Evaluation of localized semantics: Data, methodology, and experiments

KOBUS BARNARD, QUANFU FAN, RANJINI SWAMINATHAN, ANTHONY HOOGS,
RODERIC COLLINS, PASCALE RONDOT, JOHN KAUFHOLD

Computer Science Department, The University of Arizona

GE Global Research, One Research Circle, Schenectady, New York, 12309

{kobus, quanfu, ranjini}@cs.arizona.edu

hoogs@crd.ge.com, collins@research.ge.com

University of Arizona, Computing Science,
Technical Report, TR-05-08,
September 12, 2005

Abstract. We present a new data set encoding localized semantics for 1014 images and a methodology for using this kind of data for recognition evaluation. This methodology establishes protocols for mapping algorithm specific localization (e.g., segmentations) to our data, handling synonyms, scoring matches at different levels of specificity, dealing with vocabularies with sense ambiguity (the usual case), and handling ground truth regions with multiple labels. Given these protocols, we develop two evaluation approaches. The first measures the range of semantics that an algorithm can recognize, and the second measures the frequency that an algorithm recognizes semantics correctly. The data, the image labeling tool, and programs implementing our evaluation strategy are all available on-line (kobus.ca//research/data/IJCV).

We apply this infrastructure to evaluate four algorithms which learn to label image regions from weakly labeled data. The algorithms tested include two variants of multiple instance learning, and two generative multi-modal mixture models. These experiments are on a significantly larger scale than previously reported, especially in the case of the multiple instance learning. More specifically, we used training data sets up to 37,000 images and training vocabularies of up to 650 words.

We found that image level word prediction, which is a cheaper evaluation alternative, does not correlate well with region labeling performance, thus validating the need for region level analysis. We also found that for the measures sensitive to occurrence statistics, we needed to provide the multiple instance learning methods with an appropriate prior for good performance. With that modification used when appropriate, we found that the EMDD multiple instance learning method gave the best overall performance over three tasks, with one of the generative multi-mixture models giving the best performance on one of them.

1 Introduction

Demonstrating recognition requires specifying where an entity is, in addition to whether or not it is present. Intuitively, if a program "recognizes" a horse in an image, but attaches the location of the horse to the grass that the horse is standing on (Figure 1), then true recognition performance is limited, and certainly does not match that of a program that can specify where the horse is. Thus to properly evaluate recognition approaches, we need to consider localization. Our goal with this research is to provide publicly available infrastructure to automate the evaluation of the localization of image semantics for widespread domains. Since specifying detailed localized image semantics is a time consuming task, it should be done so that it serves diverse evaluation endeavors, thereby maximizing the gain for the effort. This means that the image semantics need to be characterized independently of proposed algorithms. Our approach is to provide general purpose tools that can then map results from specific experiments into

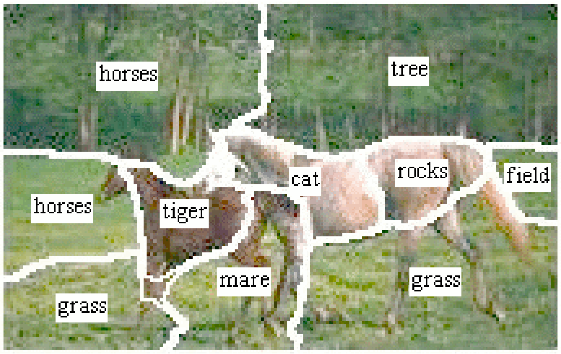


Figure 1. An example of region labeling that gives relatively good annotation results despite several obvious incorrect correspondences. Both "horses" and "mares" are good words for the image, but neither are correctly placed. In the training data used for this experiments, horses are and grass of the above color and texture co-occur often, and are only rarely a part, making it difficult to learn the difference. Higher throughput localization performance measurement will help characterize our systems.

characterized image semantics.

To do this we begin with human image segmentations available from a separate research project (Martin et al., 2001) which provides the localization of image semantics. To label the entities we use the WordNet (Fellbaum et al., ; Miller et al., 1990) system for maximal accuracy and flexibility. As described below, this data set can then be adapted to the evaluation of algorithms using different localizations and vocabularies. In particular, algorithm localization can be translated into ground truth localization, and algorithm vocabularies can be mapped into a semantic space. This is important because we wish to support a wide range of vocabularies, from object categories to free form text associated with images gathered from the web.

Given these protocols, we propose two evaluation strategies. The first measures the range of semantics that an algorithm can recognize, and the second measures the frequency that an algorithm recognizes semantics correctly. The first considers the semantic entities as equal, and measures the performance on each one over the data set. It is thus effectively a count over the entities in the semantic space. It is immune to how common the entity is. The second takes the opposite approach, effectively counting performance over recognition attempts. It thus rewards good performance on common entities. Published results tend to be along the lines of the second strategy (or under conditions that the two approaches are similar), but the two approaches provide different useful information. In experiments which simply provide a category label for each image, then both approaches can be merged by using semantically well spaced categories, with an equal number of images for each category. However, if the task is to understand the entire image, then an image collection and a single measure cannot easily be devised that gives results for both approaches simultaneously, and it is simpler to report results for both.

1.1 Benefits of evaluating semantics with localization

We reiterate that measuring recognition sensibly requires that the evaluation considers localization. Pragmatically, we need to characterize localization performance in order to understand when performance is the consequence of the visual characteristics of relevant semantic entities, and when it is due to correlations with other entities. If we can separate the effect of these two sources of information, then we can better integrate them, and design methods that exploit context, but are not overly fooled when it is not informative. Further, while we expect that the performance of algorithms that make excessive use of the background will decrease rapidly as testing conditions depart from training conditions, characterizing localization performance exposes this more explicitly and effectively. We claim this because understanding the degree to which new testing data is different seems a difficult task in itself which may in fact be helped by our data.

Localization performance helps characterize correspondence ambiguity in methods that attempt to learn from loosely labeled data. All algorithms evaluated as part of this work fall into this genre. These methods attempt to learn how to identify regions in images based on labels which apply to the image as a whole. Given a single image with multiple labels and regions, it is clearly not possible to resolve the ambiguity as to which region(s) should be linked to which word(s). Region-word consistency over multiple images, can, however, be used to resolve the ambiguity, provided that there is sufficient variety in the images and the labels. However, if words always co-occur, or effectively co-occur in patterns that are hard to characterize, then the ambiguity cannot be resolved (see Figure 1). It is thus important to be able to measure the reduction in correspondence ambiguity possible by various approaches.

Semantically segmented and labeled data is also a source of high quality training data for learning oriented algorithms. In addition to supporting standard learning approaches to vision, the such data is especially useful in understanding how a small amount of supervisory data can help augment loosely labeled data which is available in large quantities.

1.2 Related work

An early region level semantically labeled image data set is the Sowerby image data base[1] (Vivarelli and Williams, 1997). Here roughly 200 images relevant to driving on British roadways were hand segmented, and given one of 54 labels from a hierarchy. This data is still useful, despite the limited domain.

More recently, the development of algorithms that use large scale weakly labeled data has produced a need for evaluating region performance. In this domain, images are assumed to have associated text such as keywords, but the part of the image that the words refer to is not known. Thus the goal of localizing the semantics is indirectly related to what is available in the data, and measuring performance directly requires region level evaluation. Due to the effort involved in gathering such data, the bulk of the evaluation of these methods has been on the proximal measure provided by the weak labels which measure how well region understanding supports predicting words relevant to the image as whole (image-annotation). Despite the hazards of this approach, it has the advantage of supporting large scale evaluation (Barnard et al., 2003b; Barnard and Forsyth, 2001). Nonetheless, results of region labeling for a number of algorithms for 500 hand labeled regions are available (Barnard et al., 2003a). Unfortunately, this data is specific to both the segmentations and the vocabulary used in the experiments. Additional region labeling results were reported in (Carbonetto et al., 2004).

Recent progress in recognizing object categories (see for example: Agarwal et al., 2004; Berg et al., 2005; Fei-Fei et al., 2004; Fergus et al., 2003; Torralba et al., 2004; Weber et al., 2000) has prompted the gathering of a variety of image data which have been grouped together for the PASCAL Object Recognition challenge (www.pascal-network.org/challenges/VOC/). The CalTech 101 database (Fei-Fei et al., 2004) provides category data at an image level. Images are assumed to be an object from the category and some background. Despite the lack of localization, the second data set is challenging because of the large number (101) of relatively diverse categories. Data sets which have some localization data include the TU Darmstadt Database (Leibe and Schiele) the UIUC Image Database for Car Detection (Agarwal et al.), the Caltech Database (Fergus and Perona), the TU Graz-02 Database (Opelt and Pinz), and the MIT-CSAIL Database of Objects and Scenes (Torralba et al.).

