chosen to compute the distribution inherited by the words on a cluster by cluster basis (we use $p(l|B,c,d)$ instead of $p(l|B,d)$). We denote this model by **D-0** (D for dependent). In analogy with the independent case, we can consider cluster dependent level distributions, using $p(l|c,d)$ instead of $p(l|d)$ to get **D-1**, or we can drop the dependency on the training set, using $p(l)$ instead of $p(l|d)$ to get **D-2**.

To use the model for annotation, we again estimate $p(l|d)$ by $p(l|B)$ ("doc-vert"), and use this to compute the word posterior. For labeling regions, applying (4) to predict the words for a particular blob makes more sense than with the independent models because now words are emitted conditioned on blobs. As described later, if we wish to use such a model for annotation, we simply sum the contributions of all the blobs. Interestingly, this gives almost the same equations for word prediction (equivalent if there is only once cluster, as in the linear topology).

5.2 Paired Word and Region Emission at Nodes

Our second method further tightens the relationship between the regions and words. Here we assume that the observed words and regions are emitted in pairs so that $D = \{(w,b)_i\}$ and (1) becomes:

$$p(D|d) = \sum_c p(c) \prod_{(w,b)\in D} \left[\sum_l p((w,b)|l,c)p(l|d) \right]. \tag{6}$$

We will refer to this as Model **C-0** (C for "correspondence"). Models I-1 and I-2 are similarly modified to get models **C-1** and **C-2**. Equation (6) evaluates the likelihood assuming a proposed correspondence. Since we are most interested in training on data where the correspondence is not provided, we need to estimate correspondence as part of the training process. We have experimented with several methods for doing so, discussed below. Regardless, this additional step is added before the E step in the model fitting process, and the assignment is then used in the estimation of the expectation of the indicator variables. All methods for computing the correspondence assume that the probability that a word and a segment correspond can be estimated by the probability that they are emitted from the same node. Using $w \Leftrightarrow b$ to denote that the word, w, and the region, b, correspond, we use, in the case of C-0:

$$p(w \Leftrightarrow b) \approx \sum_c p(c) \sum_l p((w,b)|l,c)p(l|d). \tag{7}$$

Here we have made the choice that the proposed correspondence should be shared over all cluster hypotheses. Doing so makes intuitive sense, and significantly reduces the number of correspondence estimates that are required. Note that training is no longer pure gradient descent (the log-likelihood can decrease a small amount as the training process iterates). For true gradient descent, we would need to marginalize over possible correspondences, and this is impractical. Instead, we approximate the sum with that obtained by a maximal, or close to maximal match (this approximation is common in the literature on statistical learning of lexicons, see for example Melamed, 2001).

We have experimented with several possible strategies for estimating matches from (7). In this work we report results using graph matching (Jonker and Volgenant, 1987). This algorithm gives a polynomial time method to assign edges to a bipartite graph so that each vertex is connected to only vertex, and the sum of costs associated with each possible edge is minimized. Our costs are negative log probabilities from (7), so that likelihood is maximized.

Regardless of the matching strategy, we need to deal with differing numbers of words and regions. In our implementation, we first ensure that there are more regions than words by repeating the region collection if needed. Then we ensure that there are sufficient words by repeating the word collection until there are as many words as regions.

5.3 Correspondence Models, NULL, Fertility and Refusal to Predict

Correspondence comes with a variety of annoying difficulties which we have skated over above. The primary issue is the choice of correspondence model. Should there be a one-one map between regions and words (usually impossible, because of the numbers are different)? In the work described here, we require regions to be linked to words; there is no option of deciding that a region corresponds to no word. This forces models of image regions corresponding to particular words to cope with a large pool of outliers. This problem could, in principle, be handled by appending a special word, NULL, to the text of each data item and a special image region, NULL, to the image regions of each data item. This is a traditional solution in the machine translation literature; the tendency of single words in some languages to generate more than one word in others (a property referred to as "fertility") can be modeled explicitly in this framework (Brown et al., 1993, Melamed, 2001). In our limited experience, such models are not easy to fit to our data sets, because of a tendency to fit a model where every word is generated by the NULL image regions and every image region generates the NULL word. This is clearly a matter to be resolved by a prior model of deletion of words or image regions, respectively. One complication is that the probability that an annotation is absent is not independent of the annotation—annotators always mention "tigers," but only sometimes mention "people." A simple strategy that offers some benefits of directly modeling NULL words is to refuse to predict an annotation when the annotation with the highest probability given the region has too low a probability; this discourages predictions by regions whose identity is moot. This is crude, because it doesn't mitigate the effect of all the outliers in the fitting process.

6. Evaluation Methods

Our annotation models can be used on two different kinds of data. First, we can try to annotate images which were well represented in the original data set, for example, annotating images arriving at an archive. Second, we can try to annotate images from a collection which is not well represented by the original training data, for example, performing object recognition.

Correspondence models present further difficulties. The issue now is whether we predict appropriate words for each particular region. Typically, the only way to obtain an accurate answer to this question is to look at the picture. This form of manual evaluation is very difficult to do for a satisfactory number of images. A less strict, but nonetheless informative, test is to determine the annotation performance for a correspondence model, on the grounds that poor annotation performance implies poor correspondence performance (crucially, the contrapositive is not necessarily true).

6.1 Measuring Annotation Performance

We can measure annotation performance by comparing the words predicted by various models with words actually present for held-out data. In most data sets, including ours, image annotations typically omit some obviously appropriate words. However, since our purpose is to compare methods this is not a significant problem as each model must cope with the same set of missing annotations. Performance comparisons can be carried out automatically and therefore on a substantial scale. We express prediction performance relative to predictions obtained using the empirical word frequency of the training set. Matching the performance empirical density is required to demonstrate non-trivial learning. Doing substantially better than this on the Corel data is difficult. The annotators typically provide several common words (for example, "sky," "water," "people"), and fewer less common words (for example, "tiger"). This means that annotating all images with, say, "sky," "water," and "people" is quite a successful strategy. Performance using the empirical word frequency would be reduced if the empirical density was flatter. Thus for this data set, the increment of performance over the empirical density is a sensible indicator. We look at word prediction on held out data, and rank models using three measures.

