Object Recognition as Machine Translation – Part 2: Exploiting Image Database Clustering Models

Kobus Barnard[1], Pinar Duygulu[1], Nando de Freitas[2], and David Forsyth[1]

[1] Computer Science Division, University of California at Berkeley, Berkeley, CA, USA
{kobus, duygulu, daf}@cs.berkeley.edu
[2] Computing Science, University of British Columbia, Vancouver. BC, Canada
nando@cs.ubc.ca

Abstract. We treat object recognition as a process of attaching words to images and image regions. To accomplish this we exploit clustering methods which learn the joint statistics of words and image regions. We show how these models can then be used to attach words to images outside the training set. This "auto-annotation" process has applications such as image indexing, as well as being related to object recognition. Predicted words can be compared to actual words associated with images in a held out set, and we introduce several performance measures based on this observation. These measures are then used to make principled comparisons of model variants, and proposed enhancements.

Word prediction is most simply done as a function of the entire image. However, for recognition we need to learn the correspondence between words and specific image regions. Here we first show that the existing models can be used for this purpose, and then we propose modifications to improve performance based on this goal. Finally, we propose word prediction performance as a segmentation measure and report the results for two segmentation approaches.

Keywords. Object recognition, segmentation, correspondence

1 Introduction

We treat object recognition as a process of attaching words to images—translating visual representations into language. This indirect approach has several advantages. First, it is concrete and testable. Second, it is general—we do not need to specify in advance which objects or scene semantics are to be considered. Third, it can exploit large existing data sets. These include indexed image databases (Corel, online museum data, stock photo collections), web images with captions or other associated text, and video data (together with speech recognition). Finally, because the approach is general but testable, it can be used to develop vision tools which are more applicable to the general recognition problem. As an example of this, we propose word prediction performance as a segmentation measure, and use it to show a significant difference between two popular state of the art segmentation approaches.

To attach words to pictures we exploit clustering methods which learn the joint statistics of image words and segments. Barnard et al [1, 2] have introduced models for these statistics, and applied them to traditional image data base applications such as browsing and search, as well as the novel application of attaching words to pictures ("auto-annotate"), which we study in depth here. As suggested in those studies, and replicated below, these methods can produce words that are clearly associated with images outside the training set. Being able to provide image keywords automatically is very useful—most image datasets are accessed via keywords [3-6]. Furthermore, the predicted words are indicative of scene semantics. Due to the strong connection between the predicted words' meanings and the scene context, this process has clear ties to recognition.

We measure performance by comparing predicted words to words which are associated with images outside the training set. We use such measures to make principled comparisons of model variants. Going further, we propose using word prediction to evaluate image segmentations in such a way that is relevant for recognition. One could easily evaluate other basic computer vision processes such as feature selection. The key point is that processes which support discovering semantics, rather than low level tasks (e.g. edge finders for stereo matching [7, 8]), can be identified and improved.

To further develop the ties to recognition we investigate learning the **correspondence** between the predicted words and **particular** image regions. In a companion paper [9] we consider in detail a direct approach for learning this correspondence. Here we develop a different group of approaches which build on existing co-occurrence models. First we show that these models can be used for region based annotation. Then we propose modifications to enhance the performance on this task. We find that doing so increases performance on both the scene-based and the region-based annotation tasks.

We mention a few other approaches to inferring semantics from image features. Campbell et al [10, 11] learn to classify regions based on labeled region data. Wang et al [12] learn features for a small number of pre-specified categories which are then applied to find images likely to also be of that category. Maron [13] uses Multiple Instance Learning to learn connections between image features and concepts from sets of positive and negative examples. Finally, the literature on object recognition based on specific, object dependent, signatures learned from labeled training data is vast—see [14] for an overview.

2 Scene based word prediction (auto-annotation)

The models introduced in [1, 2] (based on ones developed for text [15, 16]) can be used to generate appropriate words for an image ("auto-annotation"). Image items (words and regions) are assumed to be generated by a statistical process, with words and regions are considered analogously. Image items are generated from nodes which are arranged in a tree structure. Clusters are associated with paths from leaf nodes to the root. 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

$$P(D \mid d) = \sum_c P(c) \prod_{i \in D} \left(\sum_l P(i \mid l,c) P(l \mid d) \right) \quad (1)$$

where c indexes clusters, i indexes items (words or image segments), and l indexes levels. D is the set of observations for the document (words and image regions). For $P(i \mid l,c)$ we use frequency tables in the case of words and Gaussian distribution with diagonal covariance over the features for the regions. $P(l \mid d)$ 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 and Forsyth [1, 2], we also experiment with allowing a cluster dependent level structure. Here $P(l \mid d)$ is replaced with $P(l \mid c,d)$ (Refered to as 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 $P(l \mid d)$ or $P(l \mid c,d)$ which are specific to the documents in the training set. This makes the model powerful for search applications, but considering a document not in the training set requires either estimating the vertical mixing weights or marginalizing out the training data as done in [17]. To estimate the mixing weights we can use a cluster specific average computed during training, or re-fit the model with the document under consideration.