While the above collections goes some distance in providing the community's need, there are several properties of the data set described here that are not available in the existing data. In particular, all regions of reasonable size are labeled (there is no generic concept of background), object contours are of high quality (many of the existing localizations are of the form of bounding boxes), labels link into a semantic structure (WordNet), and the extent of the semantic space is large (over 1,000 WordNet words).

2 Developing an image data set with localized semantic labels

Our goal is to specify localized image semantics to provide ground truth for a wide variety of evaluation experiments. To have this generality, we first focus on semantically labeling the images independently of any particular experiment. The data can then be automatically distilled for a given experiment.

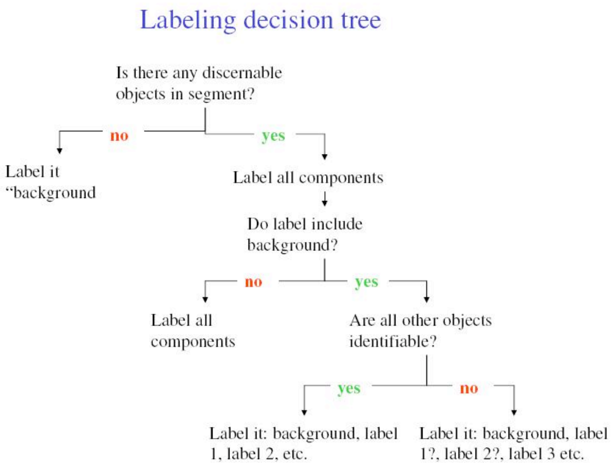
We begin with images segmented into semantically coherent regions. Of course, current segmentation algorithms are not able to deliver accurate, semantically sensitive, segmentations. Thus we exploit the human segmentations for 1014 images produced for a different study (Martin et al., 2001). In that work, multiple segmentations were produced for each image. For this data set we used the segmentation for each image with the median number of segments.

To accurately capture the semantics of each region, human labelers were given the full WordNet (Fellbaum et al., ; Miller et al., 1990) vocabulary. WordNet terms are sense disambiguated, and are organized into semantic hierarchies, which are key to link the labeling to other vocabularies, as described below (§3.4).

2.1 Labeling guidelines

To encourage consistency among labelers, we developed a set of labeling guidelines. We remark that some of the information collected according to these rules (e.g., synonyms) is effectively ignored by the specific processing strategies described below (§3), as the rules were developed in anticipation of additional uses. Our labeling rules are:

1) Words should correspond to their WordNet definition.
2) Words should be lowercase.
3) Words should be singular.
4) The sense in WordNet (if multiple) should be mentioned as word (i), where i is the sense number in Word net except if i=1. (e.g. tiger (2)).
5) Vegetation should be used for any group of plants.
6) Indiscernible objects, which clearly belong to the background, should be labeled background.
7) Add the first synonym given in WordNet as an additional entry. (e.g. building edifice).
8) Words that restrict to a part or class of an object should be word_type, like bobcat_head, the word itself should appear as another entry for that part, (e.g. bobcat bobcat_head). Wherever possible the sense of the word should be incorporated in the part also (e.g. carriage (2) rig (6) carriage (2)_shelter (2)). It is not required to use the additional synonym entry for labeling a part or class.
9) If a word is a compound word and another word can describe it as well add the single word (e.g. “birch tree”, birch).
10) If the same object can be described in a different way, but not necessarily synonym, label as is (e.g. grass, ground).
11) If no objects are discernable in a segment it will be called “background”.
12) If several objects are discernable in a segment then all components will be labeled.
13) If an object is subject to human interpretation, add a question mark at the end of the word (e.g. human?)



Labeling decision tree

Is there any discernable objects in segment?

no

Label it "background

yes

Label all components

Do label include background?

no

Label all components

yes

Are all other objects identifiable?

yes

Label it: background, label 1, label 2, etc.

no

Label it: background, label 1?, label 2?, label 3 etc.

Figure 2. A decision tree that specifies how to handle segments that may include multiple objects and/or background that does not have any identifiable objects in it.

We have developed additional rules specific to humans which are listed in Appendix A. While much of the additional data specific due to humans is not relevant to the experiments presented below, it is very useful to specialized applications. Figure 2 shows a labeling decision tree for the process, and Figure 3 shows two labeling examples.

2.2 Labeling tool and process

We implemented a labeling tool in Java to make the execution of the above approach as efficient as possible. The image being labeled is displayed in a window with the segment being labeled identified by a

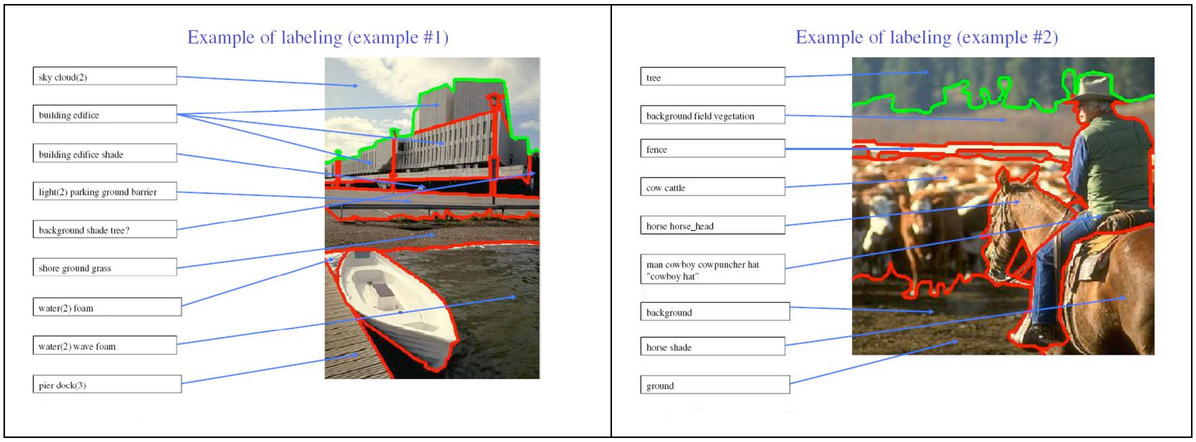


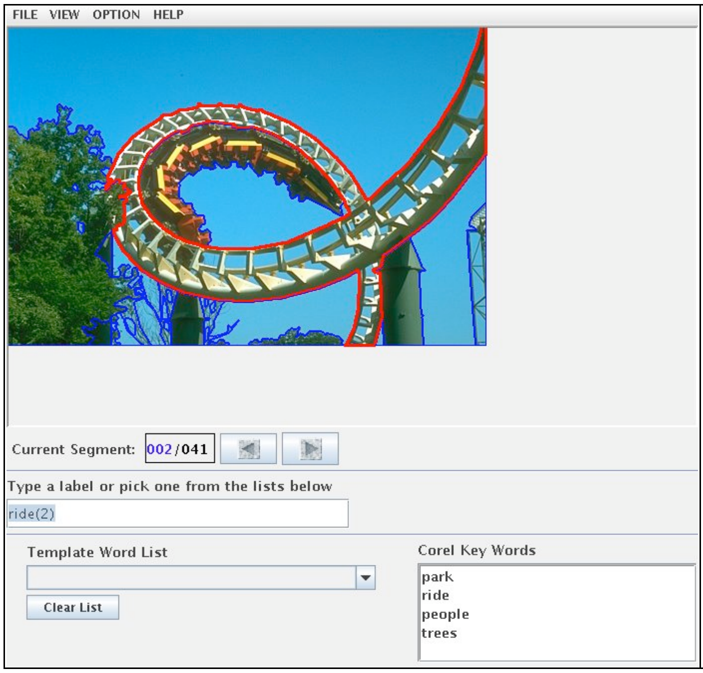
Figure 3. Examples of image labeling that are consistent with the labeling rules described in the text.

red outline (see Figure 4). The human labeler enters appropriate words either by typing them or selecting from lists. During this process, an on-line interface to WordNet is also at hand, and often words are cut and pasted from WordNet. Words that have been used recently, or with chosen similar images, and any available keywords (senses have to be added), are all available for selection. The tool can read previous labelings which is critical for checking and iterative refinement.

For this study four people contributed to the labeling of 1014 images. We estimate that a serviceable preliminary labeling took about 10 minutes per image. After all images were labeled, one labeler went over all images to check for consistency and errors.