6.1.1 MEASURING THE QUALITY OF WORD POSTERIOR DISTRIBUTION

As our models are predictive in nature, we estimate the Kullback-Leibler (KL) divergence between the computed predictive distribution, $q(w|B)$, and the target distribution, $p(w)$. Unfortunately, the target distribution is not known, and for this we simply assume that the actual words should be predicted uniformly, and that all other words should not be predicted at all. By definition, the error contribution for one document with this measure, $E_{KL}^{(model)}$, is given by:

$$E_{KL}^{(model)} = \sum_{w \in vocabulary} p(w) \log \frac{p(w)}{q(w|B)}. \tag{8}$$

To further smooth $q(w|B)$, we add the minimum of the empirical word distribution and renormalize. To compute a combined measure for a group of images, we simply average the quantity in (8) over that set. We can relate this to the conditional log likelihood (normally computed on held out data). If there are K words for the image under consideration, Since $p(w) = 1/K$ for $w \in observed$, and 0 otherwise, and we declare $0 \log \frac{0}{q} = 0$,

$$\begin{aligned} E_{KL}^{(model)} &= \frac{1}{K} \sum_{w \in observed} \log \frac{p(w)}{q(w|B)} \\ &= constant - \frac{1}{K} \sum_{w \in observed} \log q(w|B). \end{aligned}$$

When averaged over the images in the held out set, this is essentially the held out log likelihood, conditioned on the image regions, weighted so that each image contributes roughly the same, regardless of the number of words it has. As mentioned above, we express performance relative to that using the empirical word distribution of the test data, and we further arrange it so that larger values correspond to better performance. Specifically, we report:

$$E_{KL} = \frac{1}{N} \sum_{data} \left(E_{KL}^{(empirical)} - E_{KL}^{(model)} \right).$$

which is negative when the model is worse than the prior and positive when it is better. This adjustment has several benefits. First, it is immediately clear when we are doing well (the number is positive), and second, it reduces the variance of values computed on different sets, as the difficulty of the set is partly reflected in $E_{KL}^{(empirical)}$.

6.1.2 HOW WELL DO MODELS PREDICT WORDS?

On the whole, a better fitting model should predict a better set of words, but we need some concrete measurement of the goodness of the omitted words. The difficulty here is that one needs a loss function, and traditional zero-one loss is highly misleading. For most conceivable applications, certain errors (“cat” for “tiger”) are less offensive than others (“car” for “vegetable”). Because the number of classes that we can predict is large (the size of the vocabulary), we normalize the correct and incorrect classifications.

Specifically, we compute $E_{NS}^{(model)} = r/n - w/(N-n)$ where N is the vocabulary size, n is the number of actual words for the image, r is the number of words predicted correctly, and w is the number of words predicted incorrectly. This score gives a value of 0 for both predicting everything and predicting nothing, and 1 for predicting exactly the actual word set (no false positives, no false negatives). The score for predicting exactly the complement of the actual word set is -1. As in the KL case, we report the difference of this error and that for the empirical word distribution, such that larger values correspond to better performance (that is $E_{NS}^{(model)} - E_{NS}^{(empirical)}$)

The number of words predicted, $r+w$, can be determined by the algorithm on a case by case basis. Thus one benefit of this measure over simply counting the number of correct words in a fixed number of guesses is that it can be used to reward a good estimate of how many words to predict. The word prediction scores reported here are based on predicting all words which exceed a certain probability threshold.

As is clear from Figure 4, a value for the threshold which maximizes the performance of the comparison method (training data word frequency) is also a good value for most other methods of word prediction, and therefore we used this value computed on training data for the reported results. Finally, we report the results using a simpler but related word prediction measure, $E_{PR}^{(model)} = r/n$, based on the n best words. Thus if there are three keywords, "sky," "water," and "sun," then $n = 3$, and we allow the models to predict 3 words for that image. The range of this score is clearly from 0 to 1.

6.2 Measuring Correspondence Performance

Measuring the performance of methods that predict a specific correspondence between regions and words is difficult, because images must be checked by hand. This limits the size of the pool that can be used, and also means that measurements may contain significant noise (it is surprisingly difficult to establish, and stick to, an exact policy about what regions should carry, say, the label "people"). However, we can use a region based method for annotation by summing over the word posteriors for all the regions. Furthermore, we can reasonably expect that a method that cannot predict annotations accurately is unlikely to predict correspondence well. This means that annotation measures offer a plausible proxy.

6.2.1 Using Annotation as a Proxy

We report results using both the image based and region based word prediction methods. For the image based methods, we can (and do) compute the annotations in the natural way. However, a second strategy is to use these methods as region based methods, and then compute the image annotations as for the region based methods. Recall that all our annotation models can provide correspondence, despite not being explicitly trained to do so (we identify this by the suffixes "region-only" and "region-cluster"). When we compute region word posteriors using the image based annotation methods, the word posteriors do not necessarily sum to one because there is no requirement in these models that each region emits any word. However, we enforce this requirement by normalizing the posteriors for each region before summing them. The annotation performance of correspondence models can be assessed by marginalizing out the correspondence from the model, and then testing the model as an annotation model. This means that the measures of error described above apply in a straightforward fashion. Notice that this does not test the correspondence component of the model, but a model that performs poorly by this measure is likely to be unhelpful as a correspondence model.

6.2.2 MANUAL CORRESPONDENCE SCORING

To corroborate the above measure, we also score some correspondence results by hand. While this method directly looks at the correspondence, it does require human judgment. We hand labeled each region in a number of images with every appropriate word in the vocabulary. We insisted that the region has a plausible visual connection to the chosen words. Thus the word "ocean" for "coral" would be judged incorrectly because the ocean is transparent. Other difficulties include words like "landscape" and "valley" which normally apply to larger areas than our regions, and "pattern" which can arguably be designated as correct whenever it appears, but we scored it as incorrect because "pattern" recognition isn't particularly helpful. Some regions could not be linked with any vocabulary term, and these regions were omitted from consideration in computing the scores. Producing the labeled data set is clearly a time consuming and error prone process, and thus we are only able to use this ground truth for a modest number of images (50 images for each of ten test sets). With the hand labeled set, we are able to compute the same measures as for the image annotation case, although over a much smaller test set.

7. Experiments

For our experiments we used images from 160 CD's from the Corel image data set. Each CD has 100 images on one relatively specific topic such as "aircraft." From the 160 CD's we drew samples of 80 CD's, and these sets were further divided up into training (75%) and "standard" held out (25%) sets. The images from the remaining CD's formed a more difficult "novel" held out set. Predicting words for these images is difficult, as we can only reasonably expect success on quite generic regions such as "sky" and "water"—everything else is noise. Each such sample was given to each process under consideration, and the results of 10 of such samples were averaged. This controls for both the input data and EM initialization. Images were segmented using N-Cuts (Shi and Malik, 2000). We excluded words which occurred less than 20 times in the test set, which yielded vocabularies of the order of 155 words. We used a modest selection of features for each segment, including size, position, color, oriented energy (12 filters), and a few simple shape features. For the discrete translation model, we used 500 clusters for vector quantization. For linear topologies we used 500 nodes, and the trees were binary trees with 9 levels (511 nodes). For the MoM-LDA method, we used 50 mixture components and 10 latent factors.

