Efficient re-indexing of automatically annotated image collections using keyword combination

Alexei Yavlinsky and Stefan Rüger

Department of Computing
Imperial College London
South Kensington Campus, London SW7 2AZ

ABSTRACT

This report presents a framework for improving the image index obtained by automated image annotation. Within this framework, the technique of keyword combination is used for fast image re-indexing based on initial automated annotations. It aims to tackle the challenges of limited vocabulary size and low annotation accuracies resulting from differences between training and test collections. It is useful for situations when these two problems are not anticipated at the time of annotation. We show that based on example images from the automatically annotated collection, it is often possible to find multiple keyword queries that can retrieve new image concepts which are not present in the training vocabulary, and improve retrieval results of those that are already present. We demonstrate that this can be done at a very small computational cost and at an acceptable performance tradeoff, compared to traditional annotation models. We present a simple, robust, and computationally efficient approach for finding an appropriate set of keywords for a given target concept. We report results on TRECVID 2005, Getty Image Archive, and Web image datasets, the last two of which were specifically constructed to support realistic retrieval scenarios.

1. INTRODUCTION

Owing to increasingly large amounts of unlabelled digitised images and videos available on the Internet and in private archives, the challenge of content based image retrieval is receiving growing attention from the information retrieval and the computer vision communities. A number of researchers have recently investigated the use of automated image annotation for querying image databases using text as an alternative to the more common way of querying with exemplar images, arguing that this is a more natural modality for most users. Substantial progress has been made by assuming that users can cope with imperfect retrieval results and performing retrieval of images according to the *probability* of containing a particular concept of interest. Such probabilities are assigned to unlabelled images based on models of low-level image feature distributions for each concept, and good retrieval performance has been achieved with this approach on a number of different datasets.[1–6] Progress in this direction has thus enabled a number of real-world image and video retrieval systems to take advantage of automated annotation techniques to improve retrieval performance.[7,8]

Despite encouraging results reported in the literature, one will likely face challenges when applying automated image annotation to real world image collections. Current annotation models are usually trained on high quality reference datasets which are manually labelled by specialists; such datasets typically have relatively few keywords with adequatly large samples of associated training images, thus limiting the size of the vocabulary. On the other hand, adding new keywords to the vocabulary will typically incur substantial computational cost for test collections of large size. Finally, while results of automated annotation approaches are traditionally reported on test collections which come from the same source as the images used for training keyword models, this is unlikely to be the case when one attempts to annotate large, real world image collecions in which images come from a variety of sources.

In this report we propose an indexing framework based on automated image annotation that aims to support efficient refinement and augmentation of the annotation vocabulary for large test collections. Within this

Further author information: (Send correspondence to Alexei Yavlinsky.)
Alexei Yavlinsky.: E-mail: alexei.yavlinsky@imperial.ac.uk
Stefan Rüger.: E-mail: s.rueger@imperial.ac.uk

framework, keyword probabilities, assigned to images by a traditional annotation model, are reused for labelling images with new concepts which are not present in the annotation vocabulary and for improving retrieval of those concepts that do have corresponding keywords, but which are not modelled accurately because of differences between training and test sets. This is achieved by automatically identifying a small set of keywords which discriminate best between images that contain the given concept and those which do not. This approach is equivalent to using keyword probabilities as highly selective image features for classifying images, and we borrow this idea from the field of text retrieval, where word frequencies are used for document retrieval and categorisation.

Our framework is intended for test collections in which ground truth relating to a particular concept may become available for a proportion of images *after* automated annotation has been performed, such as a large video archive of a broadcasting company, in which new concepts are constantly discovered by staff. In this case, it is much quicker to combine precomputed keyword probabilities for labelling the remainder of the collection with respect to that concept instead of doing so with a dedicated annotation model trained directly on the new images. It should be noted that our objective is not to beat a dedicated annotation model, instead we would like to achieve an acceptable tradeoff between the computational efficiency and accuracy of our approach, as compared to the dedicated model.

In this report we present a simple, efficient, and robust algorithm for keyword selection and contrast it to the Support Vector Machine classifier adapted for this task. Both keyword combination approaches are then compared to modelling each concept using a dedicated annotation model and differences in their respective accuracies are reported. To illustrate the generality of this approach, we state our retrieval results on three large yet realistically constructed image collections, each of which is significantly different to the dataset that was used for training keyword models.

The remainder of this report is organised as follows. Section 2 gives an overview of related work. Section 3 presents the keyword combination algorithms and the formal evaluation procedure which we use to carry out the experiments. We report expermental results and analyse robustness of our approach in Section 4 and conclude with a discussion of their implications in Section 5.