3 General purpose evaluation methodology

Our labeling system is designed to very generally capture image semantics with locality, and thus can support a number of different methodologies for the evaluation of inferring semantics from image data. No single evaluation strategy optimally measures all interpretations and applications of this task, and we expect that our data will be used in a variety of ways. However, as an important part of this work we examine the issues in developing an evaluation methodology, and propose strategies consistent with specific preferences regarding what should be measured.



Current Segment: 002/041

Type a label or pick one from the lists below

ride(2)

Corel Key Words

park
ride
people
trees

Figure 4. A screen shot of our labeling tool. We have made the tool available so that others can improve and extend the data set. Features that make the labeling process more efficient include a list of words from a template for labeling similar genres of images, a list of any image key words, and the ability to reload an earlier labeling result and continue previous work, check existing labelings, or improve the labeling.

We find that the most critical preference is whether to focus on the *range* of semantics that can be correctly identified, or, alternatively, how *often* semantics can be correctly identified. A simple example will clarify the difference. Consider two algorithms, one which reliably identifies tigers, but nothing else, and a second one which reliably identifies sky, but nothing else. By the first notion, these two algorithms have the same performance (one semantic entity). By the second notion, the second algorithm performs better because sky is much more common, and thus a count of correctly identified entities over a reasonable test set will be higher. The second algorithm is also likely to be easier to develop because training examples are easier to come by.

In what follows we will develop the first preference, and then modify the method to embody the second preference (§3.5). In the first preference, we would like to reward algorithms which learn a wider range of entities and which excel at learning from fewer examples. We also wish to provide a mechanism for scoring the identification of entities at various levels of generality (say "cat" for "tiger"). Finally, we suggest that labels which are synonyms according to WordNet should be treated identically. Below we develop a method which embodies these choices. In particular we propose a scoring system which measures how well an algorithm identifies entities, giving equal weight to the performance on each. Conveniently, with this approach the performance of any "guessing strategy" is easy to characterize. Because the performance on each entity over the entire data set is weighted equally, there is no advantage to guessing any entity over any other, and all strategies which do not use visual information (e.g. guessing the most common word) will score the same. This is very desirable because it means that there is less error in evaluation due to randomness which will be differently biased for each algorithm.

To implement our scoring strategies, we need to map each algorithm's native semantics (vocabulary) into the ground truth semantic space, and each algorithm's localization into the ground truth segmentation. As an example, consider evaluating the region labeling in Figure 5. The segmentation is not semantically accurate and does not correspond to the one used in the evaluation infrastructure. The words are not necessarily in the ground truth vocabulary, and, unfortunately, are not sense-disambiguated. Ideally, instead of "tiger", we would have tiger(2) indicating the animal meaning of "tiger" in the WordNet system, but computer vision data typically does not have sense specific labels. Further, while

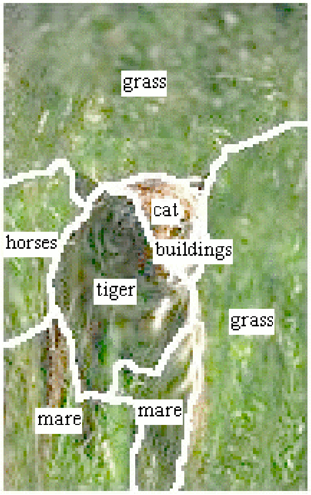


Figure 5. An example of an image labeling effort that must be mapped into the ground truth localization and semantic spaces. The segmentation used by the algorithm breaks up semantic regions leading to the questions such as how the scores for the two regions labeled “grass” should be weighted. Intuitively the region labeled “cat” should receive some score, but not as much as “tiger”. A methodology for dealing with these issues is developed in the text.

some words are either clearly correct (“tiger”) or incorrect (“buildings”), some, like “cat” are in-between, and perhaps should be attributed an in-between score.

3.1 Ground truth vocabulary pre-processing

Due to the general and flexible data collection methodology described above, many ground truth labels have multiple labels. As a first pre-processing step we remove from consideration ones that do not interact with the vocabulary under consideration, and ones that are ancestors of others in the label set. For example, if a ground truth region has both “cat” and “tiger”, then “cat” is dropped. We also remove synonyms, so that there is at most one word from any given WordNet synset (synonym set). Multiple labels are still possible after these pruning processes if there are multiple diverse entities in a region, reflecting a segmentation that was not fine enough. Here we distribute the score available for the region equally among the multiple labels. This reflects the fact that each label of the region will contribute to the scoring, but only one of them is correct for a given location within the region.

3.2 Ground truth region weights

We omit ground truth regions which are too small to be realistically identified. This threshold (1% of image area) is necessarily somewhat arbitrary. If a region has more than one diverse label, we divide the area by the number of such labels before testing against the threshold. A priori, all remaining ground truth regions have equal weight by our main preference. However, we make an adjustment for algorithms which omit parts of the image for various reasons. For example, in the experiments below we only use up to the 16 largest segments from our machine based segmentation. In this case we weight the ground truth segment by the fraction of it that is not excluded. Notice that some ground truth segments may become irrelevant by this process. We then divide each ground truth region by the number of labels as suggested above (§3.1). Finally we normalize the ground truth region weights so that they sum to one for each image. We denote this weight, for a ground truth region G as $w(G)$. The region weight, $w(G)$, is used both in scoring, and in setting the word weights (§3.4). The above computation embodies the choice that all ground truth images contribute equally to the semantic space, regardless of the number of (equally weighted) regions, but has the side effect that similar regions can score differently in different images. The methodology can instead weight regions equally across the entire data set by omitting the final normalization over the region weights for each image.

3.3 Mapping algorithm localization to ground truth

Our approach requires that we compute localized scores with respect to the *ground truth regions*, but every algorithm will, a priori, specify semantics relative to its own localization. This localization may be a classic segmentation, pixel grid blocs, or localized descriptors. In the case of segmentation, we compute the score for each vocabulary word, *v*, for a segment, R, $s_r(v,R)$, as a weighted sum over the scores for v for the labels *l* for the ground truth segments that intersect R. For proper book-keeping in the ground truth semantic space, we set the weights to the fraction of the area of each of the overlapping *ground truth segments*, G. Symbolically, if we denote the score for *v*, against a ground truth label, *l*, by $s_w(v,l)$, and use | | to denote the area of a region, then $s_r(v,R)$, is given by:

$$s_r(v,R) = \sum_G w(G)\left(\sum_{l \in G} s_w(v,l)\right) \cdot \frac{|R \cap G|}{G} \tag{1}$$

The inner sum accounts for the possibility that G has more than one label. Recall that $w(G)$ includes a scale factor so that an average of $s_w(v,l)$ over l is implicit in $w(G)\left(\sum_{l \in G} s_w(v,l)\right)$. Figure 6 illustrates the computation with a simple example.

Notice that to the extent the segmentation is effectively random, larger algorithm segments are weighted more, because on average they occupy more of the semantic entities. In general, this approach

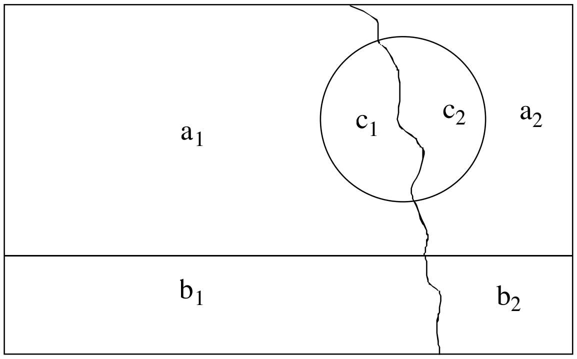


Figure 6. Illustration of the method for weighting the scores for labels from machine segmentations against the ground truth label for a ground truth region. The squiggly line represents an edge from a machine segmentation which divides each of three ground truth regions (a,b, c), with labels (A,B,C) respectively, into two parts. We assume that (A, B, C) is the entire vocabulary, that $s_w(x,y)$ is one if $x=y$ and zero otherwise, that the ground truth region weights, $w(G)$, are all one, and that $|c_1| = |c_2|$. Then both regions score the same for C, but region one scores more for A and B. Because region one has more of the image's semantic space, measured by the effective number of ground truth regions covered, region one scores more in general. Other choices for distributing the ground truth score are discussed in the text.

promotes the notion that scoring should be as independent as possible of the segmentation used by the algorithm. For example, a large sky area should score the same, whether it is properly segmented as one region, or whether it is two regions due to segmentation error, and whose scores are additive.