7.1 Annotation Results

We first looked a performance of the hierarchical clustering based methods as a function of the number of EM iterations used to train the models. This is important to check for over-fitting, especially for the models which are not truly generative. We take one preemptive strategy to reduce fitting problems. We train the models on a subset of the data for a few iterations, and use that model as the starting point for training on another subset of the data. This is repeated 5 times, to obtain an initial point for training the full data. Preliminary experiments indicated that there was generally a slight benefit for doing so. We have yet to experiment either with tempering the training as suggested in Hofmann and Puzicha (1998) or stochastic versions of EM (Celeux et al., 1995).

7.1.1 THE NUMBER OF TRAINING ITERATIONS, AND OVER FITTING

Figure 3 plots performance in terms of KL divergence-based error for several models as a function of the number of training iterations. We limit the plots to 30 iterations, but we have verified that the trend established by the first 30 iterations holds even if the training continues for hundreds of iterations. In general, the results indicate that over-fitting is not a big problem.

As one would expect, performance on the training set generally improved with the number of iterations, with one exception. In the case of I-0, with several inference procedures, performance on the training set reached a peak and then decreased. This is because the methods used for reasonably fast inference were necessarily ad hoc, as I-0 is not generative. Similar behavior is also possible (but intuitively less likely) with D-0 and C-0; we did not observe this behavior for these models. For the held out sets, we found that most of the possible benefit was reached after 10 iterations of the training algorithm, and in many cases, dropped off after that (most severely in the case of I-0). As one would also expect, the performance on the novel set—containing images from CD's not represented in the training set—dropped much faster than that of the standard held out set. Evaluating the model on the novel sets is a test of its ability to learn properties of the data that generalize to very different images. For simplicity the rest of the results reported in this paper are for 10 iterations (except MoM-LDA which was run to convergence). 10 iterations is roughly optimal for the standard held out data, and sub-optimal for the novel held out data (5 iterations would give better results).

7.1.2 SCORING ANNOTATIONS WITH THE NORMALIZED SCORE, AND THE EFFECT OF REFUSAL TO PREDICT

Next we studied the behavior of our normalized score measure (a score comparing the predicted words with those actually present, as in 6.1.2) as a function of the minimal probability required to predict words (Figure 4). With this measure predicting either no words or all words gives zero. Therefore, as expected, the general behavior is to go from zero to some peak, and then to drop down to zero again. The peak found using the training data empirical word distribution is used to set a conservative refuse to predict level used for Table 2.

7.1.3 COMPARISON OF MODELS USING DIFFERENT SCORES

We provide comprehensive annotation results in Table 1 (for the prediction score of 6.1.2, PR, $E_{PR}^{(model)} - E_{PR}^{(empirical)}$), Table 2 (for the normalized score of 6.1.2, NS, $E_{NS}^{(model)} - E_{NS}^{(empirical)}$) and Table 3 (for the KL score of 6.1.1, $E_{KL}^{(model)} - E_{KL}^{(empirical)}$). Models I-1 and D-1 are omitted, as there performance is similar to that of models I-0 and D-0. (I-1 (D-1) is a bit better than I-0 (D-1) on training data, slightly better on held out data, and slightly worse on the novel set. This is not surprising given that I-I (D-I) has more parameters than I-0 (D-0)). For the linear topologies, we give results for only one of the four inference methods used in the case of clusters. The "ave-vert" method does not make sense without more than one cluster (and gives poor results), and the other three methods are equivalent in the case of a single cluster.

Close study reveals that the results are far from consistent across the three measures. Taking a broad view, the hierarchical clustering based methods give surprisingly similar results when paired with a reasonable inference strategy (note error estimates provided in parenthesis). Perhaps surprisingly, results with linear arrangements of the nodes are also roughly comparable. This is discussed further below.

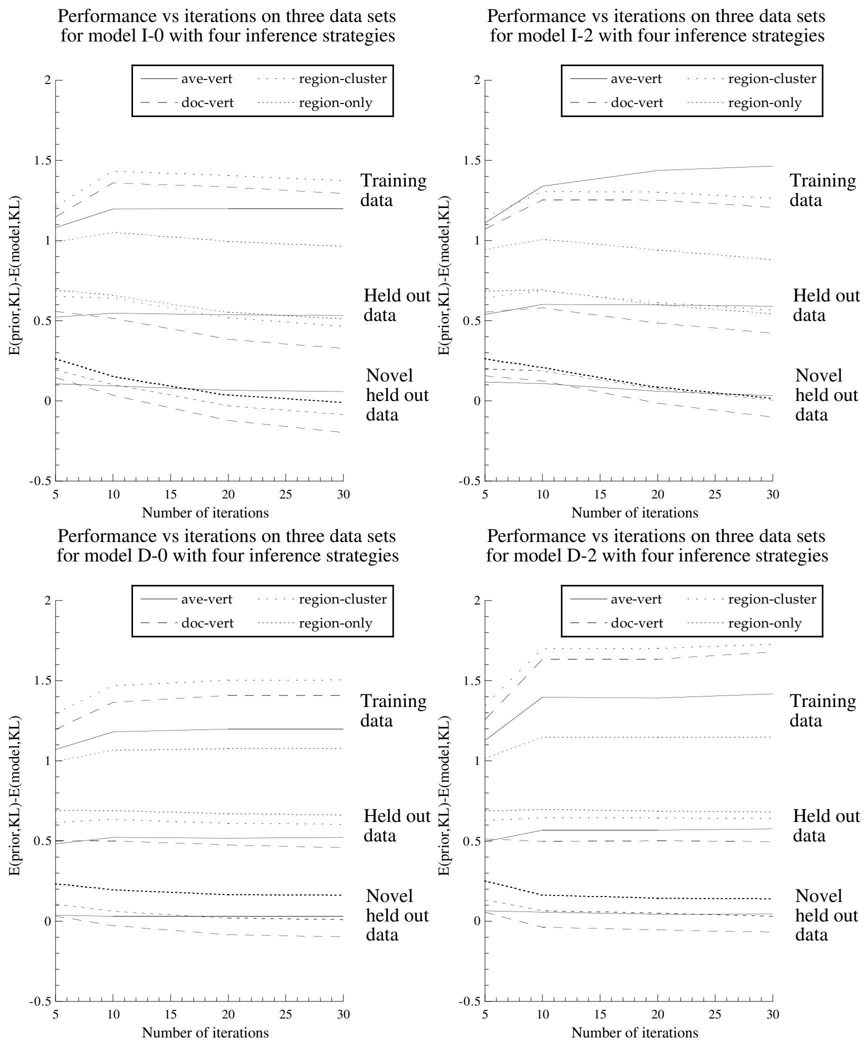

Figure 3: Annotation performance using the KL-divergence between predictive density and the empirical word occurrence density, for 4 models as a function of the number of training iterations. Performance is relative to that for the empirical word distribution, with the vertical axis being the extent to which using one of the models is better than using the empirical distribution (bigger is better). The results are the average of 10 runs with different training and test sets. Performance is shown using three different test sets: the training set, a held out set, and a held out set which is substantially different in character from the training set. Notice that for I-0, the performance on the training set actually decreases with increasing iterations after the peak. This is due to the ad hoc inference methods introduced to efficiently compute the required distribution despite the fact that the model is not truly generative. These plots show that for our task, most of the benefit is obtained after 10 iterations.