Preliminary results indicate that for some applications the simple strategy of using the cluster specific average for the vertical weights works well, indicating that the selection of vertical nodes used to model a document is of the primary importance, and that further modeling of the distribution over the levels is not crucial. This lead to the proposal in [1] of a truly generative alternative model (Model I-2 in the results):

$$P(D) = \sum_c P(c) \prod_{i \in D} \left(\sum_l P(i \mid l,c) P(l \mid c) \right) \quad (2)$$

Word Prediction. To predict words from images we assume that we have a new document with a set of observed image segments, S. We wish to compute $P(w \mid S) \propto P(w,S)$ for each word, w, in our vocabulary. Using (1) we get:

$$\begin{aligned} P(w \mid S) &\propto \sum_c P(c) P(w \mid c) P(S \mid c) \\ &= \sum_c P(c) \left(\sum_l P(w \mid c,l) P(l \mid c) \right) \prod_{s \in S} \left(\sum_l P(s \mid l,c) P(l) \right) \end{aligned} \quad (3)$$

Here we have dropped the document index, d, from the vertical weights $P(l)$ ($P(l \mid c)$ in I-1 and I-2), as we are normally interested in applying (3) to documents outside the training set. For the vertical weights we either use cluster specific average mixing weights (labeled "ave-vert" in the results), or refit the model based on S (moderately

expensive, “doc-vert”). Refitting based on (S, w) is also possible, but is substantially more expensive. Marginalizing out the training data gives good results on small data sets, especially on training data, but it is very expensive to compute on data of the scale of interest, and thus we do not report results here.

3 Region based word prediction (correspondence)

The models described in §2 implicitly learn some correspondence through co-occurrence because there is a fitting 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 be generated by a shared node, as there are far fewer nodes than observations. Thus we can ask how well the nodes bind words and regions by the following simple word prediction method for a single region:

$$P(w \mid s) \propto \sum_{c} P(c) \left(\sum_{l} P(s \mid l,c) P(w \mid c,l) P(l \mid c) \right) \tag{4}$$

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|S). We label these strategies as pair-only and pair-cluster in the results.

Notice that using (4) is not quite the same as replacing the set of regions, S, in (3) with the single region of interest, s. 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 applicable to that cluster. Thus we are using the model differently than it was trained. Nonetheless, as demonstrated below, it is possible to extract correspondence using this method.

Integrating correspondence. The fact that we had to abuse the original models to consider correspondence strongly suggests that we can do substantially better by integrating correspondence into the model and learning it during training. To do so, we assume that the observed words and regions are emitted in pairs so that $D = \{(w,s)_i\}$ and (1) becomes:

$$P(D \mid d) = \sum_{c} P(c) \prod_{(w,s) \in D} \left(\sum_{l} P((w,s) \mid l,c) P(l \mid d) \right) \tag{5}$$

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.

Binding the word and segment emission at the nodes means that a query on a segment leads to a node which has learned about co-occurring words. In the previous models a query on a segment produces a likely cluster, but the predicted words have more freedom to be modeled by other nodes used by that cluster. The cluster conditional independent emission of words and regions in the original model is therefore somewhat counter to identifying correspondence. The new model introduces stronger ties between the emitted words and regions at the expense of a more complicated learning algorithm which we discuss next.

Equation (5) 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 it as part of the training process. Further, since there are too many possibilities to sum over, we sample likely correspondences using estimates of the probabilities that each word is emitted with each segment. We have also experimented with a greedy assignment similar to that used for machine translation by Melamed [18], but preliminary results suggest that sampling works better. 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.