3.4 Equal semantic scoring with arbitrary vocabularies

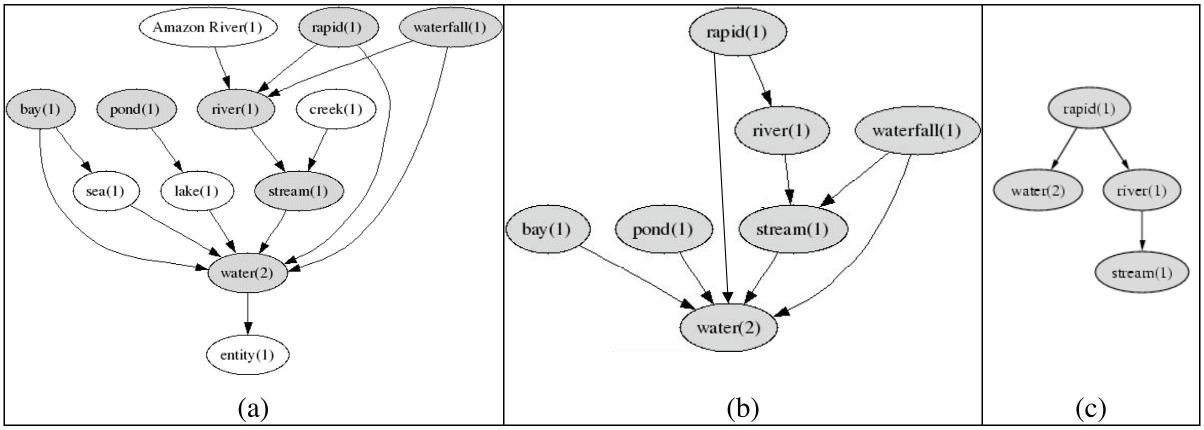
We assume for the moment that the vocabulary words have specified senses, deferring until later the case that the sense is not known (§3.6). We simplify semantic scoring by specifying that synonyms, as given by the WordNet synsets, are interchangeable. We argue that synonyms cannot be visually distinguished, and algorithms and evaluation methodologies should not attempt to distinguish them. The fact that the vocabulary can have differing numbers of words for different concepts suggests that one may want to adjust scores by the number of synonyms that occur in the vocabulary. However, the statistics of the underlying concepts are best represented if all the synonyms are counted together, and thus our default is count them as is, without further normalization.

The next issue is scoring matches with different degrees of specificity. For example, consider a system which does not have "tiger" in its vocabulary, and thus predicts the word "cat" for a region appropriately labeled as "tiger". Clearly some score is warranted. One approach is to make use of measures of semantic similarity developed in the text domain (Banerjee and Pedersen, 2003; Jiang and Conrath, 1998; Leacock and Chodorow, 1998; Lin, 1998; Resnik, 1995), typically based on the paths between the words in WordNet or a notion of information content. Using one of these measures we could establish that "cat" is semantically close to "tiger", and thus using "cat" for "tiger" should score high. Unfortunately, these measures are not designed to reflect the difficulty of any particular word prediction task, and thus are not optimal for evaluation of this task. More importantly, they will not implement uniform weight for each semantic entity. Thus we propose a scoring strategy which is easily configured for this goal. As mentioned above, this is equivalent to making it so that blindly guessing any semantic entity in the ground truth vocabulary will achieve the same score.

3.4.1 Semantic Directed Acyclic Graph

To score matches with different degree of specificity, we first construct a semantic directed acyclic graph (DAG) derived from the WordNet hierarchy. WordNet can be viewed as a directed graph in which nodes are words and links are semantic relations between words. WordNet supports many semantic relations. We used four of them in the *nouns* category: *is-a*, *part-of*, *member-of* and *instance-of*. A word w_j is an *ancestor* of another word w_j if there is one path from w_i to w_j through one of these relations. Let W be the merged vocabulary of the ground truth and the algorithm vocabulary. We construct a DAG by adding edges, $E(i,j)$, from $w_i \in W$ to $w_j \in W$ if w_j is the nearest ancestor of w_i among the words in W according to WordNet. Figure 7-b shows an example of such a DAG with 7 words. Note that for any word $w_i \in W$ in the DAG, we can populate a breadth first search (BFS) tree $T(w_i)$ that is rooted at w_i which encodes the shortest path from w_i to all of its ancestors (see Figure 7-c).

To set the node weights, we construct the BFS tree $T(w_i)$ for each ground truth word, w_i, and then add the weight $w(G)$ (computed in §3.2) for each ground truth region that has word w_i to all nodes in $T(w_i)$. We set the weight for the node to be the reciprocal all the sum of all the weight deposited at that node for all regions for all images. This means that the total amount of weight available to each entity over all the regions is the same. Notice that higher level nodes such as "cat", will receive significant weight due to more specific terms ("tiger", "lion") even if they occur infrequently themselves. Although most learning algorithms would avoid "cat" if it did not occur often in training, we assume that any algorithm can consult WordNet and consider "cat" instead of "tiger". Our approach assures that there is



(a) (b) (c)

Figure 7. The use of WordNet to establish semantic scoring for related words. The WordNet hierarchy can be viewed as a directed graph in which words are nodes and edges represent different semantic relationships between words such as hyponym, holonym, has-member and has-instance. Figure a) shows a subgraph of the hierarchy for nouns. The shaded words are in the ground truth. The number in the parenthesis is the sense of the word set by WordNet. Figure b) shows the directed acyclic graph (DAG) constructed over the ground truth words using the approach described in the paper. Note that a word (e.g. “rapid”) can have more than one path to its in the DAG. Thus for each word we construct a breadth first search (BFS) tree which encodes the shortest path to related words. Figure c) shows the BFS tree populated from the DAG for the “rapid”.

no advantage to doing so without intelligent processing. Finally, we remark that the node weights are slightly dependent on the segmentation if parts of the image were excluded (e.g. perhaps they are considered to be too small) as this can effect $w(G)$. One simplification of the methodology is thus to approximate the node weights for all segmentations assuming that segments are never excluded.

3.4.2 Scoring the vocabulary words

To score the match between a vocabulary word and a ground truth word, we consider the BFS trees of the two words. For a non-zero score, the vocabulary word must be in the BFS tree for the ground truth word, meaning that there must exist a WordNet path from the ground truth word to the WordNet root that includes the vocabulary word. The word specific part of the score, $s_w(v,l)$, is then simply the weight of the most specific common node. Thus if the vocabulary word is “tiger”, and the ground truth word is “cat”, then the score is the value of the node for “cat”. This score needs to be combined with the other factors in (1) in order to yield an overall score for predicting the word with the particular locality.

In the case that the vocabulary word is a child of the ground truth word, or in the case that the words are siblings, we propose that the score should be zero. It might seem reasonable that a child (“tiger” for “cat”) or a sibling (“lion” for “tiger”) should justify some reward. However, the potential gain for being

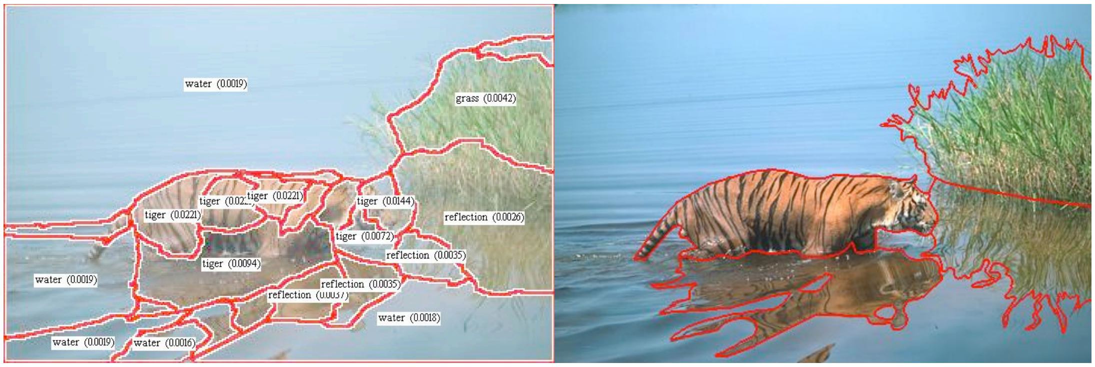


Figure 8. Illustration of mapping of the ground truth data onto experiment specific data. The regions are labeled with the word that would receive maximal score (in parenthesis) under the semantic range scoring method. The machine produced segments (left) are scored as a blend of the underlying ground truth segments (right). The blend is not simple because the vocabulary words need to be connected to one or more ground truth words which may be scored according to how frequent they are (depending on the scoring approach).

correct for “lion”, say, compared with “cat”, has to be offset by the risk of being wrong. If “lion” was rewarded for being a sibling to “tiger”, then it would be better to guess “lion” than “cat” despite a lack of evidence that the cat under consideration is a lion. Thus we propose giving zero score for a semantic sibling. This is consistent with the intuition that being more specific increases the chances that one will be wrong.

