KEYWORD-DETECTION APPROACH TO AUTOMATIC IMAGE ANNOTATION

Ville Viitaniemi and Jorma Laaksonen

Neural Networks Research Centre
Helsinki University of Technology
P.O.BOX 5400, FI-02015 TKK, Espoo, Finland
{ville.viitaniemi,jorma.laaksonen}@tkk.fi
fax +358-9-451 3277

Keywords: Automatic image annotation, mutual information, self-organising map

Abstract

In this paper we consider the problem of automatically annotating images with keywords. We first discuss performance measures for the problem in some length. We propose a new information-theory based measure – de-symmetrised mutual information (DTMI). We then describe a straightforward solution to the annotation problem. We first train a set of classifiers to detect the presence of each individual keyword in the set of training images. For this we use the PicSOM image analysis framework. We then describe a method of converting the classifier outputs back into keyword annotations for the test set. We compare the performance of the proposed method experimentally to that of other methods presented in the literature. For the experiments we use data from the Corel database. The result of the comparison is favourable to the proposed method.

1 Introduction

In recent times the problem of matching words and images has attracted considerable research interest [1, 2, 3, 6, 7]. This problem provides another point of view to the difficult problem of general image content understanding. Our own research originates from the direction of interactive content-based image retrieval (CBIR), for which we have been developing the PicSOM software system (e.g. [5]). In CBIR one of the main problems is bridging of the large semantic gap between low level image descriptors and the user's desire to query the systems using high level semantic concepts. Natural language, i.e. words, readily offers a symbolic representation of semantic concepts. Using textual annotations as proxy might offer some help in overcoming the semantic gap.

A wealth of automatic image captioning methods has been proposed in the literature. But how good are the methods? It is somewhat difficult to say since there is also a wide variety of performance measures in use. In this paper we intend to shed light into this issue by using the PicSOM methodology for image similarity assessment and derive an automatic captioning method in a most straightforward imaginable way from the assessments. We compare the performance of our proposed method against some published methods using a commonly used performance measure and data sets from the commonly used commercial Corel image database. In this way we get an idea of the level of the state of the art performance in the field since we then can directly relate it to the performance of the PicSOM system. On the other hand, this conceptually elementary image annotation method will provide an easily understandable baseline for future improvements.

In this paper, we will first discuss some performance measures in Section 2. Even though there is more than adequate number of various measures in use currently, we still feel that they are somehow unprincipled. Therefore, we introduce one more performance measure, inspired by information theoretic reasoning. In Section 3, we briefly describe the method for deriving classifiers for keywords. Then in Section 4 we describe how to use the classifier outputs for annotating images. In Section 5 we experimentally test our method with widely used Corel data sets. In Section 6 we draw final conclusions from the results.

2 Performance evaluation

The keyword annotation problem can be reworded as "maximising the goodness of the predicted annotations". The solution is thus inherently determined by the used "goodness" measure.

We will use notation where N and W denote the number of images and keywords in the test set, respectively. We describe the predicted and ground truth annotations with binary matrices $\mathbf{A}^{pred}$, $\mathbf{A}^{gt} \in \{0,1\}^{N \times W}$ where the columns corresponds to keywords and rows to the different images. We choose our performance measure to be a function of the two matrices $\mathbf{A}^{pred}$ and $\mathbf{A}^{gt}$ only. Thus, intuitively speaking, we want to define some sort of similarity measure of those two matrices.

2.1 Normalised score

A widely used performance measure in the literature is the average normalised score [1, 3, 6]

$$\text{NS} = E_I \left[\frac{c(I)}{w(I)} - \frac{n(I)}{W - w(I)} \right], \tag{1}$$

where E_I denotes the average over the test set images I, $w(I)$ is the actual annotation length, i.e. the number of keywords annotating an image, $c(I)$ is the number of correctly predicted keywords and $n(I)$ the number of incorrect predictions. Thus the NS measure can be written also in the form

$$\text{NS} = \frac{1}{N} \sum_i \sum_j \mathbf{A}_{ij}^{pred} \left(\frac{\mathbf{A}_{ij}^{gt}}{\sum_k \mathbf{A}_{ik}^{gt}} - \frac{1 - \mathbf{A}_{ij}^{gt}}{W - \sum_k \mathbf{A}_{ik}^{gt}} \right). \tag{2}$$

The measure has some intuitively appealing characteristics. First of all, it attains its maximum value iff the matrices are exactly the same. Similarly, the minimum value is attained iff the matrices are complements of each other. Furthermore, for a given number of correctly (incorrectly) predicted keywords the score decreases (increases) with the increasing number of incorrectly (correctly) predicted keywords.