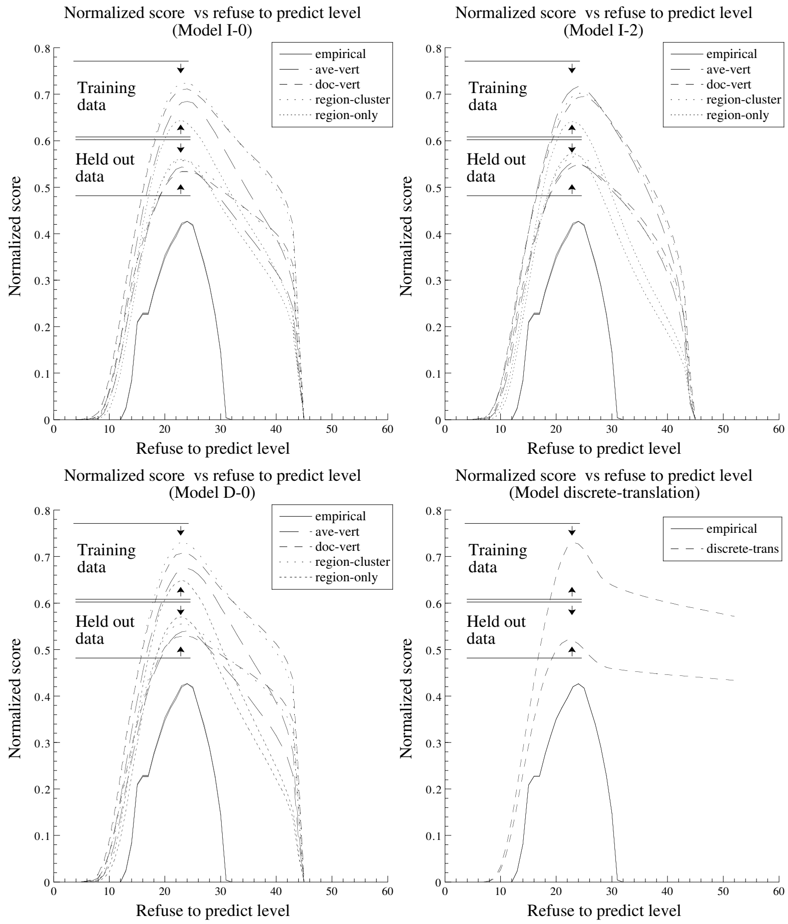

Figure 4: Normalized word prediction performance versus refuse to predict level for 4 model. When a single value is required (as in Table 4) we use a refuse to predict level for which the empirical word distribution gives the maximum (x=25). The results for the training and held out sets when using the empirical distribution are very close and the curves are essentially on top of one another. The refuse to predict level is the probability of word emission which decreases exponentially from left to right ($p = 10^{-(x/10)}$), where x is the "level" recorded on the x axis). As x increases (and p decreases), the number of words predicted increases, and, performance first increases, and then decreases. All methods illustrated here perform significantly better than prediction based on training word frequency at all refuse to predict levels.

Method	Training data	Held out data	Novel data
linear-I-0-doc-vert	0.130 (0.003)	0.095 (0.003)	0.057 (0.003)
binary-I-0-ave-vert	0.130 (0.005)	0.082 (0.004)	0.023 (0.005)
binary-I-0-doc-vert	0.152 (0.005)	0.094 (0.004)	0.034 (0.005)
binary-I-0-region-cluster	0.157 (0.005)	0.099 (0.004)	0.038 (0.005)
binary-I-0-region-only	0.140 (0.005)	0.099 (0.003)	0.037 (0.006)
binary-I-2-ave-vert	0.141 (0.005)	0.087 (0.003)	0.023 (0.004)
binary-I-2-doc-vert	0.137 (0.005)	0.090 (0.004)	0.036 (0.004)
binary-I-2-region-cluster	0.141 (0.005)	0.099 (0.004)	0.039 (0.004)
binary-I-2-region-only	0.133 (0.005)	0.104 (0.004)	0.040 (0.005)
linear-D-0-doc-vert	0.147 (0.002)	0.102 (0.002)	0.059 (0.004)
binary-D-0-ave-vert	0.126 (0.005)	0.081 (0.003)	0.024 (0.005)
binary-D-0-doc-vert	0.160 (0.005)	0.094 (0.003)	0.037 (0.005)
binary-D-0-region-cluster	0.166 (0.005)	0.100 (0.003)	0.040 (0.005)
binary-D-0-region-only	0.143 (0.005)	0.103 (0.003)	0.038 (0.005)
binary-D-2-ave-vert	0.143 (0.005)	0.088 (0.003)	0.021 (0.005)
binary-D-2-doc-vert	0.173 (0.005)	0.102 (0.003)	0.036 (0.005)
binary-D-2-region-cluster	0.177 (0.005)	0.108 (0.003)	0.039 (0.005)
binary-D-2-region-only	0.152 (0.005)	0.108 (0.003)	0.036 (0.005)
linear-C-0-region-only	0.103 (0.003)	0.067 (0.002)	0.035 (0.005)
binary-C-0-ave-vert	0.115 (0.004)	0.070 (0.003)	0.015 (0.005)
binary-C-0-doc-vert	0.137 (0.003)	0.078 (0.002)	0.025 (0.004)
binary-C-0-region-cluster	0.142 (0.003)	0.085 (0.002)	0.030 (0.005)
binary-C-0-region-only	0.128 (0.003)	0.085 (0.003)	0.032 (0.005)
discrete-translation	0.129 (0.004)	0.073 (0.003)	0.029 (0.005)
MoM-LDA	0.053 (0.002)	0.050 (0.002)	0.038 (0.002)

Table 1: Image annotation performance for some of the methods developed in the text. Methods I-0 and I2 use hierarchical clustering models with cluster conditional independence, D-0 and D-2 increases the dependence of word emission on blobs, C-0 integrates strict correspondence, discrete-translation is the discrete translation method, and MoM-LDA is latent Dirichlet allocation with 50 mixture components and 10 factors. The values are the increase in the annotation word list prediction score (measure PR) over that computed using the empirical word distribution (about 0.19). There are on average about 3 words to predict, so a value of 0.1 (good result on held out data) corresponds to predicting about 0.9 of them, as opposed to 0.6 with the empirical distribution. Predicting words for images from the novel CD's is very difficult, but all methods consistently do a little better than the empirical distribution on this task. Errors (shown in parentheses) were estimated from the variance of the word prediction process over 10 different test sets, with at least 1000 samples in each set being averaged for the result for each set.