The same reasoning applies in the child case, but here we remark that such examples are a symptom that that ground truth data is not specific enough. As the data is refined, and more general labels are replaced by more specific ones, inaccuracies due to this problem will be removed. On the other hand, if the label is general because more specific semantics are not discernable, then a program that attempts to increase its score by “guessing” a more specific label should score less than the program that decides that the more general term is the best that can be done (i.e., agreeing with the human labelers).

3.5 Scoring frequency of correctness

We can apply the above analysis to implement the preference that the frequency of correctness should matter. In this preference, it is advantageous for an algorithm to focus on the common entities, which is of course what statistical learning algorithms tend to do. Thus it can be argued that a frequency of correctness measure provides an important alternative view on performance.

The above analysis should make it clear that simply counting frequency of occurrence of labels is not a very accurate approach. The issues of locality mapping, sense, synonyms, and levels of specificity still remain. If we ignore levels of specificity, then a perfect score might be achievable by simply labeling everything by “entity”. Fortunately, we can easily modify the above methodology to score frequency of occurrence so that these issues are addressed. The only change that is required is that instead of using the node weight of the most specific sense common to the two paths, we use the ratio of that weight to the weight of the node for the ground truth term. If we are evaluating a word at the same level of specificity, then the score is of course one, and the method reduces to simply counting correct labels. If the algorithm chooses to use a less specific term, (“cat” for “tiger”), then the score for being correct is less, but the chances that the term is correct are greater. Importantly, there is no advantage to using WordNet to replace a more specific term with a more general one.

To further explore the effect of choosing a more general term over a more specific one, consider the simple case of a direct ancestor, *a*, to some ground truth terms. By construction, the weights for the nodes of the ground truth terms, $n(w_i)$ are inversely proportional to the counts of each term in the ground truth. Also by construction, the weight for the node for the ancestor is inversely proportional to the sum of all counts. By our proposed rule, the score for labeling a region with ground truth w_i with *a* is:

$$s_w(a,w_i) = \frac{n(a)}{n(w_i)} = \frac{c(w_i)}{\sum_j c(w_j)} \tag{2}$$

Labeling regions by the relatively general term, *a*, only gives a positive score in the case of the descendents, w_i. The score achieved by labeling all regions by *a* is then:

$$\sum_i \sum_{G(w_i)} c(w_i) s_w(a,w_i) = \frac{\sum_i c(w_i)^2}{\sum_j c(w_j)} \tag{3}$$

Alternatively, the score of being more specific, and labeling all regions by a particular w_i will be:

$$\sum_{G(w_i)} c(w_i) s_w(w_i,w_i) = c(w_i) \tag{4}$$

Elementary considerations reveal that (3) never exceeds (4) evaluated for the w_i with largest $c(w_i)$. If the $c(w_i)$ are uniformly distributed, then (3) and (4) are equal. Equality is also approached at the other extreme when the maximum $c(w_i)$ proportionally dominates the overall count. This corresponds to the case that there is only one relevant ancestor, and the more specific and more general terms become equivalent. For distributions of $c(w_i)$ which are in-between the two extremes, there is some advantage to being more specific and choosing the most common w_i, which reflects the overall bias of the measure towards common terms.

3.6 Vocabularies without senses

Typical word prediction vocabularies, such as Corel^{TM} keywords, are not sense specific. While it is better if the sense information is available, it is important that the infrastructure can be applied regardless. In general it is easiest to deal with sense ambiguity by combining the results over senses that occur in the ground truth vocabulary. Senses not in the ground truth must be ignored. The senses in WordNet reflect, in rough order of sense number, common usage. However, a particular image corpus, such as Corel^{TM}, will have a limited subset of senses, with differing statistics. For example, the first WordNet sense for "tiger" is a kind of person, reflecting usage in the reference corpora used to develop WordNet. This sense does not occur in Corel^{TM}, but there are many examples of the second sense (animal).

In the "range of semantics" case, any convex combination of the scores over the senses will give the same answer. In the "frequency correct" case, the best score would be achieved if we assumed the most common sense in the ground truth. However, since by assumption, algorithms are not able to choose the sense, we do not need to compensate for the possibility that an algorithm could achieve this score by simply knowing the scoring methodology. Thus we propose weighting each sense in proportion to the frequency of occurrence in the ground truth data, in order that the score reflects the ambiguity when compared to other terms. Notice that frequency of occurrence is readily available from the corresponding weight of the node in our DAG.

An alternative strategy would be to attempt to infer the sense based on text using one of the many algorithms developed for that task (see, for example: Agirre and Rigau, 1995; Gale et al., 1992; Karov and Edelman, 1998; Mihalcea and Moldovan, 1998; Yarowsky, 1995), as we have done for data from the Fine Arts Museum of San Francisco (Barnard et al., 2001). However, word sense disambiguation is still an unsolved problem, and performance beyond that of assuming the most common sense is difficult to achieve (Traupman and Wilensky, 2003).

3.7 Infrastructure software components

To encourage others to explore our semantic scoring methodology, we provide a program (region_word_score) which takes as input a vocabulary and the segmentation masks for the ground truth images and produces a scoring matrix for each image. The scoring matrices have one row per image region, and one column for each vocabulary word. The matrix entries are the scores deserved for predicting that word for that region. We provide both the semantic range oriented scoring and the frequency correct oriented scoring. To support algorithms whose scoring is not defined with respect to a segmentation, we also provide an intermediate scoring matrix encoding a general score for matching of each vocabulary word with each ground truth word for a particular image, independent of segmentation.

4 Evaluation of region labeling algorithms

Our data and methodology enables significantly better evaluation of region labeling algorithms than previously available. Because we are interested in large scale experiments, we consider only learning approaches which can be trained on images with associated text and which can then produce labels for regions. We again draw the distinction between algorithms which provide words for images as a whole (auto-annotation), and those that label image regions. Every algorithm that labels regions can also provide image annotation by combining the region result in some way. For example, in the annotation results below, we assume that each region has equal weight, and use the sum of the normalized word prediction vectors over the regions.

Interestingly, many of the algorithms for image annotation which do not support region labeling nonetheless use image regions as the carriers of semantics (see, for example: Barnard and Forsyth, 2001; Jeon et al., 2003; Lavrenko et al., 2003). This makes sense because the compositional nature of images means that an image annotation word is specific to a localized entity. We expect that most image annotation methods that use regions can be easily modified to expose the region information that is implicitly used to explicitly produce region labels. However, in this work we restrict our attention to existing region labeling approaches, specifically variants of two very different approaches: generative multi-modal statistical models (Barnard et al., 2003a) and multiple instance learning (Andrews and

Hofmann, 2004; Andrews et al., 2002a; Andrews et al., 2002b; Chen and Wang, 2004; Maron, 1998; Maron and Lozano-Perez, 1998; Maron and Ratan, 1998; Tao and Scott, 2004; Tao et al., 2004a; Tao et al., 2004b; Zhang and Goldman, 2001).

4.1 Generative multi-modal models

We tested two generative multi-model statistical models, specifically the dependent and correspondence models with linear topology (no document level clustering) that we developed in earlier work (Barnard et al., 2003a). Here we assume that images and associated text are generated by choosing one or more concepts (latent factors), *l*, from a prior distribution, *P(l)*, and then by generating regions and associated words conditionally independent given the latent factors. Thus, the joint probability of an image region and a word can be expressed as

$$P(w,r) = \sum_{l} P(w \mid l) P(r \mid l) P(l) \tag{5}$$

where w denotes a word, r denotes a region, *l* indexes latent factors, $P(w \mid l)$ is a probability table over the words, and for the blob model, $P(r \mid l)$, we use a Gaussian distribution over features, with the somewhat naive assumption of diagonal covariance matrices being required to keep the number of parameters reasonable. For the experiments below we set the number of factors to be 2,000 which is roughly comparable, relative to the number of training images, to the 500 factors used previously (Barnard et al., 2003a).

The dependent and correspondence models differ in how generating words and regions jointly relates to generating words for the image as a whole. This is a critical issue because words for the image as a whole are all that is available during training. In the case of the dependent model we assume that the regions for an image provide a posterior over the latent factors which is then sampled for the words for the image. It is thus close to the discrete translation model for multi-media data (Duygulu et al., 2002). We train the model with the expectation maximization (EM) algorithm (Dempster et al., 1977), with the hidden factors responsible for each word and region being represented by missing values.