The balance between sensitivity and specificity is somewhat arbitrary, though. In practice, given the current prediction accuracy and the value $W - w$ for the balance constant, maximising the NS performance leads to the prediction of large number of keywords for each image. This is because false positives are punished relatively mildly. Other ways of balancing specificity and sensibility (such as [2]) do exist, but in general, they do not appear any less arbitrary. If the annotation problem is restricted to predicting a fixed number of keywords (e.g. [7]), the balancing problem disappears, though.

The normalised score has its merits, however, and as we want to compare the performance of our method with other methods, we use also this often used performance measure in the experiments.

2.2 Mutual information measures

The normalised score does not capture the intuitive notion of some words being easier to predict than others. If almost every image is annotated with the word "sky", correctly predicting the word is not that difficult and should not be given much weight in the performance measure. The information-theoretic notion of *information* captures this distinction.

A natural candidate for a performance measure would be the mutual information. But to calculate this, we would need random variables and their distributions, as opposed to the two fixed matrices $\mathbf{A}^{pred}$ and $\mathbf{A}^{gt}$. In order to arrive at random variables, we consider each row of $\mathbf{A}^{pred}$ to be a realisation of random variable $\mathbf{a}^{pred}$, and for $\mathbf{A}^{gt}$ similarly. Now we could, in principle, estimate the distributions $p(\mathbf{a}^{gt})$ and $p(\mathbf{a}^{gt}|\mathbf{a}^{pred})$ and use their mutual information

$$I(\mathbf{a}^{gt}; \mathbf{a}^{pred}) = H(\mathbf{a}^{gt}) - H(\mathbf{a}^{gt}|\mathbf{a}^{pred}) \tag{3}$$

as a performance measure.

As a practical matter, this is not feasible as each of the random variables has 2^W possible values, and there would not be enough test data available to estimate the distributions. Therefore, we (somewhat incorrectly) factorise the distributions:

$$\begin{aligned} p(\mathbf{a}^{gt}) &\approx \prod_i p(\mathbf{a}_i^{gt}) & (4) \\ p(\mathbf{a}^{gt}|\mathbf{a}^{pred}) &\approx \prod_i p(\mathbf{a}_i^{gt}|\mathbf{a}_i^{pred}). & (5) \end{aligned}$$

Here we have used the notation

$$\mathbf{a}^{gt} = \left[\mathbf{a}_1^{gt} \ldots \mathbf{a}_W^{gt} \right], \ \mathbf{a}^{pred} = \left[\mathbf{a}_1^{pred} \ldots \mathbf{a}_W^{pred} \right]. \tag{6}$$

With this assumption of keyword independence, we can use the modified mutual information

$$\begin{aligned} I'(\mathbf{a}^{gt}; \mathbf{a}^{pred}) &= H'(\mathbf{a}^{gt}) - H'(\mathbf{a}^{gt}|\mathbf{a}^{pred}) \\ &= \sum_i H(\mathbf{a}_i^{gt}) - \sum_i H(\mathbf{a}_i^{gt}|\mathbf{a}_i^{pred}) \quad (7) \end{aligned}$$

as a performance measure.

This performance measure still has a shortcoming. Let the binary random variable X denote the presence of a keyword in the ground truth and Y the presence of the corresponding keyword in the prediction. The mutual information between X and Y is symmetric around the completely random behaviour, i.e. it only measures the deviation of the X, Y dependence from randomness, not the direction. Maximum mutual information is obtained even if the variables are of the opposite polarity, i.e. $Y = 0$ implies $X = 1$ and vice versa. To overcome this shortcoming, we modify the expression for the conditional entropy.