2. RELATED WORK

The goal of automated image annotation is to model the process of a human annotator assigning relevant keywords to images; probabilistic approaches to this problem assume that a human is prompted for a single annotation keyword for the image x, and that he chooses keyword w from a finite vocabulary W, with probability $p(w|x)$. After the pioneering work of Mori[9] several such approaches have been developed. Duygulu *et al.*[1] created a discrete 'vocabulary' of clusters of image regions (blobs) across an image collection and applied a model, inspired by machine translation, to translate between the set of blobs comprising an image and annotation keywords. A number of nonparametric approaches have also been investigated: Jeon *et al.*[2] recast image annotation into a problem in cross-lingual information retrieval, applying a cross-media relevance model to perform image annotation and ranked retrieval, obtaining better retrieval performance than in the translation model of.[1] Lavrenko *et al.*[3] adapted the model of[2] to use continuous probability density functions to describe the process of generating blob features, hoping to avoid the loss of information related to quantization; they achieve substantially better retrieval performance on the same dataset. Feng *et al.*[4] replace blobs with rectangular blocks and model image keywords using a multiple Bernoulli distribution, thus achieving better results than in.[3] In an attempt to reduce the computational cost of the nonparametric methods, Ghoshal *et al.*[5] annotate images using a Hidden Markov Model in which states represent concepts conditioned upon low level image features, but report that the performance is significantly below that reported in.[3] Whilst the above approaches focus on different ways to model image-keyword relationships, Yavlinsky *et al.*[6] investigate different low-level features which can be used for automated annotation and show that modelling image keywords using features as simple as global colour statistics can provide near state of the art performance.

In our framework, images are retrieved using keyword combinations; our hypothesis is that given a particular

concept, a suitable combination of precomputed probabilities can be used to identify relevant images*. We borrow the idea of retrieval using multiple keywords from *automatic query expansion* and *text categorisation* – techniques which are used in text retrieval. Automatic query expansion is a relevance feedback method which improves recall by inserting related query terms into the user's query.[10,11] Additional terms are selected from documents matching the user's information need; these documents are either selected manually or are defined to be the top n documents returned in response to the original query (the latter approach is usually referred to as blind relevance feedback[12,13]). On the other hand, text categorisation is concerned with constructing a classifier which correctly groups documents into two or more predefined categories. Such a classifier is trained on a manually categorised document collection and its output is typically a weighted combination of different term frequencies (for a good overview of text categorisation the reader is referred to[14]). In the next section we describe in detail how we adapt these methods for image retrieval.

The idea of combining concept detector outputs in image retrieval has been studied before.[15–23] Naphade *et al.*[15,16] express relationships between concept classifiers in video keyframes using a Bayesian network, with the aim of modelling higher-level semantic classes of such keyframes. Smith *et al.*[17] perform query-by-example retrieval with images represented by vectors of different concept classifier outputs and this method proves effective for a range of image queries on the TRECVID dataset. The latter approach was subsequently used by Rasiwasia *et al.*[23] to improve recall in example driven retrieval. Natsev *et al.*[18] exploit concept classifier dependence in video keyframes to construct new classifiers for concepts with insufficient numbers of training examples. Hauptmann *et al.*[20] combine concept predictions using a logistic regression classifier, whilst Amir *et al.*[19] do so with a Support Vector Machine. Wu *et al.*[21] attempt to model relationships between concepts based on a predefined ontological hierarchy. Finally, Yan *et al.*[22] use a range of graphical models for representing concept relationships to enhance concept detection. Our approach is novel compared to this body of work in the following ways. Instead of seeking to improve query-by-example or concept detection performance, our aim is to provide a computationally cheap alternative for refining existing concepts and modelling ones that do not exist in the training vocabulary. We do so by selecting a small number of keywords that can represent the concept of interest with a reasonable tradeoff in accuracy compared to a dedicated, annotation model. Importantly, we explicitly investigate situations where the training collection from which annotation models are estimated is *substantially different* to the collection that is being indexed. We believe that in such cases it is most likely that the quality of initial annotations will need to be improved over time.

3. OUR FRAMEWORK

Our framework integrates the two text retrieval techniques described in the previous section. Initially the test collection is indexed using a probabilistic model trained on images from an unrelated dataset. Subsequently, given a set of images from the test collection which turn out to be relevant to a particular concept, we automatically choose a small set of keywords which, when combined, discriminate best between the above images and images which are irrelevant to that concept. There are two possible situations when this technique can be used:

The target concept has a corresponding keyword in the vocabulary. In this case we are using keyword combination in an attempt to improve retrieval performance of this concept as compared to using a single keyword.

The target concept has no corresponding keywords in the vocabulary. In this scenario we are using keyword combination as a substitute to training a dedicated model for the new concept and using it to annotate the rest of the test collection.

We next give details of how images are automatically annotated and of our keyword combination methods and their computational cost. We end this section with a formal description of our experimental procedure used for evaluating our approach.

*Note the distinction in our use of the terms 'keyword' and 'concept' in this report: 'concept' refers to the information need of a user, whereas 'keyword' refers to the probability of a keyword, assigned to images by the annotation model and used for subsequent querying.

3.1. Automated image annotation

The fundamental building blocks of our framework are automatically generated annotations with respect to a particular training collection. For this we use a simple nonparametric model proposed by,[6] which is reported to perform on par with other, more elaborate, annotation methods. The probability of a human picking a keyword w given an image x is defined as

$$p(w|x) = \frac{f(x|w)p(w)}{f(x)}. \tag{1}$$

Here we interpret $f(x)$ as the probability density of image x and $f(x|w)$ as the density of x conditional upon the assignment of keyword w. Image x is represented by a d-dimensional feature vector and the conditional density $f(x|w)$ is estimated using kernel smoothing:

$$\hat{f}(x|w) = \frac{1}{|S_w|} \sum_{x^{(i)} \in S_w} k(x - x^{(i)}; h), \tag{2}$$

where S_w is the training sample of images annotated with the keyword w and $k(\cdot)$ is the kernel function which we define precisely in 4.3. We model the prior probability $p(w)$ of the keyword w as

$$\hat{p}(w) = \frac{|S_w|}{\sum_w |S_w|}, \tag{3}$$

where $|S_w|$ is the size of the training sample for the keyword w. Finally we define $f(x) = \sum_w f(x|w)p(w)$.