In general, the range of results support the key notion that word prediction is facilitated by honoring the compositional nature of images and associated text. That is, words are generally

Method	Training data	Held out data	Novel data
linear-I-0-doc-vert	0.301 (0.005)	0.174 (0.007)	0.081 (0.007)
binary-I-0-ave-vert	0.294 (0.006)	0.154 (0.006)	0.064 (0.008)
binary-I-0-doc-vert	0.325 (0.006)	0.160 (0.007)	0.065 (0.008)
binary-I-0-region-cluster	0.332 (0.006)	0.168 (0.007)	0.068 (0.008)
binary-I-0-region-only	0.234 (0.006)	0.160 (0.006)	0.062 (0.008)
binary-I-2-ave-vert	0.331 (0.006)	0.164 (0.008)	0.068 (0.007)
binary-I-2-doc-vert	0.322 (0.006)	0.170 (0.008)	0.074 (0.008)
binary-I-2-region-cluster	0.324 (0.006)	0.179 (0.008)	0.076 (0.008)
binary-I-2-region-only	0.228 (0.006)	0.163 (0.006)	0.068 (0.007)
linear-D-0-doc-vert	0.321 (0.005)	0.167 (0.006)	0.076 (0.008)
binary-D-0-ave-vert	0.284 (0.007)	0.151 (0.007)	0.061 (0.008)
binary-D-0-doc-vert	0.321 (0.007)	0.157 (0.007)	0.064 (0.008)
binary-D-0-region-cluster	0.330 (0.006)	0.166 (0.008)	0.067 (0.008)
binary-D-0-region-only	0.239 (0.006)	0.162 (0.007)	0.064 (0.007)
binary-D-2-ave-vert	0.312 (0.005)	0.162 (0.003)	0.066 (0.005)
binary-D-2-doc-vert	0.358 (0.005)	0.172 (0.003)	0.069 (0.005)
binary-D-2-region-cluster	0.360 (0.005)	0.179 (0.003)	0.072 (0.005)
binary-D-2-region-only	0.248 (0.005)	0.167 (0.003)	0.066 (0.005)
linear-C-0-region-only	0.240 (0.005)	0.124 (0.007)	0.046 (0.006)
binary-C-0-ave-vert	0.252 (0.006)	0.143 (0.007)	0.060 (0.008)
binary-C-0-doc-vert	0.281 (0.006)	0.148 (0.006)	0.054 (0.007)
binary-C-0-region-cluster	0.290 (0.006)	0.157 (0.007)	0.064 (0.007)
binary-C-0-region-only	0.233 (0.006)	0.163 (0.006)	0.071 (0.006)
discrete-translation	0.318 (0.005)	0.111 (0.007)	0.016 (0.008)
MoM-LDA	0.125 (0.005)	0.107 (0.005)	0.041 (0.007)

Table 2: Image annotation performance for some of the methods developed in the text. The values are the increase in the normalized classification score (measure NS) over that computed using the empirical word distribution (about 0.425). The refuse to predict level corresponds very roughly to predicting 80 percent of the words; the increase of 0.160 (typical for our held out data results) corresponds to reaching this level with about 40 guesses as compared with roughly 70 for the empirical distribution (vocabulary size is around 155, depending on the test set). See the caption for Table 1 for additional details.

associated with **pieces** of images, not the entire image. Building representations for those pieces which incorporate both word and region features is a much cleaner approach to word prediction than attempting to represent all images. (The set of objects is much smaller than the set of all common arrangements of objects).

When we look at the effect of topology, one important observation can be made: Methods which use image clustering are very reliant on having images which are close to the training data. As discussed above, we included clustering in some of the models to exploit context. However, our results indicate that forcing images into clusters is too strong a condition for doing so. Although

Method	Training data	Held out data	Novel data
linear-I-0-doc-vert	1.235 (0.02)	0.688 (0.02)	0.258 (0.01)
binary-I-0-ave-vert	1.210 (0.03)	0.563 (0.02)	0.060 (0.01)
binary-I-0-doc-vert	1.385 (0.02)	0.587 (0.02)	0.061 (0.02)
binary-I-0-region-cluster	1.429 (0.03)	0.651 (0.02)	0.094 (0.02)
binary-I-0-region-only	1.061 (0.02)	0.684 (0.02)	0.160 (0.02)
binary-I-2-ave-vert	1.367 (0.03)	0.608 (0.02)	0.084 (0.01)
binary-I-2-doc-vert	1.320 (0.03)	0.627 (0.02)	0.129 (0.01)
binary-I-2-region-cluster	1.342 (0.03)	0.694 (0.02)	0.156 (0.01)
binary-I-2-region-only	1.016 (0.02)	0.709 (0.02)	0.211 (0.01)
linear-D-0-doc-vert	1.376 (0.02)	0.714 (0.02)	0.268 (0.01)
binary-D-0-ave-vert	1.169 (0.03)	0.550 (0.02)	0.057 (0.01)
binary-D-0-doc-vert	1.417 (0.03)	0.601 (0.02)	0.074 (0.01)
binary-D-0-region-cluster	1.466 (0.03)	0.669 (0.02)	0.105 (0.02)
binary-D-0-region-only	1.086 (0.02)	0.700 (0.02)	0.175 (0.02)
binary-D-2-ave-vert	1.310 (0.005)	0.627 (0.003)	0.089 (0.005)
binary-D-2-doc-vert	1.589 (0.005)	0.674 (0.003)	0.102 (0.005)
binary-D-2-region-cluster	1.613 (0.005)	0.739 (0.003)	0.132 (0.005)
binary-D-2-region-only	1.155 (0.005)	0.747 (0.003)	0.180 (0.005)
linear-C-0-region-only	0.980 (0.02)	0.472 (0.02)	0.106 (0.01)
binary-C-0-ave-vert	1.020 (0.02)	0.516 (0.02)	0.071 (0.01)
binary-C-0-doc-vert	1.205 (0.02)	0.541 (0.02)	0.042 (0.01)
binary-C-0-region-cluster	1.254 (0.02)	0.601 (0.02)	0.104 (0.01)
binary-C-0-region-only	1.015 (0.02)	0.643 (0.02)	0.179 (0.01)
discrete-translation	1.347 (0.02)	0.433 (0.002)	-0.072 (0.01)
MoM-LDA	0.452 (0.01)	0.401 (0.01)	0.171 (0.01)

Table 3: Image annotation performance for some of the methods developed in the text as measured by the reduction of the KL divergence from that computed using the empirical distribution (roughly 4.8). We use these numbers largely for comparison—an intuitive absolute scale is not readily available. See the caption for Table 1 for additional details.

clustering improved the training set results, it slightly degraded performance on the held out images from the same CD's. For the novel CD's, clustering significantly reduces performance. This is quite understandable in light of the discussion in the previous paragraph, but the degree of degradation—especially in the case of held out images from the same CD's as training—was unexpected.