For the correspondence model we assume that regions and words are strictly emitted as pairs. This means that differing numbers of words and regions needs to be addressed, which we do by assuming duplication of words or regions where required. Training the model is more difficult than in the dependent model because specifying that each word must be paired with a region means that computing the expectations for the missing values requires marginalizing over all pairings. Since this is impractical, we use graph matching (Jonker and Volgenant, 1987) to choose a maximally likely pairing inside the expectation step of the EM fitting.

4.2 Multiple instance learning

Multiple instance learning (MIL) has been applied to image annotation and categorization in many of the above mentioned papers, but we are not aware of results on region labeling. The connection to region labeling is quite explicit because in this approach an image is labeled with a word, *w*, if the image contains a region with word, *w*. Hence we consider multiple instance learning to be a region labeling approach, despite that fact that apparently it has not been used as such previously. Multi-instance learning has also not been applied on the scale that we present below.

In the multiple instance learning paradigm, each data item is considered to be a collection (bag) of items. Each item is either a positive example or a negative example, and each bag is labeled as either a positive bag (contains at least one positive example), or a negative bag (contains all negative examples). To use multiple instance learning in the image annotation framework, each word is considered a category, and a classifier is built to determine if a region instance is in that category. Thus a classifier needs to be built for every item in the vocabulary. This is very expensive with vocabularies with hundreds of items as is the case in our experiments. Experiments reported on so far have been on a substantively lesser scale, and it is not clear to what extent this approach is appropriate for large scale image annotation---hence our interest in including it in our experiments.

Because multiple instance learning treats each word independently, it ignores the relative frequency of the words in the vocabulary. This is potentially an advantage with the “semantic range” evaluation approach, but it is a significant handicap when the frequency of being correct is being scored. Further, while multiple instance learning methods are generally developed in the context of binary classification, both methods that we implemented can output a soft scoring. Soft scoring seems more sensible with our scoring methodology because it provides an opportunity for the algorithm to break ties. Thus for each multiple instance learning method, we consider four sub-variants: hard and soft weighting, either as is, or multiplied by the empirical distribution of the training vocabulary, serving as a prior. We denote these sub-variants by the suffixes, "HARD", "SOFT", "HE", and "SE".

We implemented the expectation maximization-diverse density (EM-DD) algorithm (Zhang and Goldman, 2001), and the mi-SVM algorithm (Andrews et al., 2002b). Because of the large scale of our data, we restricted the number of positive and negative bags to 200, randomly chosen from the entire data set. When there were fewer than 200 positive examples (rare words), we included random duplicates to bring the number up to 200. For the EM-DD algorithm we set the optimal threshold for classification by holding out 10% of the training data.

We implemented the mi-SVM algorithm with a linear kernel. Experiments on the same data as in the original mi-SVM paper suggested that increasing performance with non-linear kernels is difficult, and that it would be very expensive to find good kernels with cross-validation on data of the scale below. The mi-SVM algorithm iterates until there is no change in imputed labels. We did not exceed 100 iterations even if changes were still being detected. If the set of labels kept switching in a loop within the maximum number of iterations, we chose the model which gave the best accuracy on a held out data set. For our implementation we took advantage of the freely available libsvm software (version 2.8) (Chang and Lin). We used the facility for scaling the data before generating the models.

4.3 Experimental protocol

For our experiments we prepared four data sets. We began with 39,600 Corel^{TM} images. For each of these images we have a small number of keywords (typically between three and five). We segmented the images with a modified version of normalized cuts (Gabbur, 2003; Shi and Malik., 2000). We extracted features similar to those used in (Barnard) representing color, texture, size, position, shape and color context (Barnard et al., 2003b). More specifically:

- 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 represented using the average and standard deviation of (r=R/(R+G+B), g=G/(R+G+B), S=(R+G+B)) over the region. We use this color space instead of RGB to reduce correlation among the three bands.
- Texture is represented using the average and variance of 16 filter responses. We use 4 difference of Gaussian filters with different sigmas, and two sets of 12 oriented filters, aligned in 30 degree

increments, each one at a different scale. 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.
- Color context is represented by four colors each one representing the color of adjacent regions, restricted to four 90 degree wedges (Barnard et al., 2003b).

We excluded a few images due to problems with pre-processing. We withheld the 1,014 labeled images from training. We then constructed an easier subset ("restricted") by removing images from the CorelTM CD's that did have any representative images in the 1,014 images[2]. We constructed vocabularies by limiting the keywords to those that occur 20 times in each of the data sets (50 for a second pair of data sets). Images that were left devoid of images were excluded. These 4 training sets are summarized in Table 1.

4.4 Performance measures

We used three evaluation measures, name image annotation performance and the two approaches to region labeling performance ("semantic range" and "frequency correct"). For annotation performance we use the key word prediction measure from previous work. Specifically, if there are M keywords for the image, we allow each algorithm to predict M words. We than simply record the percent correct.

For the two region labeling evaluation methods, our methodology provides a score for predicting any vocabulary word for each region. Each algorithm provides a weight vector of sum one which expresses a preference for each word (e.g. a posterior probability distribution). From this we simply give the score for the word with the maximal value.

4.5 Results

The results for three measures on four data sets are provided in Tables 2 through 5. These results confirm our expectation that the MIL methods require modification when the task rewards performance on

Data set	Minimum number of times each vocabulary word is used	Vocabulary size	Number of images
R20	20	509	26,078
R50	50	272	25,985
U20	20	656	37,337
U50	50	383	37,257

Table 1. A summary of the training data sets constructed for the experiments. In all cases the test set was drawn from the 1014 human evaluated images. (We used all with R50 and U20; two were omitted with R20 and U20). In all cases, algorithms expressed semantic prediction with respect to the training vocabularies that were then evaluated using a mapping into ground truth vocabulary.

common entities. With our proposed modifications (soft classification, multiplication by empirical distribution), we observed much improved performance on image annotation and region labeling where frequency correct was counted. In the case of annotation, the resulting performance was excellent, substantively exceeding the translation model. The results also confirm that if the task instead requires recognition independent of how common entities are (semantic range measure) the modification does not make sense.

The dependent translation model gives the best results for the "frequency correct" region labeling measure, followed by EMDD-SE. We expect that if we provide the MIL methods the empirical

Algorithm	Annotation (keyword prediction)	Region labeling semantic range performance (relative to mapped human labels)	Frequency that region semantics are correctly identified (relative to mapped human labels)
Test data empirical distribution	0.207	0.0068	0.102
Dependent translation model	0.292	0.0186	0.127
Correspondence translation model	0.251	0.0146	0.096 (–)
mi-SVM (hard)	0.016 (–)	0.0086	0.007 (–)
mi-SVM (soft)	0.033 (–)	0.0165	0.018 (–)
mi-SVM (hard, empirical as prior)	0.198 (–)	0.0089	0.074 (–)
mi-SVM (soft, empirical as prior)	0.310	0.0116	0.083 (–)
EMDD (hard)	0.021	0.0115	0.017 (–)
EMDD (soft)	0.099 (–)	0.0260	0.030 (–)
EMDD (hard, empirical as prior)	0.256	0.0146	0.045 (–)
EMDD (soft, empirical as prior)	0.240	0.0083	0.079 (–)

Table 2. The results for the R50 data set. The maximum in each column is identified by a heavy border. All numbers are an average over the 1012 human labeled images held out from training. The results for the translation models are further the average over 5 training runs with different random initializations. The annotation is how well algorithms predict the 3-5 keywords for each image. Because some of the Corel^{TM} words occur repeatedly, it is difficult to do better than the empirical distribution on tasks which reward frequency correct (first and third tasks). Algorithms that perform worse than that baseline are marked with a (–).

The second column is the semantic range performance, relative to that of the score that would be achieved using the appropriate mapped manual labeling (0.320) which is the best score possible given our machine segmentations and training vocabulary. The fact that the numbers are far less than unity reflects the fact that general object recognition is an unsolved problem. The third column is the frequency that a semantic entity is correct, again relative to that of the appropriate mapped manual labeling (1.393).

We shade the results for variants of mi-SVM and EMDD which we assumed would be less appropriate for the task being measured. In particular, we expected soft classification to do better than hard classification, and we expected that using the empirical distribution would be necessary for good results in the case of annotation and frequency correct, and that it would be a liability when applied to semantic range performance. With only one exception over all four experiments, the variant that we assumed made most sense gave the best result.

distribution of the words over the regions in the training data, then they would do as least as well, but this information is not readily available. However, these experiments suggest that it may be worthwhile to approximate it. It makes sense that the translation model does well by this measure because it is the only approach which simultaneously learns region frequency statistics and how to recognize them.