It should be noted that the computational complexity associated with assigning the probability of a single keyword w to an image x is linear in the number of examples in the training sample S_w, and in the dimensionality of x, making it a computationally costly annotation method for keywords with large numbers of associated training images.

3.2. Models for keyword combination

Greedy keyword multiplication. Suppose that we ask an annotator to repeatedly assign a new keyword n times for an image x. The probability of the event that keywords $w_1, \ldots, w_n$ are selected can be modelled as

$$p(w_1, \ldots, w_n|x) = \prod_{i=1}^{n} p(w_i|x). \tag{4}$$

The above equation exploits the assumption that all keywords are assigned independently. The goal of the model is to find a set of keywords for which the average precision of a given concept is maximised when images are ranked according to Equation (4). We solve this problem by fixing the number of keywords, n, and using a greedy search algorithm that, for a given concept, repeatedly adds the keyword which produces the largest increase in average precision on the training set, until the desired number of keywords are inserted into the product.

Greedy keyword addition. It is also possible to model the event that given an image x the annotator will pick a keyword from a set W of n different keywords. The probability of this event is

$$p(W|x) = \sum_{w \in W} p(w|x). \tag{5}$$

For this model we likewise use greedy search to select n keywords which maximise the concept's average precision once images are ranked according to Equation (5).

Linear combination model. Here we take a different view to the above two models and use the automatically generated keywords within the standard classification framework. We aggreate the keyword probabilities of each image into a vector v and use a Support Vector Machine (SVM) to find a linear hyperplane that discriminates well between keyword vectors of images which contain a particular concept and of those which do not. This is

similar to the work of Joachims[24] in which term vectors of text documents are classified with an SVM, and to that of Amir[19] and Hauptmann,[20] as described in Section 2.

SVMs[25] are learning machines that are capable of performing binary classification. Given a set of l training points belonging to two separate classes the objective of the SVM is to separate them with a hyperplane function $\langle w, v\rangle + b = 0$ such that

$$\min |\langle w, v_i\rangle + b| = 1,$$

subject to the constraint

$$y_i[\langle w, v_i\rangle + b] \geq 1.$$

This specifies that the hyperplane must separate the two classes correctly with the maximum margin possible. The solution to this problem is found by the minimisation of the function $\frac{1}{2}\|w\|^2$ subject to the above constraints, which can be solved using quadratic programming.

The SVM is trained on keyword vectors to derive the hyperplane that separates the positive and the negative examples with least error. Once trained, the relevance score of an unseen image is defined as the distance of its keyword vector v to the hyperplane, which is just a linear weighted sum of the keyword vector's components offset by the constant factor b. In this context, the hyperplane represents the set of weights for the keywords that minimise the error on the training sample.

SVMs are known to have favourable generalisation properties compared with many other binary classifiers and fast implementations are widely available. One such implementation - SVM^{light}[26] - is used for our experiments.

3.3. Computational complexity

The computational cost associated with labelling images using the keyword combination methods is slight compared to doing so with a concept-specific dedicated nonparametric model described in Section 3.1. Greedy keyword combination requires only n multiplication or summation operations per image, respectively, where n is the number of keywords we wish to choose. The linear combination model requires m summation operations, where m is the number of nonzero elements in the hyperplane vector w. By contrast, the cost of the nonparametric model grows linearly in the number of training examples and in the dimensionality of the low-level feature vectors, as noted earlier. Both greedy and linear combination approaches have inexpensive parameter estimation procedures.

3.4. Performance evaluation

We use the following formal experimental procedure to evaluate our framework. We are asked to index a collection of images, A, in which images are manually labelled with concepts from vocabulary W_A (but which are assumed to be unobservable at the time of indexing). We annotate the entire collection using the nonparametric annotation model, described in Section 3.1, trained on a *reference collection* B, with its respective vocabulary W_B. A is then split into training and test sets, A_{train} and A_{test}, respectively. For each concept in W_A, keyword combination models outlined in Section 3.2 are trained on A_{train} and the resulting combinations of keywords picked from W_B are evaluated on A_{test}. Accuracies obtained on A_{test} (defined as average precision values) are averaged across all concepts in W_A. This simulates the process of re-indexing the collection using keyword combination for every concept in W_A. Additionally, we train a nonparametric annotation model on A_{train} and use it to annotate A_{test} directly with respect to all concepts in W_A. This approach serves as the *upper bound*, *i.e.* it shows how accurately it is possible to re-index A_{test} with a dedicated, state-of-the-art method. By comparing our model accuracies to the upper bound we will be able to establish the relative effectiveness of our keyword combination strategies. An important condition for assessing the generality of our approach is that the images in the reference collection come from a different source to those which are in the collection we are trying to index. This is meant to prevent any unanticipated overfitting of keywords from W_B to concepts in W_A. We observe this requirement in our experiments.