We impose the constraint that the same correspondence applies regardless of the proposed cluster membership. Further we 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. This final simplified probability is easy to express given our models. Using $w \Leftrightarrow s$ to denote that the word, w, and the region, s, correspond, we use, in the case of C-0:

$$P(w \Leftrightarrow s) \approx \sum_{c} P(c) \sum_{l} P((w,s) \mid l,c) P(l \mid d) \qquad (6)$$

To produce a proposed correspondence we consider each segment in turn and choose words using (6) restricted to the words not yet accounted for, or without restriction once all words have been paired. Once all words have been paired, we terminate the process once the next multiple of the number of segments in the image has been reached. Thus we assume that the generating process emits some high probability duplicates to maintain the criterion that segments and words are emitted in pairs.

4 Measuring Performance

Performance measurement is a critical part of our approach. Because the task is difficult we need to be able to objectively distinguish between results which are far below human ability. Making gains requires observing small differences. An important part of this work is therefore to demonstrate careful but stable performance measurement.

To measure performance we look at word prediction on held out data. Since the held out data also has associated text, we can measure performance by comparing the predicted words with the actual words. Not all words appropriate to the image are included with the data, but this is not a significant problem when the purpose is simply to compare methods. The strategy has the advantage that performance comparisons can be carried out automatically and therefore on a substantial scale. To consider how well the models are doing in a more absolute sense, we compare their performance against word prediction based on the frequency of the words in the training set (word prior).

We report two performance measures. First, as the models produce posterior distributions for word occurrence, we report the KL divergence between the computed posterior distribution, P(w|S), and the target distribution. Unfortunately, the target distribu-

tion is not known, and for this we simply assume that the actual words should be predicted uniformly, and that all others words should not be predicted at all.

The second error measure quantifies how well the systems perform on the named task—specifically emitting words. An ideal measure would be based on an appropriate loss function, itself provided by a specific application. One difficulty in specifying a general purpose loss function is the observation that certain errors ("cat" for "tiger") are less critical than others ("car" for "vegetable"). Without an appropriate loss function, we are left with simply counting the words. However, since the task is difficult, and the number of inappropriate words far exceeds the number of appropriate ones, simply subtracting the number of incorrect predictions from the number of correct ones is too harsh—clearly predicting five good words in ten tries should give a score greater than zero. 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 $r/n - w/(N - n)$ 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.

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 1, 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.

Measuring region oriented word prediction performance is more difficult than the straight annotation because we do not have the correspondence information. We can cautiously use the annotation task as a proxy, because performance on the two tasks should be strongly correlated. We report results using the image based word prediction methods (ave-vert, doc-vert) as well as summing over the words emitted by the regions (pair-cluster, pair-only). Since the correspondence model was trained on the assumption that every region emits a word, we normalize the probabilities before forming the sum. Increased performance on these region based methods relative to the image based ones is suggestive of correspondence learning.

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. Here we looked at each region, and counted the number of times the word with the highest probability for that region was reasonable as an index term, was relevant to the region, and had a plausible visual connection to it. Thus the word "ocean" for "coral" would be judged incorrectly because the ocean is transparent. When the regions crossed natural boundaries we judged the word correct if it applied to more than half the region. 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 when it appears, but we scored it as incorrect because it is not suggestive of recognition. See [9] for more on vocabulary issues.

5 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 test (25%) sets. Each such sample was given to each process under consideration, and the results of 5 of such samples were averaged. This controls for both the input data and EM initialization. For comparison with the method developed in the companion paper [9], we used the same data in that work, which was a similar, but different sample from the same set of CD’s. Images were segmented using N-Cuts [19, 20]. We used a modest selection of features for each segment, including size, position, color, oriented energy (12 filters), and a few simple shape features. Except where noted, the tree topologies were binary trees with 9 levels (511 nodes).

We computed the performance for the six models (I-0, I-1, I-2, C-0, C-1, C-2) using the four word prediction strategies (ave-vert, doc-vert, pair-cluster, and pair-only). Results using the normalized classification score are reported in Table 1, and results using KL divergence are in Table 2. We also show how the normalized classification score changes as a function of the refuse to predict level (Figure 1).

The results indicate that learning correspondence is helpful for the annotation task, especially when measured using the normalized classification score. Thus doing so should be applicable even to tasks which do not require correspondence, such as automatically generating indexing keywords. Although the annotation scores do not directly measure correspondence, we expect that the performance is increased by doing so. This is corroborated by the increase in performance of the region based annotation methods (pair-cluster, pair-only) relative to the image based methods.