The MIL methods do proportionally better on the semantic range performance task, with the mi-SVM-SOFT being perhaps comparable to the translation method, and EMDD-SOFT being consistently better.

One high level question that we wanted to address was the extent to which the annotation score is a good proxy measure for region labeling performance. The above summary suggests that the correlation is weak at best. If we consider only the two translation methods, then annotation score correlates somewhat with region labeling performance. This concurs with our earlier findings on a much less comprehensive test (Barnard et al., 2003a). However, with disparate algorithms there seems to be significant difficulties inferring region labeling performance from annotation score. The clearest example is that the dependent

Algorithm	Annotation (keyword prediction)	Region labeling semantic range performance (relative to mapped human labels)	Frequency region semantics are correctly identified (relative to mapped human labels)
Test data empirical distribution	0.210	0.0045	0.092
Dependent translation model	0.277	0.0130	0.116
Correspondence translation model	0.241	0.0097	0.091 (–)
mi-SVM (hard)	0.018 (–)	0.0043	0.005 (–)
mi-SVM (soft)	0.040 (–)	0.0132	0.021 (–)
mi-SVM (hard, empirical as prior)	0.167 (–)	0.0061	0.072 (–)
mi-SVM (soft, empirical as prior)	0.308	0.0079	0.082 (–)
EMDD (hard)	0.018 (–)	0.0078	0.012 (–)
EMDD (soft)	0.078 (–)	0.0201	0.027 (–)
EMDD (hard, empirical as prior)	0.298	0.0095	0.064 (–)
EMDD (soft, empirical as prior)	0.341	0.0075	0.110

Table 3. The results for the R20 data set. This data set is similar to R50, but the vocabulary is larger (509 versus 272), since every word only needs to occur only 20 times in the training data instead of 50. The scores using the mapped human labels are (0.491) and (1.533) for column three and four respectively. See the caption for Table 2 for details.

translation model, which is often bettered by at least one MIL method at annotation, is the only method that does well on the task of scoring frequency of correct region semantics. A second example is that mi-SVM with soft classification is comparable to translation in region semantic range performance, but the comparable version for annotation with the empirical distribution does significantly better at annotation.

The dependent translation method always performed better than the correspondence method. This is in contrast with earlier work that suggested that the correspondence model should be better at labeling regions. While the logic still seems sound, the results suggest that the simple assumption that mismatches between the number of words and regions can be accommodated by simply repeating them is too harsh.

In summary we were surprised by how effective the combination of MIL and the empirical distribution was for annotation. Further, the EMDD-SOFT/SE pair was the most robust performer over all three tests, with the dependent translation method being reliably the best performer on the frequency correct region labeling task.

Algorithm	Annotation (keyword prediction)	Region labeling semantic range performance (relative to mapped human labels)	Frequency region semantics are correctly identified (relative to mapped human labels)
Test data empirical distribution	0.222	0.0056	0.099
Dependent translation model	0.280	0.0155	0.126
Correspondence translation model	0.238	0.0097	0.086 (–)
mi-SVM (hard)	0.017 (–)	0.0039	0.005 (–)
mi-SVM (soft)	0.028 (–)	0.0137	0.018 (–)
mi-SVM (hard, empirical as prior)	0.157 (–)	0.0078	0.064 (–)
mi-SVM (soft, empirical as prior)	0.334	0.0098	0.095 (–)
EMDD (hard)	0.013 (–)	0.0088	0.017 (–)
EMDD (soft)	0.026 (–)	0.0220	0.042 (–)
EMDD (hard, empirical as prior)	0.269	0.0116	0.074 (–)
EMDD (soft, empirical as prior)	0.352	0.0090	0.116

Table 4. The results for the U50 data set. This data set is similar to R50, but with 40% more training images, most of which are completely unlike the test images. The scores using the mapped human labels are (0.381) and (1.432) for column three and four respectively. See the caption for Table 2 for further details.

5 Discussion

Our experiments have gone significantly beyond previous work in two ways. First the data set size and the vocabularies were substantively larger than what has gone before. Second, the scale of evaluation, carried out on a region level for over 1,000 images, is also significantly larger than earlier efforts.

In our experiments we have found the MIL methods to be a very interesting alternative to the translation methodology, and that they have potential for excellent performance. However, we wish to emphasize that currently that performance comes at a *very large cost*. The MIL models are trained one word at a time at non-negligible cost, and our vocabularies contained hundreds of words. If fact, we expended an order of magnitude more computational resources on the MIL methods compared with the translation methods, which took only a few hours to learn a model for the complete vocabulary. Thus we are investigating on how to adopt the ideas from MIL that work into a more scalable learning strategy.

The details of our results are naturally a function of implementation choices and our data. What is key is that our measurement infrastructure provides necessary feedback to improve each algorithm. Importantly, the results show that the performance on the three tasks is not simply linked. How an

Algorithm	Annotation (keyword prediction)	Region labeling semantic range performance (relative to mapped human labels)	Frequency region semantics are correctly identified (relative to mapped human labels)
Test data empirical distribution	0.222	0.0039	0.0907
Dependent translation model	0.260	0.0093	0.108
Correspondence translation model	0.230	0.0058	0.089 (–)
mi-SVM (hard)	0.014 (–)	0.0018 (–)	0.003 (–)
mi-SVM (soft)	0.041 (–)	0.0080	0.013 (–)
mi-SVM (hard, empirical as prior)	0.202 (–)	0.0042	0.061 (–)
mi-SVM (soft, empirical as prior)	0.345	0.0064	0.095
EMDD (hard)	0.012 (–)	0.0042	0.006 (–)
EMDD (soft)	0.048 (–)	0.0165	0.032 (–)
EMDD (hard, empirical as prior)	0.301	0.0089	0.063 (–)
EMDD (soft, empirical as prior)	0.357	0.0064	0.108

Table 5. The results for the U20 data set. This data set is similar to R20, but with 40% more training images, most of which are completely unlike the test images. The scores using the mapped human labels are (0.557) and (1.562) for column three and four respectively. See the caption for Table 2 for further details.

algorithm performs over the suite of three tasks gives substantive insight into what it does well and why, and this can be used to take a more guided approach to improving performance. For example, it seems clear that a blend of the ideas from the MIL and translation frameworks could increase performance over what we have measured so far.

The fact that annotation performance is not trivially a good proxy for recognition validates our assumption that careful region labeling is necessary to characterize performance. Since this is a labor intensive task, we wished to do so in a way that provides maximal benefit to the vision community. In particular, we have provided an infrastructure for using the labeled data to automatically evaluate a diverse range of recognition results. Our initial experiments have already provided some insight into region labeling methods, suggesting that this kind of data is extremely useful. We look forward to hearing about other interesting applications of our infrastructure.

Acknowledgments

We acknowledge substantive help in this project. David Martin, Charless Fowlkes, and Jitendra Malik supplied the human level segmentations. The same group, together with Doron Tal, also supplied a version of the normalized cuts software, which was modified by Prasad Gabbur for the machine segmentations used in the experimental data set. Nikhil Shirahatti labeled a number of images. Finally, we are grateful to Lockheed Martin who funded the initial data gathering work and who graciously has let the data go into the public domain.

Appendix A: Labeling rules specific to humans

a) If the skin color is obvious in the segment classify human as white, black or Asian.

b) If the origin or function is obvious by the clothing label as the clothing suggest (e.g. eskimo, south american, arab, soldier, skier, cowboy, etc.)

c) If a piece of clothing is selected in addition to human label the part as the clothing (e.g. dress) or the clothing part (e.g. sleeve).

d) If the human is wearing a special gear in the segment, label it (e.g. headdress, gloves, weapon, etc.)

e) If, in the segment the hair color is obvious, label it (“blond hair”, “black hair”, “brown hair”, “red hair”, “white hair”)

f) Connect body part with human label (e.g. boy, boy_face) but not clothes (e.g. boy, shirt).

g) While labeling parts of the human face, it is not required to mention skin color for parts such as eye, nose, mouth etc. (e.g. boy, boy_eye).