Why is the nonparametric density estimate our upper bound? This is a subtle but an important point in our work. In principle, there is nothing fundamentally limiting the precision of keyword combination to be lower than that of a nonparametric density estimate of the target concept in low-level feature space. However, since the keyword probabilities have been estimated in the same feature space but on a different dataset, it is

unlikely that combining a small number of them will model the distribution of the target concept in low-level feature space as accurately as the nonparametric estimate. This is acceptable as our aim is to achieve a significant saving in computational effort at a reasonable performance tradeoff.

4. EXPERIMENTAL RESULTS

4.1. Image data

The goal of our experiments is to evaluate whether our approach addresses the challenges outlined in the introduction — the small size of the vocabulary and the content disparity between training datasets and large, realistic image collections. Unfortunately, datasets which have been used in the past to evaluate automated annotation are not particularly suitable for this task. For instance, the collection of Corel images used by[1–4] has a very small test set of just 500 images. Additionally, on closer inspection many of the test images appear to be very similar to the corresponding training images – an unlikely situation when one attempts to annotate a very large set of images such as *e.g.* portions of the World Wide Web.

For our experiments we define a realistic collection as one that contains 10,000 or more images and in which images come from a variety of sources. We have compiled two such datasets ourselves – *Getty* and *Web images* – which, we believe, reflect two different realistic image retrieval scenarios. We also report results on a subset of the TRECVID 2005 collection, and we use a selection of manually annotated images from the Corel Photo Stock as our reference collection. We use the latter to generate the initial automatic annotations for these three collections. The list of image IDs, their respective annotations and the vocabularies can be downloaded for each dataset from our report website[†].

We give details of all four datasets below, while at the same time we very much hope that ultimately there will be a lively debate in the community as to what constitutes a truely 'realistic' dataset.

Getty dataset. We attempted to build a collection of images which are similar to those stored and distributed by commercial image archives. For this we downloaded 20,000 medium-resolution thumbnails of photographs from the Getty Image Archive website[‡], together with the annotations assigned by the Getty staff to catalogue those pictures. The photographs were obtained by submitting the following two queries to the Getty website – "*photography, image,* not *composite,* not *enhancement,* not *'studio setting',* not *people*" (18,796 images) and "*photography, image, people,* not *composite,* not *enhancement,* not *'studio setting'*" (1,204 images) – both queries having the additional search option to exclude illustrations. With these queries we sought to obtain a random selection of photos, which excludes any non-photographic content, any digitally composed or enhanced photos and any photos taken in unrealistic studio settings. The constraint to exclude people in the majority of the photographs is imposed to reduce the semantic ambiguity of annotations. The resulting dataset contains pictures from a number of different photo vendors, which – we hope – reduces the chance of unrealistic correlations between keywords and image contents.

Annotations for Getty images come in three different kinds: subjects (e.g. '*tiger*'), concepts (e.g. '*emptiness*') and styles (e.g. '*panoramic photograph*'). We created our vocabulary for this dataset by selecting the following 45 annotations from over 10,000 available subject keywords:

food, one animal, one person, night, leopard, cloud, mammal, sky, clear sky, christmas tree, lion, underwater, field, leaf, sunset, vegetable, tree, snow, urban scene, building exterior, school of fish, stem, rainforest, mountain, lush foliage, horizon, dusk, sand dune, fog, cloudscape, skyline, window, sea, winter, skyscraper, cityscape, non-urban scene, woods, sun, grass, river, plant, flower, landscape, insect

We randomly split the dataset into training and test partitions of equal size.

Web images. We constructed a different dataset to measure the effectiveness of our approach on images that can typically be found on the World Wide Web. For this we obtained a set of images from the PicSearch[§] search engine with the 11 following queries that define the dataset vocabulary:

[†]http://mmis.doc.ic.ac.uk/www-pub/reindexing/

[‡]http://creative.gettyimages.com

[§]http://www.picsearch.com

aerial view, building exterior, clouds, crowd, flower, grazing animal, jug, mountain, mugshot, sunset, underwater fish

We manually filtered out irrelevant images from each query result, leaving between 400 and 1,200 images per query and 8,714 images in total. We then used the ESP dataset[27] to obtain a representative sample of the great variety of images available on the web that are not captured by the above categories[¶]. From this dataset we excluded images with annotations related to any of our categories leaving 61,100 images which we label as 'nonrelevant'. Images downloaded from PicSearch were aggregated and randomly split into equal training and test partitions and the remaining ESP images were randomly partitioned such that the training and the test partitions have, respectively, 10,000 and 59,814 images in total. Partitioning the collection in this way reflects the fact that most of the images on the World Wide Web are irrelevant to any given user query.

TRECVID 2005. Finally, we use TRECVID 2005 as our third evaluation dataset[28] with collaborative annotations provided by IBM.[29] In this dataset, each keyframe is tested for the presence of 39 different concepts by multiple independent human annotators. We processed these annotations in a simple vote-like manner: a keyframe is judged to be relevant to a particular concept if the majority of the annotators believe that it is in fact relevant. We removed the concepts *Face*, *Person*, and *Outdoor* as these occur far too frequently in the data and would therefore favourably bias the results. 20,000 keyframes were kept for the training set and 41,906 keyframes were used for evaluation.

Corel dataset. We use 14,081 images from the Corel Photo Stock as our reference collection to automatically assign keywords to images in the above datasets. We compiled a diverse vocabulary of 253 keywords from the available annotations and the exact details can be found on our report website.