The performance of the discrete-translation was worse than that for the most similar non-discrete model (linear-D-0-doc). This is consistent with our belief that it is better to *simultaneously* learn the models for the blobs and their linkage to words. The performance on the three different data sets indicates that there may be overfitting problems as well. This in turn may be due to the fact that errors due to early quantization are artifacts of the training data.

The results for the MoM-LDA model are worth noting. While the performance on the training data is worse than for the other models, the annotation results are only slightly poorer than the other models when the test data resembles the training data, and are competitive with the other models when the test data are novel. Thus MoM-LDA shows a strong resistance to overfitting. Moreover, given that the MoM-LDA model that we used has far fewer clusters than the I0 and I1 models (MoM-LDA has 50 mixture components and 10 LDA factors, versus 256 mixture components and 9 aspect model factors for I0 and I1), further study of larger-scale MoM-LDA models is warranted.

Exploiting context is important to remove ambiguities but our results indicate that we need to explore more subtle approaches for doing so. Of course, clustering is warranted for many applications such as browsing and search where characterization of the training set is important. However, for recognition, something else is needed to deal with ambiguity. If we think of recognition as identifying the sky in an image of a jet, having been exposed to sky only in jungle scenes, then it is clear that trying to put the jet image into an inappropriate cluster should yield poor results.

7.2 Correspondence Results

Figure 6 shows region annotations for a few sample images. For this result we labeled each region with the maximal probability word, using model C-2. In Table 4 we provide quantitative correspondence results computed over 50 images from each of the 10 held out sets. Results for each of the three error measures is provided. For region based word prediction, it is perhaps most reasonable to predict only a few words for each region. This process is most closely studied with the simple keyword prediction error, $E_{PR}^{(model)} - E_{PR}^{(empirical)}$. Here the results suggest that the methods which have been developed to learn correspondence do in fact do better at this task, relative to the performance on the annotation proxy. For, example, using the PR measure, linear-C-0-region-only scores 0.067 with the annotation proxy, which is significantly exceeded by the performance of linear-I-0-region-cluster (same as linear-I-0-doc-vert) which scores 0.094. Using the correspondence measure, they are comparable.

Although the paired word-blob emission approach had the intended effect of improving correspondence performance over annotation performance, we are disappointed that its correspondence performance is still matched by several methods, and significantly bettered by at least one of them. We expect that we need to integrate NULL's properly into this approach. There are two possible benefits: First, the model should no longer be compelled to predict words that it cannot predict and second, the joint probability table may be fitted more accurately because the fitting process should be protected from a large number of outliers caused by forcing each region to correspond to some word. Currently, for both correspondence and annotation, linear-D-0-region-only (same as linear-D-0-doc-vert), appears to be the best overall choice, taking all measures and data sets into account.

8. Discussion

We have compared a variety of methods for predicting words from pictures. Each of these methods can predict some words rather well, and some can predict correspondence well for some words, too. There are practical applications for such methods. Furthermore, they offer an intriguing way to think about object recognition. A great deal remains to be done.

A large variety of other models and fitting methods appear natural. For example, one might model the conditional distribution of image features, given a word, as a Gaussian; there would be one such Gaussian per word. Fitting a model of this form presents some practical difficulties, but

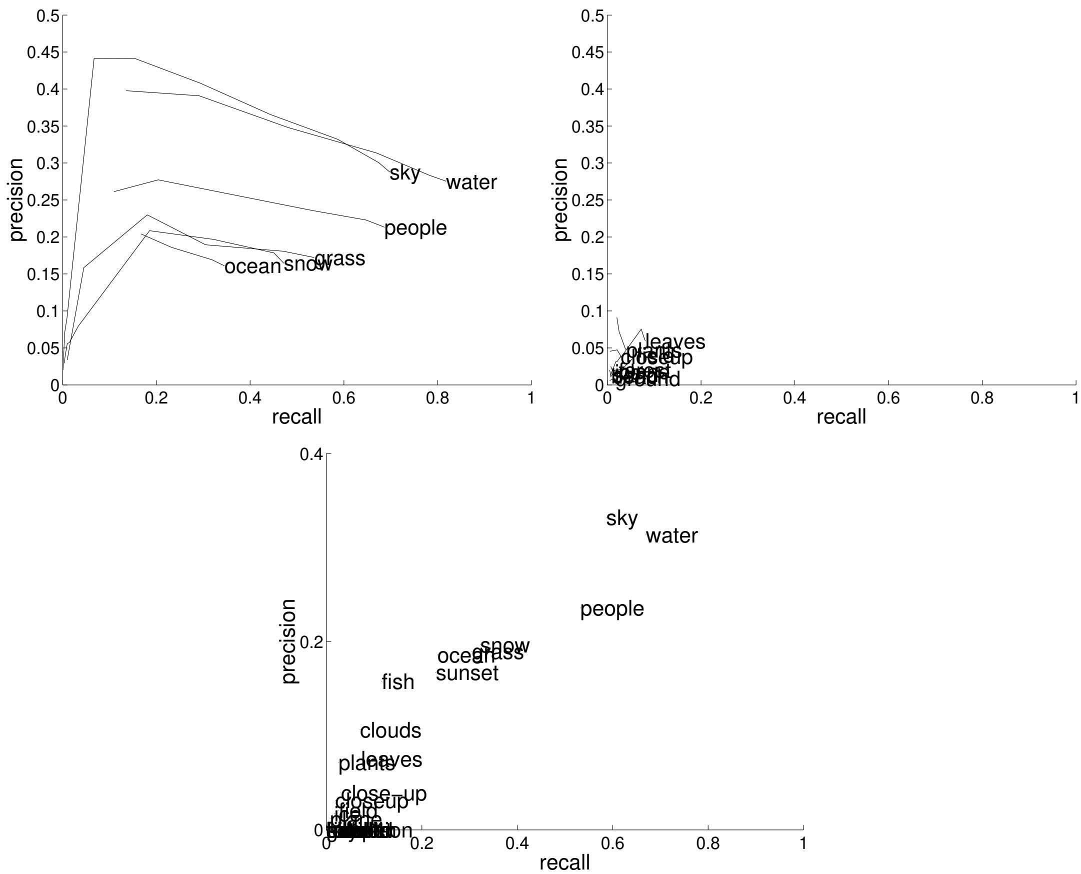

Figure 5: Specific word precision and recall using the discrete translation model. For each word, the task was to predict it for the images where it was a keyword, and not to predict it otherwise. A word is predicted if it is the maximal word predicted by any of the blobs in the image. Precision is the total number of correct predictions over all images, divided by the number of predictions made (duplicate predictions count as a single prediction). Recall is the total number of correct predictions divided by the number of occurrences as a keyword. Results are the average over the 10 held out data sets. In (a) and (b) the relationship between precision and recall is modulated by the refuse to predict level which changes along the curves. As the level increases, fewer words are predicted, and recall goes down, but the words which are predicted are predicted with more certainty, and precision goes up. In (a) we show the results for some words with good performance, and (b) we show some words with poor performance. Because the scales are the same, the curves for the poorly predicted words are all located in the bottom left corner. In (c) the refuse to predict level is fixed, but we show the performance for the words as a scatter plot. These figures show that we do quite well on a modest set of words, and that performance on the rest is limited.