5.1 Comparing with direct correspondence learning

In Table 3 we compare the methods developed here with translation approach for learning correspondence described in a companion paper [9]. The results using KL divergence are comparable. In the case of the normalized classification measure, the translation method does not do well, but further investigation reveals that the choice of setting the refuse to predict level by the performance of the prior on training data hurts this method relative to the others, at least on this data set. The best refuse to predict level for the translation method is somewhat less than that for the other methods whose optimums are either flatter or closer to that for the prior. Setting the refuse to predict level for each method based on training data should help performance in general, and reduce this particular discrepancy.

5.2 Recognition Performance

To truly test how well we learn the correspondence we need to visually inspect the results, as correspondence information is not available for our data set (nor any other large, general, image data set). We subjectively scored 100 held out images similar to those in Figure 2 as described in Section §4. Each method was asked to predict one word per region. The C-1-pair-cluster method predicted an appropriate word 15.5% of the time, the C-1-pair-only method predicted an appropriate word 17.3% of the time, and the translation method predicted an appropriate word 15.0% of the time. Not enough people have scored enough images to have a good estimate of the error. We currently regard the numbers as roughly comparable. For further comparison we scored the results using I-1-pair-only. Again, this region based annotation uses Model 1 differently than how it was trained. The result here was 14.2%. Despite the preliminary nature of these results, we are encouraged that they are completely consistent with our other measures, and intuitively make sense. For example, training with correspondence improved performance from 14.2% to 17.3%. Note that for this task, unlike image annotation, intelligent guessing using the prior is completely hopeless. The strategy here would be to attach the most common word ("water") to each segment. The score obtained doing is 5.7%.

6 Word prediction as a segmentation measure

The machinery developed above for testing word prediction can be applied to an open problem in computer vision, namely testing segmentation performance. Recently, Martin et. al. have considered comparing segmentations to those provided by human subjects [21]. It is generally understood that that segmentation metrics should be task oriented, but a good task has not been forthcoming, or the experiment has been left undone (but see [22] for related work). We argue that word prediction is an excellent task because it is associated with higher level image semantics and recognition.

As an example of applying this strategy we used word prediction to compare N-cuts segmentation to the EM based segmentation method used in Blobworld [23, 24]. One possible confound which needs to be considered is that the number of segments, which is a function of the segmentation method and its parameters (considered fixed for this example), can affect the word prediction process. We generally restrict the number of segments used, excluding the smaller ones. Therefore we look at the performance of the two segmentation methods as a function of the number of segments chosen for word prediction. We used the same 5 data sets as in the previous experiments. The images used, and the features computed from the segments was the same for the two cases in the five separate runs. Figure 3 shows typical N-cuts and Blobworld segmentations of sample images.

Figure 4 shows the performance of the two segmentation approaches using the KL error measure. The results using the normalized word classification score are similar. The words were predicted according to Model I-1-ave-vert, using a binary tree with 9 levels. We ran the same experiment with a 5 level quad tree with similar results.

The results using N-cuts are significantly better relative to the estimated error on all three data sets—training, held out, and novel CD's. This was somewhat expected, as the N-cuts segments visually better follow the boundaries of semantically salient regions, but as can be seen in Figure 3, it is not completely obvious that these segmentations are more suitable for our task. Thus it is significant that we can separate the performance of the two methods well beyond estimated error.

7 Conclusion

Translating image regions into words is an attractive alternative view of object recognition. The approach is both general and testable. Traditional approaches to recognition require large, labeled data-sets (often helped with black cloth backgrounds). However, pseudo labeled images are everywhere—we just don't know what the labels should be attached to! This observation has lead us to focus on the correspondence problem.

There is much left to do. For example, we are currently using a very modest feature set. In future work we will use the measurement techniques developed here to select additional features to improve the system. Going even further, we will use region based word prediction posterior probabilities to help propose region merges and thereby integrate our high level approach with the lower level segmentation process. Effectively proposing high level region merges—such as joining the black and white halves of a penguin-is beyond the capabilities of current vision systems, but our approach provides a way to address these challenges.

8 Acknowledgements

This project is part of the Digital Libraries Initiative sponsored by NSF and many others. Kobus Barnard also receives funding from NSERC (Canada), and Pinar Duygulu is funded by TUBITAK (Turkey). We are gratef ul to Jitendra Malik and Doron Tal for normalized cuts software, and Robert Wilensky for helpful conversations.