4.2. Image features

We use global colour, texture, and frequency domain features to model image densities. The image is split into 9 equal, rectangular tiles; for each tile we compute the mean and the variance of each of the HSV channel responses, as well as Tamura coarseness, contrast and directionality texture properties obtained using a sliding window.[30] Additionally we apply a Gabor filter bank[31] which has 24 filters (6 scales $\times$ 4 orientations) and compute the mean and the variance of each filter's response signal on the entire image. This results in a 129-dimensional feature vector for each image. Our choice of these simple features is motivated by results reported in[6] which demonstrate that simple colour and texture features are suitable for automated image annotation. Implementation details of Tamura and Gabor features used in this report can be found in.[32]

4.3. Choice of kernel for density estimation

Prior to carrying out the experiments we investigated two different kernel functions for nonparametric density estimation, as defined in Section 3.1. We considered the commonly used d-dimensional Gaussian kernel

$$k_G(t;h) = \prod_{l=1}^{d} \frac{1}{\sqrt{2\pi}h_l} e^{-\frac{1}{2}\left(\frac{t_l}{h_l}\right)^2}, \tag{6}$$

and the d-dimensional Laplace kernel

$$k_L(t;h) = \prod_{l=1}^{d} \frac{1}{h_l} e^{-\frac{|t_l|}{h_l}}, \tag{7}$$

where $t_l = x_l - x_l^{(i)}$, the difference between l^{th} components of the feature vector x and the basis point $x^{(i)}$. We set each bandwidth parameter h_l by scaling the sample standard deviation of feature l by the same constant λ. We evaluated the annotation accuracy on 50% of images randomly withheld from the training partitions of the Getty, Web images, and TRECVID 2005 datasets and the results are reported in Table 1. For each dataset the the factor λ was optimised with respect to mean average precision using the golden search algorithm. Since the use of the Laplace kernel results in better annotation accuracies in all three cases we chose it for our experiments.

[¶]http://hunch.net/~learning/ESP-ImageSet.tar.gz

	Laplace	Gaussian
Web images	0.4555	0.3958
Getty	0.1835	0.1574
TRECVID	0.3433	0.3148

Table 1. Mean average precision for Gaussian and Laplace kernel smoothing on withheld data

	NPDE	Product	Sum	SVM	Random
Web images	0.3309	0.2202	0.1413	0.1911	0.0072
Getty	0.2120	0.1548	0.1434	0.0923	0.0327
TRECVID	0.3693	0.1223	0.0957	0.1209	0.0244

Table 2. Keyword combination mean average precision. NPDE – dedicated nonparametric annotation model (upper bound), SVM – linear combination model

For computing the upper bound figures we use the optimal λ values obtained for each dataset, and for generating keyword probabilities we use the scaling value obtained in the same manner on a withheld portion of the Corel collection.

4.4. Annotation performance

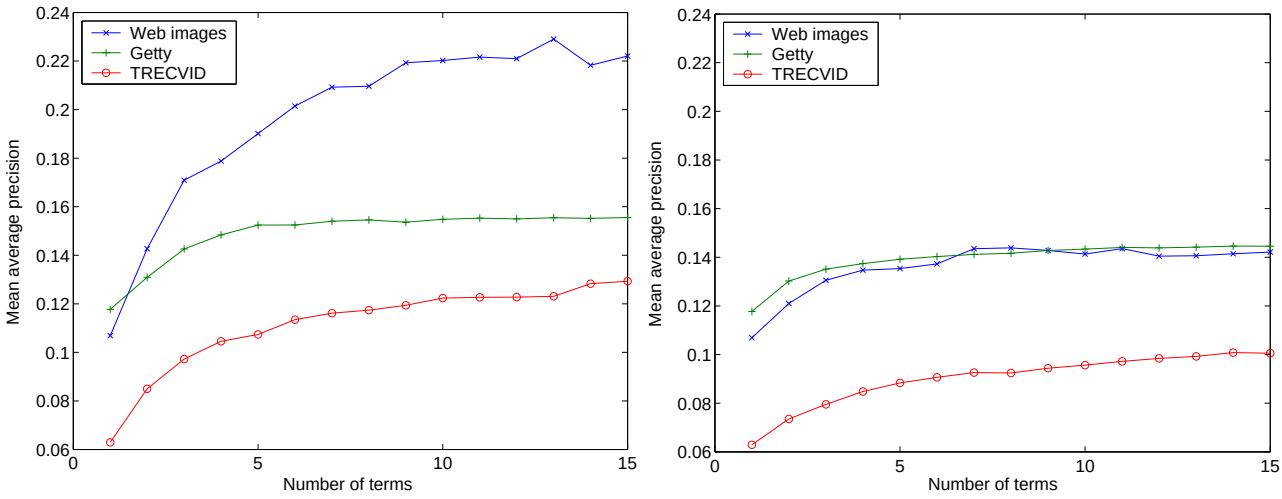
We report the mean average precision of our keyword combination methods in Table 2. By default, 10 keywords are used for each concept by the greedy addition and multiplication methods. One can observe from these results that greedy multiplication consistently outperforms the two other combination methods. Combining 10 keywords this way recovers 67% of the upper bound accuracy on Web images, 73% on the Getty dataset but only 33% on TRECVID 2005 data. The comparatively poor performance on the latter can be explained by the the fact that the TRECVID concepts are very different in nature to the keywords of our reference dataset, Corel. The linear combination model gives similar accuracies to greedy multiplication on Web images and on TRECVID, but surprisingly has the worst performance on the Getty dataset. The accuracy of greedy addition is significantly lower than that of the multiplication method, except for the Getty case where the two do not appear to differ by much. All reported figures are significantly above random chance.