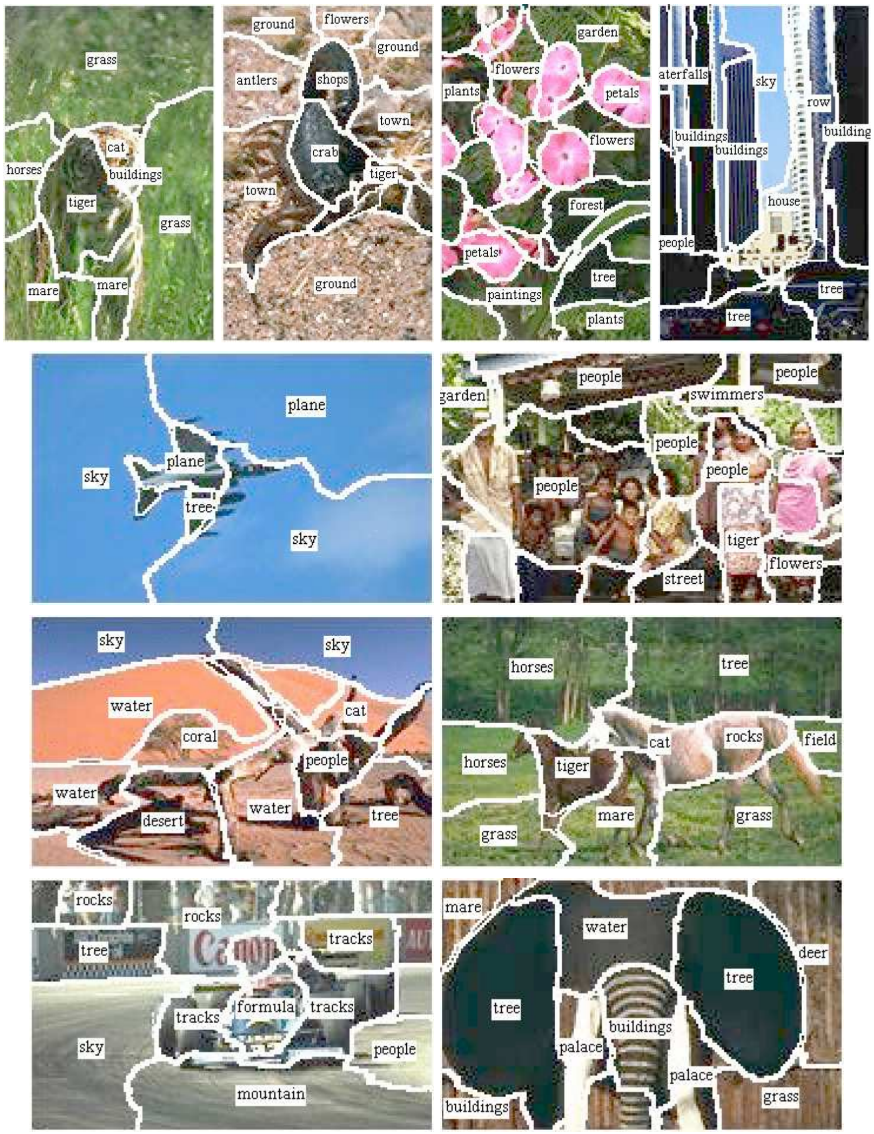

Figure 6: Examples of region based annotation using C-2-pair-only on held out data. The first two rows are good results. The left image on row 3 has some good labels, but the three water labels are likely due more to that word being common in training than the region features. The next two images have lots of correct words for the image (good annotation), but most words are not on the right region (poor correspondence). Specifically, on the car image the tires are labeled “tracks,” which belongs elsewhere. On the horse image neither “horse” nor “mares” is in the right place. The last bottom right image is an example is complete failure.

Method	PR measure
linear-I-0-region-only	0.099 (0.02)
binary-I-0-region-cluster	0.101 (0.01)
binary-I-0-region-only	0.103 (0.01)
binary-I-2-region-cluster	0.101 (0.01)
binary-I-2-region-only	0.093 (0.01)
linear-D-0-region-only	0.132 (0.01)
binary-D-0-region-cluster	0.096 (0.01)
binary-D-0-region-only	0.104 (0.01)
binary-D-2-region-cluster	0.103 (0.01)
binary-D-2-region-only	0.092 (0.01)
linear-C-0-region-only	0.101 (0.01)
discrete-translation	0.066 (0.01)

Table 4: Correspondence performance as measured over 10 sets of 50 manually annotated images from the held out set using the PR measure. All values are relative to the performance using the empirical distribution (about 0.094). For this task, the PR is arguably the most indicative measure as it corresponds to forcing each region to only emit a small number of words (the number of alternative labels). The NS measure is not appropriate because the refuse to predict level was calibrated under different conditions. Note that for comparison with the annotation results, linear-I-0-region-only and linear-I-0-doc-vert give the same results, as do linear-D-0-doc-vert and linear-D-0-region-only.

our initial experiments suggest it might be worthwhile. One might attempt to use model selection methods of one form or another to suggest synonymous words—the Corel data set contains both "train" and "locomotive"—or words effectively synonymous given our features—"jet," "plane" and "bird," say. One might also use model selection methods to search for appropriate feature sets. Note that there is no particular reason that features need to be independent of identity; an improved model would predict some bits of a word's index using a fixed feature set, and then predict other bits using features conditioned by the first bits.

The Corel data set is relatively simple because the annotations are nouns selected from a relatively small vocabulary. Many data sets contain free text annotations. It is a simple matter to tag all nouns, throw away all other text, and regard the result as an annotation. Much more might be possible; for example, one might wish to do some natural language processing to identify candidate annotations that appear to refer to the picture.

Performance is almost certainly affected by the correspondence model used; currently, we require that each region generate a word and that all words be accounted for. One might require one-one correspondence between words and a subset of regions, or insert a NULL as above. This slightly modifies the formulation of the EM fitting methods.