Appendix B: Online infrastructure

The data and associated infrastructure is available online (kobus.ca//research/data/IJCV). For each of the 1014 images we have made available the ground truth segmentations (courtesy of the UC Berkeley segmentation group) and a text file with our labels. We also have made available a Java program for labeling which can be used to improve our labelings and label additional images. To label a new image, you will first need to create a ground truth segmentation. For this we recommend the segmentation tool available from http.cs.berkeley.edu/projects/vision/grouping/segbench) which writes segmentations in the format that our program reads. We have also made available a program (currently Linux binary) which builds a scoring matrix from machine segmentations and an arbitrary vocabulary. We also provide the feature vectors for the four data sets R20, R50, U20, and U50. Some of the infrastructure is mirrored at the UC Berkeley segmentation group web site (http.cs.berkeley.edu/projects/vision/).

Notes

1. Available from the Sowerby Research Center, British Aerospace, FPC 267, PO Box 5, Filton, Bristol BS12 7NE, England.

2. The CorelTM data is organized into CD's, each with 100 images that tend to be semantically related.

References

Agarwal, S., Awan, A. and Roth, D., The UIUC Image Database for Car Detection. Available from http://l2r.cs.uiuc.edu/~cogcomp/Data/Car/.

Agarwal, S., Awan, A. and Roth, D., 2004. Learning to detect objects in images via a sparse, part-based representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(11): 1475--1490.

Agirre, E. and Rigau, G., 1995. A proposal for word sense disambiguation using conceptual distance, 1st International Conference on Recent Advances in Natural Language Processing, Velingrad.

Andrews, S. and Hofmann, T., 2004. Multiple Instance Learning via Disjunctive Programming Boosting, Advances in Neural Information Processing Systems (NIPS 16).

Andrews, S., Hofmann, T. and Tsochantaridis, I., 2002a. Multiple Instance Learning With Generalized Support Vector Machines, AAAI.

Andrews, S., Tsochantaridis, I. and Hofmann, T., 2002b. Support Vector Machines for Multiple-Instance Learning, Advances in Neural Information Processing Systems, 15, Vancouver, BC.

Banerjee, S. and Pedersen, T., 2003. Extended gloss overlaps as a measure of semantic relatedness, International Joint Conference on Artificial Intelligence, Acapulco, Mexico, pp. 805-810.

Barnard, K., Data for Computer Vision and Computational Colour Vision. Available from kobus.ca/research/data.

Barnard, K., Duygulu, P. and Forsyth, D., 2001. Clustering Art, IEEE Conference on Computer Vision and Pattern Recognition, Hawaii, pp. II:434-441.

Barnard, K. et al., 2003a. Matching Words and Pictures. Journal of Machine Learning Research, 3: 1107-1135.

Barnard, K., Duygulu, P., Raghavendra, K.G., Gabbur, P. and Forsyth, D., 2003b. The effects of segmentation and feature choice in a translation model of object recognition, IEEE Conference on Computer Vision and Pattern Recognition, Madison, WI, pp. II:675-682.

Barnard, K. and Forsyth, D., 2001. Learning the Semantics of Words and Pictures, International Conference on Computer Vision, pp. II:408-415.

Berg, A.C., Berg, T.L. and Malik, J., 2005. Shape Matching and Object Recognition using Low Distortion Correspondence, CVPR.

Carbonetto, P., Freitas, N.d. and Barnard, K., 2004. A Statistical Model for General Contextual Object Recognition, European Conference on Computer Vision.

Chang, C.-C. and Lin, C.-J., LIBSVM -- A Library for Support Vector Machines. Available from http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

Chen, Y. and Wang, J.Z., 2004. Image Categorization by Learning and Reasoning with Regions. Journal of Machine Learning Research, 5: 913-939.

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

Duygulu, P., Barnard, K., Freitas, J.F.G.d. and Forsyth, D.A., 2002. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary, The Seventh European Conference on Computer Vision, Copenhagen, Denmark, pp. IV:97-112.

Fei-Fei, L., Fergus, R. and Perona, P., 2004. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, Workshop on Generative-Model Based Vision, Washington, DC.

Fellbaum, C., Miller, P.G.A., Tengi, R. and Wakefield, P., WordNet - a Lexical Database for English.

Fergus, R. and Perona, P., The Caltech Database. Available from http://www.vision.caltech.edu/html-files/archive.html.

Fergus, R., Perona, P. and Zisserman, A., 2003. Object Class Recognition by Unsupervised Scale-Invariant Learning, IEEE Conference on Computer Vision and Pattern Recognition, Madison, WI.

Gabbur, P., 2003. Quantitative evaluation of feature sets, segmentation algorithms, and color constancy algorithms using word prediction. Masters thesis Thesis, University of Arizona, Tucson, AZ.

Gale, W., Church, K. and Yarowsky, D., 1992. One Sense Per Discourse, DARPA Workshop on Speech and Natural Language, New York, pp. 233-237.

Jeon, J., Lavrenko, V. and Manmatha, R., 2003. Automatic Image Annotation and Retrieval using Cross-Media Relevance Models, SIGIR.

Jiang, J. and Conrath, D.W., 1998. Semantic similarity based on corpus statistics and lexical taxonomy, International Conference on Research in Computational Linguistics, Taiwan.

Jonker, R. and Volgenant, A., 1987. A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems. Computing, 38: 325-340.

Karov, Y. and Edelman, S., 1998. Similarity-based word sense disambiguation. Computational Linguistics, 24(1): 41-59.

Lavrenko, V., Manmatha, R. and Jeon, J., 2003. A Model for Learning the Semantics of Pictures, NIPS.

Leacock, C. and Chodorow, M., 1998. Combining local context and wordnet similarity for word sense identification. In: C. Fellbaum (Editor), WordNet: An Electronic Lexical Database. MIT Press, pp. 265--283.

Leibe, B. and Schiele, B., The TU Darmstadt Database. Available from http://www.vision.ethz.ch/leibe/data/.

Lin, D., 1998. An information-theoretic definition of similarity, International Conference on Machine Learning.

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

Maron, O. and Lozano-Perez, T., 1998. A framework for multiple-instance learning, Neural Information Processing Systems. MIT Press.

Maron, O. and Ratan, A.L., 1998. Multiple-Instance Learning for Natural Scene Classification, The Fifteenth International Conference on Machine Learning.

Martin, D., Fowlkes, C., Tal, D. and Malik, J., 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, International Conference on Computer Vision, pp. II:416-421.

Mihalcea, R. and Moldovan, D., 1998. Word sense disambiguation based on semantic density, COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems, Montreal.

Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D. and Miller, K.J., 1990. Introduction to WordNet: an on-line lexical database. International Journal of Lexicography, 3(4): 235 - 244.

Opelt, A. and Pinz, A., TU Graz-02 Database. Available from http://www.emt.tugraz.at/~pinz/data/GRAZ_02/.

Resnik, P., 1995. Using information content to evaluate semantic similarity in a taxonomy, International Joint Conference on Artificial Intelligence, Montreal.

Shi, J. and Malik., J., 2000. Normalized Cuts and Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(9): 888-905.

Tao, Q. and Scott, S., 2004. A Faster Algorithm for Generalized Multiple-instance Learning, Seventeenth Annual FLAIRS Conference, Miami Beach, Florida, pp. 550-555.

Tao, Q., Scott, S., Vinodchandran, N.V., Osugi, T.T. and Mueller, B., 2004a. An Extended Kernel for Generalized Multiple-Instance Learning, IEEE International Conference on Tools with Artificial Intelligence.

Tao, Q., Scott, S.D. and Vinodchandran, N.V., 2004b. SVM-Based Generalized Multiple-Instance Learning via Approximate Box Counting, International Conference on Machine Learning, Banff, Alberta, Canada, pp. 779-806.

Torralba, A., Murphy, K.P. and Freeman, W.T., The MIT-CSAIL Database of Objects and Scenes. Available from http://web.mit.edu/torralba/www/database.html.

Torralba, A., Murphy, K.P. and Freeman, W.T., 2004. Sharing features: efficient boosting procedures for multiclass object detection, IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, pp. II:762--769.

Traupman, J. and Wilensky, R., 2003. Experiments in Improving Unsupervised Word Sense Disambiguation. CSD-03-1227, Computer Science Division, University of California Berkeley.

Vivarelli, F. and Williams, C.K.I., 1997. Using Bayesian neural networks to classify segmented images, IEE International Conference on Artificial Neural Networks.

Weber, M., Welling, M. and Perona, P., 2000. Unsupervised Learning of Models for Recognition. In: D. Vernon (Editor), 6th European Conference on Computer Vision, pp. 18-32.

Yarowsky, D., 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods, 33rd Conference on Applied Natural Language Processing. ACL, Cambridge.

Zhang, Q. and Goldman, S.A., 2001. EM-DD:An improved multiple-instance learning technique, Neural Information Processing Systems.