With the exception of TRECVID figures, the results are encouraging – multiplying only 10 keywords is sufficient for obtaining approximately 70% of the upper bound performance on average. Table 3 compares the computational cost of re-indexing the Web collection with one concept using greedy multiplication, to that of the nonparametric density estimate. 10,000 images are used for training and 59,814 are annotated automatically, as before. The table shows that having this performance trade-off allows us to re-index the above collection 100 times faster. Selecting keywords for multiplication in a greedy fashion takes most of the time – annotation itself requires only 9 multiplication operations per image. In constrast, the annotation cost of the nonparametric model grows linearly in the number of training examples and in the dimensionality of the low-level feature vectors.

Increasing the number of combined keywords improves the accuracy of both greedy search methods on all datasets, as Figure 1 shows, though the improvement for the multiplication method appears to be generally more profound. For example, we note a two-fold accuracy improvement when 10 keywords are chosen to be multiplied instead of just one on the Web images, as Table 4 shows. Figures 3 and 4 show, respectively, how the first 16

	Product	NPDE
Training	$64.7s$	$0s$
Annotation	$0.2s$	$8613.2s$
Total	$64.9s$	$8613.2s$

Table 3. CPU time taken to annotate the Web collection with one concept (seconds): greedy multiplication re-indexing vs. nonparametric density estimate. Measured on an Intel Pentium 4 3.00 GHz. System implemented on Sun's Java 1.5 platform



(a) Greedy multiplication (b) Greedy addition

Figure 1. Performance of greedy keyword selection vs. the number of combined keywords

search results change for the Web dataset's *flower* and TRECVID's *people marching* concepts, as more selected keywords are multiplied.

It is also interesting to observe which keywords are selected by the greedy methods. Table 4 shows the keywords picked by the multiplication method for each of the concepts in the Web images dataset. Note that for some concepts the corresponing keyword is not selected as the first one by the greedy search; this shows that it may be suboptimal to use keywords that match target concepts because our keyword models were estimated on a different dataset. It appears that the selected keywords are sometimes irrelevant to the target concept, however, since the probabilities of these keywords are estimated using low-level image features, an 'irrelevant' keyword may well be helpful in characterising a particular visual aspect of that concept. For example, insects may often appear in images containing flowers, thus making *insect* a useful keyword for the *flower* concept.

Our approach has the capability of retrieving concepts which are not present in the vocabulary of the reference collection. The Web dataset has three such concepts: *crowd*, *jug* and *mugshot*. Table 4 shows that for all three concepts the accuracy of the greedy multiplication approach is substantially lower than the nonparametric upper bound, however adding more keywords into the product is still beneficial in these cases. The approach fails when there are no keywords that characterise the target concept well. This is illustrated in Table 5 which shows examples of greedy multiplication for different concepts in the Getty and in the TRECVID collections. For each collection there is one example of a concept that is modelled well by our approach, one – not so well, and one that cannot be modelled at all.

4.5. Robustness analysis

Up to this point we have used large quantities of training images to generate relevant keyword combinations. In this section we investigate how the performance of our approach is affected by decreasing the number of positive training examples. For each concept, in turn, we retain a random sample of $n\%$ of all training images relevant to that concept, whilst keeping all other training images which are irrelevant to it. This reflects the situation where the positive examples are hard to obtain, whereas negative images are readily available in large quantities. We compute the mean average precision across all concepts for each percentage level n, and to minimise the effect of sampling errors we repeat the entire procedure 5 times and report the average across all trials.

Figure 2 reports accuracy as the fraction of mean average precision when 100% of positive examples are used versus the fraction of positive examples retained for each concept. We have chosen this visualisation to make the degradation behaviour comparable across all datasets. The results indicate that the mean average precision degrades gracefully as fewer positive examples remain available, and that performance decreases in a similar

Concept	Corel keywords	Single best keyword AP	10 keyword AP	NPDE AP
Aerial view	**panorama**, rocks, aerial, nature, vegetation rock, hillside, river, detail, harbour	0.0542	0.1908	0.3042
Building exterior	**structure**, castle, building, exterior, rocks, architecture, scenic, stone, sky, bridge	0.1023	0.1971	0.2258
Clouds	**cloudy**, wave, snow, ruins, clouds, aerial, birds, river, sky, detail	0.2377	0.2889	0.3491
Crowd	**pebbles**, forest, fortress, wings, temple, canyon, wildlife, table, detail, river	0.0202	0.0568	0.2033
Flower	**orchid**, flora, insect, herd, closeup, leaves, flower, cat, blossoms, tree	0.0893	0.2335	0.2428
Grazing animal	**grass**, herd, agriculture, field, wildlife, canyon, vegetation, building, animal, landscape	0.0842	0.2094	0.2636
Jug	**furniture**, duck, dessert, face, hills, doorway, wilderness, metal, drink, mountains	0.0405	0.1156	0.4003
Mountain	**mountains**, fortress, mountain, rock, park, sky, detail, rocks, scenic, vegetation	0.1838	0.2910	0.3605
Mugshot	**pets**, river, face, castle, sky, people, canyon, smoke, stone, forest	0.0774	0.2124	0.4407
Sunset	**dawn**, sunset, lighthouse, sun, canyon, sky, vegetation, detail, scenic, city	0.2398	0.4793	0.6131
Underwater fish	**underwater**, mountain, vegetation, river, abstract, grass, fish, glacier, desert, branch	0.0464	0.1476	0.2360
		Single best keyword MAP	10 keyword MAP	NPDE MAP
		0.1069	0.2202	0.3309