We currently have little information about the effect of supervision, but we expect that quite small supervisory input might lead to significant changes in the model. This is because missing correspondence information can generate symmetries in the incomplete data log-likelihood. For example, if "tiger" and "grass" always appear together, there is no way to determine which is which; but annotating a small number of images will break this symmetry, and could cause a substantial change in the model. A reasonable measure of performance of a model (and an associated fitting algorithm) is the quantity of supervisory input required to achieve a particular level of performance on some reference collection.

Large scale evaluation of correspondence models is genuinely difficult. The problem is important. In the not-too-distant future, there will be recognition systems that can manage vocabularies that are large enough that manual checking of labeled images is an unsatisfactory test. How can one tell how well such a system works? Our current strategy is to investigate methods that obtain extrapolated estimates of correspondence performance from proxies applied to test sets with carefully chosen properties. The key issue seems to be the entropy of the labels; if it is hard to predict the second word from the first word for each data item in the test collection, then annotation performance is likely to predict correspondence performance.

Acknowledgments

This project is part of the Digital Libraries Initiative sponsored by NSF and many others. Specifically, this material is based upon work supported by the National Science Foundation under Grant No. IIS 9817353. Kobus Barnard and Nando de Freitas also receive funding from NSERC (Canada), and Pinar Duygulu is funded by TUBITAK (Turkey). David Blei is funded by a fellowship from the Microsoft Corporation. We are grateful to Jitendra Malik and Doron Tal for normalized cuts software, and Robert Wilensky for helpful conversations.

References

L. H. Armitage and P. G. B. Enser. Analysis of user need in image archives. *Journal of Information Science*, 23(4):287–299, 1997.

K. Barnard, P. Duygulu, and D. A. Forsyth. Clustering art. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages II:434–441, Hawaii, 2001.

K. Barnard and D. A. Forsyth. Learning the semantics of words and pictures. In *International Conference on Computer Vision*, pages II:408–415, 2001.

D. Blei and M. Jordan. Modeling annotated data. Technical Report CSD-02-1202, U.C. Berkeley CS Division, 2002.

D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. In *Advances in Neural Information Processing Systems 14*, 2002.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The mathematics of machine translation: Parameter estimation. *Computational Linguistics*, 19(10):263–311, 1993.

C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002.

G. Celeux, D. Chauveau, and J. Diebolt. On stochastic versions of the EM algorithm. Technical report 2514, INRIA, March 1995.

F. Chen, U. Gargi, L. Niles, and H. Schtze. Multi-modal browsing of images in web documents. In *SPIE Document Recognition and Retrieval*, 1999.

J. Chen, C. A. Bouman, and J. C. Dalton. Hierarchical browsing and search of large image databases. *IEEE Transactions on Image Processing*, 9(3):442–455, 2000.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39:1–38, 1977.

P. Duygulu, Kobus B., J. F. G de Freitas, and D. A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *The Seventh European Conference on Computer Vision*, pages IV:97–112, 2002.

P. G. B. Enser. Query analysis in a visual information retrieval context. *Journal of Document and Text Management*, 1(1):25–39, 1993.

P. G. B. Enser. Progress in documentation pictorial information retrieval. *Journal of Documentation*, 51(2):126–170, 1995.

M. M. Fleck, D. A. Forsyth, and C. Bregler. Finding naked people. In Bernard Buxton and Roberto Cipolla, editors, *4th European Conference on Computer Vision*, pages II:591–602. Springer, 1996.

D. A. Forsyth. Computer vision tools for finding images and video sequences. *Library Trends*, 48 (2):326–355, 1999.

D. A. Forsyth and J. Ponce. *Computer Vision - A Modern Approach*. Prentice-Hall, 2002.

C. O. Frost, B. Taylor, A. Noakes, S. Markel, D. Torres, and K. M. Drabenstott. Browse and search patterns in a digital image database. *Information retrieval*, 1:287–313, 2000.

T. Hofmann. Learning and representing topic. A hierarchical mixture model for word occurrence in document databases. In *Workshop on learning from text and the web*, CMU, 1998.

T. Hofmann and J. Puzicha. Statistical models for co-occurrence data. A.I. Memo 1635, Massachusetts Institute of Technology, 1998.

R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38:325–340, 1987.

D. Jurafsky and J.H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice-Hall, 2000.

L.H. Keister. User types and queries: impact on image access systems. In *Challenges in indexing electronic text and images*. Learned Information, 1994.

M. La Cascia, S. Sethi, and S. Sclaroff. Combining textual and visual cues for content-based image retrieval on the world wide web. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1998.

C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press. Cambridge, MA, 1999.

M. Markkula and E. Sormunen. End-user searching challenges indexing practices in the digital newspaper photo archive. *Information retrieval*, 1:259–285, 2000.

O. Maron. *Learning from Ambiguity*. Ph.D. dissertation, Massachusetts Institute of Technology, 1998.

O. Maron and A.L. Ratan. Multiple-instance learning for natural scene classification. In *The Fifteenth International Conference on Machine Learning*, 1998.

D. Melamed. *Empirical methods for exploiting parallel texts*. MIT Press, Cambridge, Massachusetts, 2001.

Y. Mori, H. Takahashi, and R. Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *First International Workshop on Multimedia Intelligent Storage and Retrieval Management (in conjunction with ACM Multimedia Conference 1999)*, Orlando, Florida, 1999.

M. Oren, C. Papageorgiou, P. Sinha, and E. Osuna. Pedestrian detection using wavelet templates. In *Computer vision and pattern recognition*, pages 193–9, 1997.

S. Ornager. View a picture: Theoretical image analysis and empirical user studies on indexing and retrieval. *Swedis Library Research*, 2(3):31–41, 1996.

Shin'ichi Satoh and T. Kanade. Name-it: Association of face and name in video. In *Proceedings of 1997 IEEE Computer Vision and Pattern Recognition (CVPR '97)*, pages 368–373, June 1997.

H. Schneiderman and T. Kanade. A statistical approach to 3d object recognition applied to faces and cars. In *IEEE Conference on Computer Vision and Pattern Recognition*, page 100. IEEE, 2000.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):888–905, 2000.

R. K. Srihari. *Extracting Visual Information from Text: Using Captions to Label Human Faces in Newspaper Photographs*. Ph.d. thesis, SUNY at Buffalo, 1991.

R. K. Srihari and D. T. Burhans. Visual semantics: Extracting visual information from text accompanying pictures. In *AAAI '94*, Seattle, WA, 1994.

R. K. Srihari, R. Chopra, D. Burhans, M. Venkataraman, and V. Govindaraju. Use of collateral text in image interpretation. In *ARPA Image Understanding Workshop*, Monterey, CA, 1994.

M. J. Swain, C. Frankel, and V. Athitsos. Webseer: An image search engine for the world wide web. Technical Report TR-96-14, Computer Science Department, University of Chicago, 1996.