9 References

1. K. Barnard, P. Duygulu, and D. Forsyth, "Clustering Art," Proc. Conference on Computer Vision and Pattern Recognition, Hawaii, pp. II:434-441, 2001.
2. K. Barnard and D. Forsyth, "Learning the Semantics of Words and Pictures," Proc. International Conference on Computer Vision, pp. II:408-415, 2001.
3. P. G. B. Enser, "Progress in documentation pictorial information retrieval," Journal of Documentation, vol. 51, pp. 126-170, 1995.
4. P. G. B. Enser, "Query analysis in a visual information retrieval context," Journal of Document and Text Management, vol. 1, pp. 25-39, 1993.

5. M. Markkula and E. Sormunen, "End-user searching challenges indexing practices in the digital newspaper photo archive," Information retrieval, vol. 1, pp. 259-285, 2000.
6. S. Ornager, "View a picture. Theoretical image analysis and empirical user studies on indexing and retrieval," Swedis Library Research, vol. 2, pp. 31-41, 1996.
7. M. M. Fleck, "Multiple Widths Yield Reliable Finite Differences," IEEE Trans. Patt. Analy. Mach. Intell., vol. 14, pp. 412-429, 1992.
8. M. M. Fleck, "A Topological Stereo Matcher," Intern. Journ. Comp. Vision, vol. 6, pp. 197-226., 1991.
9. P. Duygulu, K. Barnard, J. F. G. d. Freitas, and D. A. Forsyth, "Object recognition as machine translation-I: Learning a lexicon for a fixed image vocabulary" Submitted for publication to the seventh European Conference on Computer Vision.
10. N. W. Campbell, B. T. Thomas, and T. Troscianko, "Automatic segmentation and classification of outdoor images using neural networks.," International Journal of Neural Systems, vol. 8, pp. 137-144, 1997.
11. N. W. Campbell, W. P. J. Mackeown, B. T. Thomas, and T. Troscianko, "Interpreting image databases by region classification," Pattern Recognition, vol. 30, pp. 555-563, 1997.
12. J. Z. Wang, J. Li, and G. Wiederhold, "SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries," IEEE Trans. Patt. Anal. Mach. Intell., vol. 23, pp. 947-963, 2001.
13. O. Maron, Learning from Ambiguity, MIT Ph.D., 1998.
14. D. A. Forsyth and J. Ponce, Computer Vision - A Modern Approach, in press.
15. T. Hofmann and J. Puzicha, "Statistical models for co-occurrence data," Massachusetts Institute of Technology, A.I. Memo 1635, 1998,
16. T. Hofmann, "Learning and representing topic. A hierarchical mixture model for word occurrence in document databases," Proc. Workshop on learning from text and the web, CMU, 1998.
17. D. Blei, A. Ng, and M. Jordan, "Dirichlet Allocation Models," Proc. NIPS, 2001.
18. D. Melamed, Empirical methods for exploiting parallel texts. Cambridge, Massachusetts: MIT Press, 2001.
19. J. Shi and J. Malik., "Normalized Cuts and Image Segmentation," IEEE Trans. Patt. Anal. Mach. Intell., vol. 22, pp. 888-905, 2000.
20. Normalized Cuts Software, Available from http://dlp.CS.Berkeley.EDU/~doron/software/ncuts/.
21. D. Martin, C. Fowlkes, D. Tai, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," Proc. International Conference on Computer Vision, pp. II:416-421, 2001.
22. S. Borra and S. Sarkar, "A Framework for Performance Characterization of Intermediate Level Grouping Modules," IEEE Trans. Patt. Anal. Mach. Intell., vol. 19, pp. 1306-1312, 1997.
23. S. Belongie, C. Carson, H. Greenspan, and J. Malik, "Color-and Texture-based Image Segmentation Using the Expectation-Maximization Algorithm and Its Application to Content-Based Image Retrieval," International Conference on Computer Vision, 1998.
24. C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image segmentation using Expectation-Maximization and its application to image querying" Submitted for publication to IEEE Trans. Patt. Anal. Mach. Intell., available in the interim from http://HTTP.CS.Berkeley.EDU/~carson/papers/pami.html.

Table 1. Image annotation performance for the methods developed in the text. The values are the increase in the normalized classification score over that computed using the prior (about 4.25). For the refuse to predict level used, a value of 0.1 corresponds to predicting 1-2 more words accurately than using the prior. The results confirm that when the original models are used differently than intended to extract correspondence (pair-cluster and pair-only methods with I-0, I-1, I-2) image based word prediction performance on held out data drops significantly. However, when correspondence is added to the training process word prediction performance is significantly improved, based on comparing the maxium scores for each model over the four word prediction methods (ave-vert, doc-vert, pair-cluster, and pair-only). The results also suggest that there is a slight advantage for C-1 over C-0 and C-2, and I-2 over I-1 and I-0, but further work is required to demonstrate this with certainty. The prediction task on Novel CD's is very difficult, and some methods show negative results (worse than prior). However, it is notable that the best methods consistently do a little better than the prior on this task. Errors (shown in parentheses) were estimated from the variance of the word prediction process over 500 different images over 5 input sets.