Table 4. Results of greedy keyword multiplication on Web images – keywords are shown in the order selected by the algorithm

Concept	Corel keywords	10 keyword AP	NPDE AP
Skyline *(Getty, good)*	skyline, structure, aerial, buildings, lake, tower, valley, city, mount, reflection	0.1867	0.1882
Vegetable *(Gety, ok)*	pets, herd, fruit, sea, pond, cloudy, detail, plant, sand, city	0.1312	0.2380
Christmas tree *(Getty, poor)*	pets, railway, sunlight, textile, museum, blossoms, cat, roof, shadow, valley	0.0107	0.1886
Sky *(TRECVID, good)*	sky, scenery, castle, horizon, grass, exterior, clouds, rocks, desert, building	0.4160	0.5471
Crowd *(TRECVID, ok)*	pebbles, hillside, aerial, farm, detail, forest, stone, rock, buildings, cat	0.3515	0.5257
Corporate leader *(TRECVID, poor)*	hat, cliff, tundra, exterior, bridge, structure, mountain, arch, sky, wildlife	0.0222	0.3721

Table 5. A selection of greedy keyword multiplication results on the Getty and TRECVID 2005 datasets

manner for all three datasets; this shows that our approach is robust towards different selections of positive training examples and towards their reduced availability.

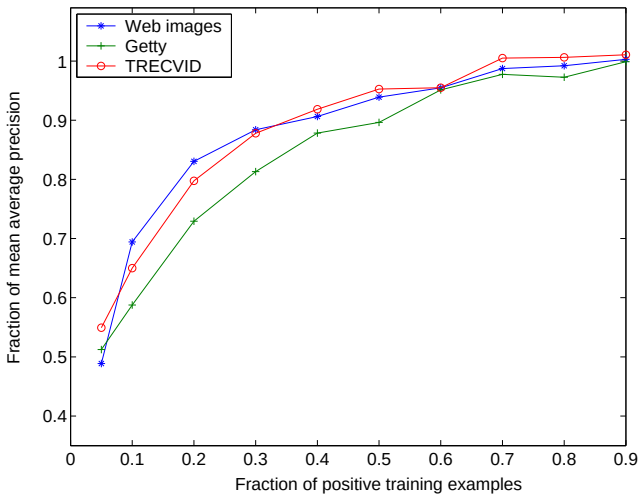


Figure 2. Relationship of greedy multiplication performance to the number of positive training examples

5. DISCUSSION AND CONCLUSIONS

We have presented a framework for efficient re-indexing of large, realistic image collections using keyword combination. We have shown that using this simple approach one can refine existing concepts and add new ones into the training vocabulary at a very small computational cost and only a moderate performance tradeoff, compared to a dedicated annotation model. We believe that this functionality could be useful for large scale image search engines when computational resources are scarce.

Whilst the keyword multiplication model attains comparable accuracy to the upper bound for concepts in the Getty and Web datasets, perormance on the TRECVID dataset reveals that the concepts in this collection are far too different to those modelled by our Corel vocabulary. Nonetheless, performance figures across all datasets improve consistently when more keywords are added and degrade similarly when the number of positive examples is reduced. This emphasises the generality of our approach.

It is interesting to observe that the SVM linear combination model is outperformed by simple keyword multiplication. This could be related to the noisy nature of the keyword probabilities computed using the Corel training set. It is possible that the greedy search approach turns out to be more robust in this case. We look forward to investigating this result in greater detail and would like to improve SVM accuracy by filtering out 'noisy' keywords using standard feature selection techniques.

We believe that in general there is further scope for applying text retrieval techniques to automatically annotated images to enhance users' experience with poorly labelled image collections. One intriguing application of our framework is that of blind relevance feedback for improving image recall. In this scenario, a small set of images is retrieved using sparsely available text metadata, and additional images are then retrieved using a combination of keyword probabilities that best describe the initial results. The robustness of our approach towards small numbers of positive examples, and its computational efficiency, could potentially support such application in near real time.

Finally, we would like to note that we have successfully applied our framework within a prototype search engine, which provides access to 1.14 million images spidered from the Web, based entirely on automatically assigned annotations[‖].

6. ACKNOWLEDGEMENTS

We would like to thank Daniel Heesch, Joao Magalhaes and Charanpal Dhanjal for very helpful comments and discussions of the subject. The first author is partially funded by the Overseas Research Scholarship award.

REFERENCES

1. P Duygulu, K Barnard, N de Fretias, and D Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of the European Conference on Computer Vision*, pages 97–112, 2002.
2. J Jeon, V Lavrenko, and R Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the ACM SIGIR Conference on Research and Development in Infrmation Retrieval*, pages 119–126, 2003.
3. V Lavrenko, R Manmatha, and J Jeon. A model for learning the semantics of pictures. In *Proceedings of the 16th Conference on Advances in Neural Information Processing Systems NIPS*, 2003.
4. S Feng, R Manmatha, and V Lavrenko. Multiple Bernoulli relevance models for image and video annotation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1002–1009, 2004.
5. A Ghoshal, P Ircing, and S Khudanpur. Hidden Markov models for automatic annotation and content based retrieval of images and video. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 544–551, 2005.
6. A Yavlinsky, E Schofield, and S Rüger. Automated image annotation using global features and robust nonparametric density estimation. In *Proceedings of the International Conference on Image and Video Retrieval*, pages 507–517, 2005.

[‖]http://www.beholdsearch.com

7. V Lavrenko, S Feng, and R Manmatha. Statistical models for automatic video annotation and retrieval. In *Proceedings of the IEEE ICASSP International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 17–21, 2004.
8. G Iyengar, P Duygulu, S Feng, P Ircing, S Khudanpur, D Klakow, M Krause, R Manmatha, H Nock, D Petkova, B Pytlik, and P Virga. Joint visual-text modeling for automatic retrieval of multimedia documents. In *Proceedings of the ACM International Conference on Multimedia*, pages 21–30, 2005.
9. Y Mori, H Takahashi, and R Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *Proceedings of the International Workshop on Multimedia Intelligent Storage and Retrieval Management*, 1999.
10. J Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, 1971.
11. S Robertson and K Sparck-Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 1976.
12. C Buckley, G Salton, J Allan, and A Singhal. Automatic query expansion using SMART: TREC 3. In *Text REtrieval Conference*, 1994.
13. M Mitra, A Singhal, and C Buckley. Improving automatic query expansion. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 206–214, 1998.
14. F Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.
15. M Naphade, T Kristjansson, and T Huang. Probabilistic multimedia objects (multijects): A novel approach to video indexing and retrieval in multimedia systems. In *Proceedings of the International Conference on Image Processing*, volume 3, pages 536–540, 1998.
16. M Naphade and T Huang. A probabilistic framework for semantic indexing and retrieval in video. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 475–478, 2000.
17. J Smith, M Naphade, and A Natsev. Multimedia semantic indexing using model vectors. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 445–448, 2003.
18. A Natsev, M Naphade, and J Smith. Exploring semantic dependencies for scalable concept detection. In *Proceedings of the International Conference on Image Processing*, volume 3, pages 625–628, 2003.
19. A Amir, W Hsu, G Iyengar, C-Y Lin, M Naphade, A Natsev, C Neti, H Nock, J Smith, B Tseng, Y Wu, and D Zhang. IBM research TRECVID-2003 video retrieval system. In *Proceedings of TRECVID*, 2003.
20. A Hauptmann, M Chen, M Christel, C Huang, and W-H Lin. Confounded expectations: Informedia at TRECVID 2004. In *Proceedings of TRECVID*, 2004.
21. Y Wu, B Tseng, and J Smith. Ontology-based multi-classification learning for video concept detection. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 1003–1006, 2004.
22. R Yan, M Chen, and A Hauptmann. Mining relationship between video concepts using probabilistic graphical model. In *To appear in the proceedings of IEEE International Conference on Multimedia and Expo*, 2006.
23. N Rasiwasia, N Vasconcelos, and P Moreno. Query by semantic example. In *Proceedings of the International Conference in Image and Video Retrieval*, pages 51–60, 2006.
24. T Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of the European Conference on Machine Learning*, pages 137–142, 1998.
25. V Vapnik. *The Nature of Statistical Learning Theory*. SpringerVerlag, 1995.
26. SVM^{light}. http://svmlight.joachims.org/.
27. L Ahn and L Dabbish. Labeling images with a computer game. In *Proceedings of the ACM SIGCHI conference on Human factors in computing systems*, pages 319–326, 2004.
28. TRECVID 2005. TREC video retrieval evaluation online proceedings. http://www-nlpir.nist.gov/projects/tv2005/, 2005.
29. T Volkmer, J Smith, A Natsev, M Campbell, and M Naphade. A web-based system for collaborative annotation of large image and video collections. In *Proceedings of the ACM International Conference on Multimedia*, pages 892–901, 2005.
30. H Tamura. Texture features corresponding to visual perception. *IEEE Transactions. Systems, Man and Cybernetics*, 8(6):460–473, 1978.

31. B Manjunath and W-Y Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8):837–842, 1996.
32. P Howarth and S Rüger. Evaluation of texture features for content-based image retrieval. In *Proceedings of the International Conference on Image and Video Retrieval*, pages 326–334, 2004.

(a) Keywords: orchid (1 keyword)

(b) Keywords: orchid, flora (2 keywords)

(c) Keywords: orchid, flora, insect, herd, closeup, leaves (6 keywords)

(d) Keywords: orchid, flora, insect, herd, closeup, leaves, flower, cat, blossoms, tree (10 keywords)

Figure 3. Results of the Web dataset "Flower" concept search using keywords provided by greedy multiplication model

(a) Keywords: stone (1 keyword)

(b) Keywords: stone, pebbles (2 keywords)

(c) Keywords: stone, pebbles, castle, remains, reflection, pattern (6 keywords)

(d) Keywords: stone, pebbles, castle, remains, reflection, pattern, structure, grass, detail, scenery (10 keywords)

Figure 4. Results of the TRECVID "People Marching" concept search using keywords provided by greedy multiplication model