Method	Training data	Held out data	Novel CD's
I-0-ave-vert	0.254 (0.008)	0.120 (0.007)	0.025 (0.007)
I-0-doc-vert	0.244 (0.009)	0.044 (0.008)	-0.068 (0.008)
I-0-pair-cluster	0.257 (0.009)	0.072 (0.008)	-0.048 (0.008)
I-0-pair-only	0.177 (0.008)	0.093 (0.008)	-0.035 (0.008)
I-1-ave-vert	0.282 (0.007)	0.121 (0.007)	0.020 (0.007)
I-1-doc-vert	0.296 (0.008)	0.052 (0.008)	-0.054 (0.008)
I-1-pair-cluster	0.296 (0.008)	0.073 (0.008)	-0.038 (0.008)
I-1-pair-only	0.186 (0.008)	0.092 (0.008)	-0.038 (0.008)
I-2-ave-vert	0.321 (0.007)	0.123 (0.007)	0.021 (0.007)
I-2-doc-vert	0.234 (0.008)	0.068 (0.008)	-0.030 (0.007)
I-2-pair-cluster	0.246 (0.008)	0.092 (0.008)	-0.015 (0.008)
I-2-pair-only	0.165 (0.008)	0.097 (0.008)	-0.017 (0.008)
C-0-ave-vert	0.222 (0.008)	0.111 (0.007)	0.009 (0.007)
C-0-doc-vert	0.234 (0.008)	0.107 (0.007)	0.001 (0.007)
C-0-pair-cluster	0.261 (0.008)	0.126 (0.007)	0.012 (0.007)
C-0-pair-only	0.216 (0.007)	0.142 (0.007)	0.035 (0.007)
C-1-ave-vert	0.248 (0.008)	0.112 (0.007)	0.016 (0.007)
C-1-doc-vert	0.272 (0.008)	0.110 (0.007)	0.000 (0.007)
C-1-pair-cluster	0.288 (0.008)	0.134 (0.007)	0.017 (0.007)
C-1-pair-only	0.229 (0.007)	0.152 (0.006)	0.042 (0.007)
C-2-ave-vert	0.258 (0.008)	0.123 (0.007)	0.006 (0.007)
C-2-doc-vert	0.267 (0.008)	0.109 (0.007)	-0.008 (0.007)
C-2-pair-cluster	0.286 (0.007)	0.138 (0.007)	0.011 (0.007)
C-2-pair-only	0.229 (0.00)7	0.145 (0.00)7	0.028 (0.007)

Table 2. Performance of the methods developed in the text. The values are the reduction of the KL divergence from that computed using the prior (roughly 5.0). We use these numbers largely for comparison—an intuitive absolute scale is not reaadily available. The results confirm that learning correspondence is generally helpful for image based word prediction on held out data in the case of Model 0 and Model 1 (comparing I-0 to I-1 and C-0 to C-1), but the advantage is generally smaller than that measured with the normalized classification score, and counter to the trend Model I-2 is better than C-2 with this measure. Also, with this measure, "pair-cluster" is better than "pair-only", which is the reverse of the result found using the normalized classification score (Table 1).

Method	Training data	Held out data	Novel CD's
I-0-ave-vert	0.850 (0.04)	0.509 (0.023)	0.212 (0.016)
I-0-doc-vert	0.916 (0.04)	0.383 (0.025)	0.083 (0.02)
I-0-pair-cluster	0.947 (0.04)	0.442 (0.025)	0.114 (0.02)
I-0-pair-only	0.646 (0.03)	0.403 (0.025)	0.038 (0.017)
I-1-ave-vert	0.923 (0.04)	0.502 (0.025)	0.212 (0.015)
I-1-doc-vert	1.061 (0.04)	0.410 (0.025)	0.101 (0.02)
I-1-pair-cluster	1.053 (0.04)	0.463 (0.025)	0.128 (0.02)
I-1-pair-only	0.697 (0.03)	0.418 (0.025)	0.047 (0.017)
I-2-ave-vert	1.086 (0.04)	0.560 (0.025)	0.248 (0.015)
I-2-doc-vert	0.902 (0.04)	0.450 (0.025)	0.149 (0.020)
I-2-pair-cluster	0.915 (0.04)	0.494 (0.025)	0.172 (0.02)
I-2-pair-only	0.568 (0.03)	0.357 (0.022)	0.021 (0.017)
C-0-ave-vert	0.797 (0.04)	0.467 (0.025)	0.141 (0.016)
C-0-doc-vert	0.890 (0.04)	0.449 (0.025)	0.121 (0.017)
C-0-pair-cluster	0.953 (0.04)	0.516 (0.025)	0.175 (0.017)
C-0-pair-only	0.772 (0.03)	0.514 (0.023)	0.214 (0.015)
C-1-ave-vert	0.876 (0.04)	0.505 (0.025)	0.143 (0.016)
C-1-doc-vert	1.016 (0.04)	0.486 (0.025)	0.110 (0.017)
C-1-pair-cluster	1.050 (0.04)	0.568 (0.027)	0.157 (0.017)
C-1-pair-only	0.807 (0.03)	0.556 (0.025)	0.206 (0.015)
C-2-ave-vert	0.960 (0.04)	0.519 (0.025)	0.153 (0.016)
C-2-doc-vert	1.043 (0.04)	0.476 (0.025)	0.106 (0.017)
C-2-pair-cluster	1.086 (0.04)	0.560 (0.025)	0.165 (0.017)
C-2-pair-only	0.823 (0.03)	0.540 (0.025)	0.197 (0.015)

Table 3. Comparison of image annotation performance of two methods developed in this paper with the alternative approach (translation) proposed in [9].

Method	Normalized classification score, less that for the prior		KL divergence subtracted from that for the prior	
	Training data	Held out data	Training data	Held out data
C-1-pair-cluster	0.271 (0.014)	0.067 (0.010)	0.885 (0.07)	0.457 (0.04)
C-1-pair-only	0.215 (0.014)	0.067 (0.010)	0.678 (0.06)	0.443 (0.04)
Translation	0.304 (0.013)	0.018 (0.011)	0.732 (0.052	0.433 (0.04)

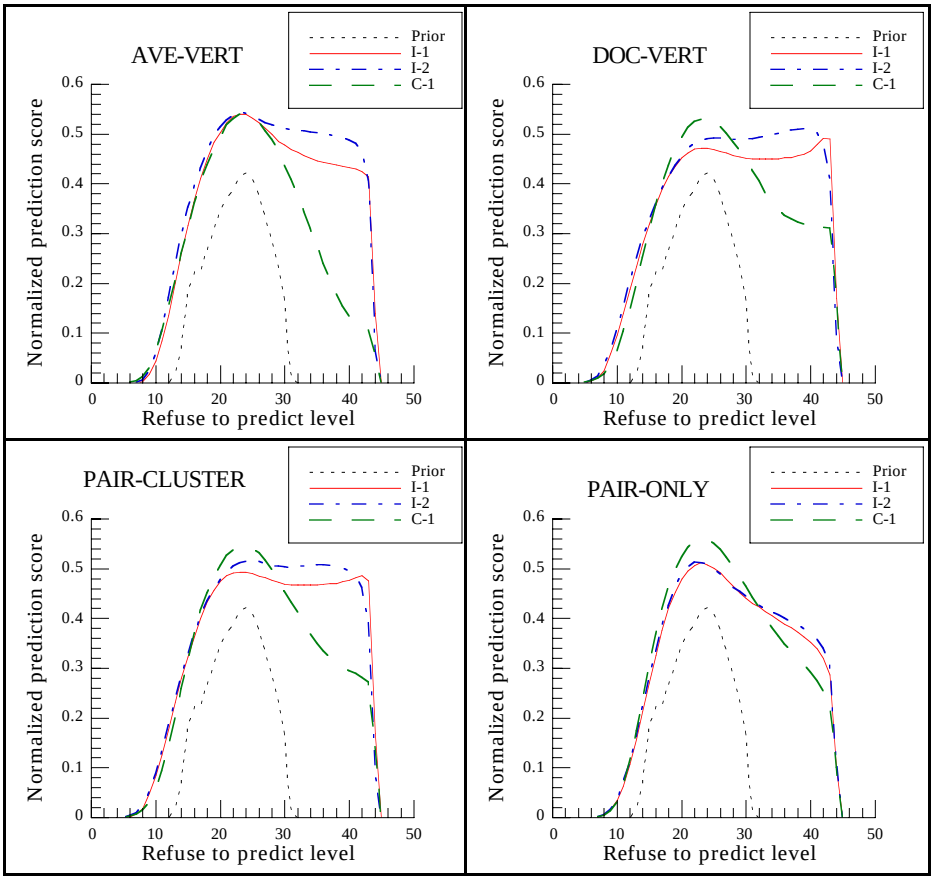

Fig. 1. Performance versus refuse to predict level with the three of the models and the four word prediction strategies. We study curves such as these because word prediction performance can be improved if algorithms can choose when not to predict words. Performance is computed using normalized word prediction counts as described in §4. 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 (or levels off). All methods illustrated here perform significantly better than prediction based on training word frequency (prior) at all refuse to predict levels. Notice that incorporating correspondence in training does not diminish peak performance when used with the basic word prediction strategy (ave-vert), but that segment oriented word emission performance is significantly increased (pair-cluster, pair-only). This suggests that some correspondence has been learned.

Fig 2. Examples of region based annotation using C-1-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, but correspondence is not good. 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 example is complete failure. The subjective scores for these images are roughly 0.4, 0.4, 0.4, 0.5 (row 1), 0.6, 0.3 (row 2), 0.4, 0.3 (row 3) 0.0, 0.0 (row 4). Average score for 100 images is 0.17.

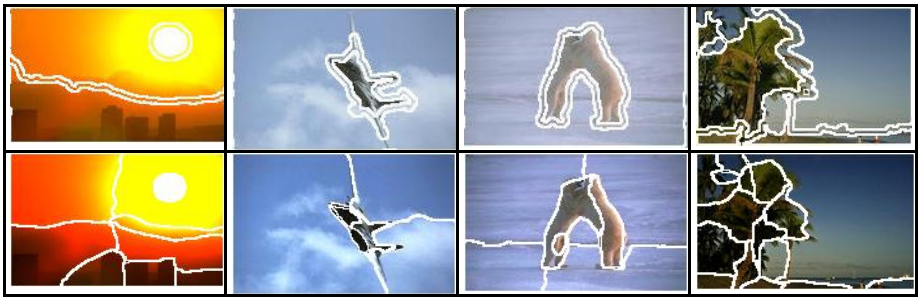

Fig 3. Examples of Blobworld segments (top) and N-Cuts segments (bottom).

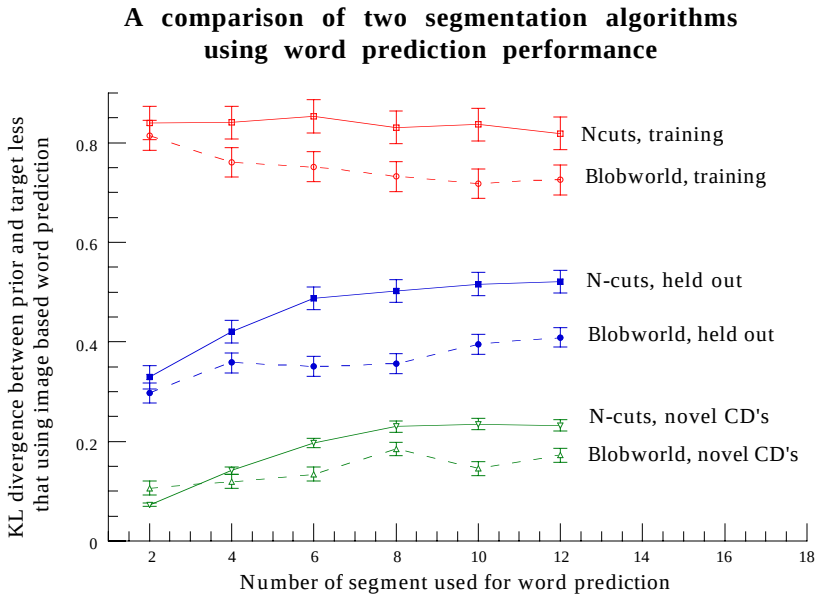

Fig 4. Segmentation methods compared using word prediction performance. Performance is measured by the KL divergence between the target distribution and the posterior word probability using Model I, subtracted from the value obtained using the training set word frequencies (prior). All values plotted are positive, which means that performance always exceeds that using the prior. The two segmentation approaches are shown to be significantly different, given the error estimates indicated by the bars around the points. The errors were estimated from the variance of the word prediction process over 500 different images over 5 input sets. The model topology was a binary tree with 9 levels.