The mutual information $I(X; Y)$ measures the decrease in the optimal code length of variable X when the coder is given knowledge about the prediction Y. We want to augment the mutual information with the requirement that Y should be of the same polarity as X. We present two alternative methods of modifying the measure to this end. In the first alternative we modify the coding setup slightly and use the code length reduction in this setup as the performance measure. In this setting the coder just refuses to take the predicted keyword into account if the polarity of the prediction is wrong. Instead, the word will be coded using the unconditional empirical distribution. The reduction in code length is denoted *rectified de-symmetrised termwise mutual information* (DTMI$_0$):

$$\text{DTMI}_0(\mathbf{a}^{gt}, \mathbf{a}^{pred}) = \sum_i H(\mathbf{a}_i^{gt}) - \sum_i \hat{H}_0(\mathbf{a}_i^{gt}|\mathbf{a}_i^{pred}), \tag{8}$$

where

$$\hat{H}_0(X|Y) = \begin{cases} H(X|Y) & \text{if } p(X=1|Y=1) \geq \\ & \quad p(X=1|Y=0) \\ H(X) & \text{otherwise} \end{cases} . \quad (9)$$

This measure does not distinguish how seriously the predictions are of wrong polarity, i.e. are the dependencies of the $0 \rightarrow 1$ type only weak or nearly deterministic. As a second alternative we heuristically modify the mutual information to take this seriousness into account. The *de-symmetrised termwise mutual information* (DTMI) measure is based on the intuitive idea

$$\text{DTMI}(\mathbf{A}^{gt}, \mathbf{A}^{pred}) = -\text{DTMI}(\mathbf{A}^{gt}, \mathbf{1}_{N\text{x}W} - \mathbf{A}^{pred}), \quad (10)$$

i.e. the DTMI will possess an odd symmetry in the polarity change of the predictions. For correct polarity predictions the DTMI will coincide with the mutual information:

$$\text{DTMI}(\mathbf{a}^{gt}, \mathbf{a}^{pred}) = \sum_i H(\mathbf{a}_i^{gt}) - \sum_i \hat{H}(\mathbf{a}_i^{gt}|\mathbf{a}_i^{pred}), \quad (11)$$

where

$$\hat{H}(X|Y) = \begin{cases} H(X|Y) & \text{if } p(X=1|Y=1) \geq \\ & \quad p(X=1|Y=0) \\ 2H(X) - H(X|Y) & \text{otherwise} \end{cases} . \quad (12)$$

The DTMI$_0$ and DTMI measures (simply the DTMI measures from here on) have a sound information-theoretical backing. Furthermore, codelength interpretation gives them a practical meaning – it makes sense to measure how many bits the keyword predictions are able to shorten the optimal code for the annotations of the test set. It is also intuitive to require prediction of the keyword to make it more probable. The code length interpretation gives the measures a natural absolute scale. To obtain a relative measure, we can divide the DTMI values by the unconditional test set entropy $\sum_i H(\mathbf{a}_i^{gt})$.

The DTMI measure exhibits all the intuitively pleasing properties mentioned in the connection with the normalised score. In addition to these properties, mutual information has the symmetry property that the measure remains the same when both $\mathbf{A}^{pred}$ and $\mathbf{A}^{gt}$ are complemented. The same applies to the DTMI measures. The normalised score does not possess this property.

2.3 Relation of DTMI to KL divergence

In [1] a Kullback-Leibler (KL) divergence based measure

$$E_{\text{KL}}(\text{model}) = E_I \left[D_{\text{KL}}(p(w|I), q(w|I) \right] \quad (13)$$

is used to measure the fit between the obtained probabilistic model $q(w|I) = q(w|B)$ and the actual conditional word distribution $p(w|I)$. Here w is a word in the vocabulary, I is an image and B its feature representation. This starting point seems suspicious as the distributions measure the probability of observing keyword w if one keyword is picked from the annotations of image I. Better alternative would be to measure the absolute probability of the image being annotated with the keyword w (conditional to I). E_{KL} thus effectively assumes each image to be annotated with a fixed number of keywords. For example, this is not the case for the Corel data used in the experiments.

There is a fundamental difference between E_{KL} and the DTMI measures as E_{KL} is not a function of $\mathbf{A}^{gt}$ and $\mathbf{A}^{pred}$ alone, but a functional of the probabilistic models instead. Consequently, E_{KL} is not directly applicable to annotation methods whose results are not formulated as probabilistic models. On the other hand, the DTMI measures could be adapted to probabilistic models by replacing the values in the binary matrix $\mathbf{A}^{pred}$ by the probabilities $p(\mathbf{A}_{ij}^{pred} = 1|\text{model, training set})$. Then a model would have to be estimated for $p(\mathbf{a}_i^{gt}|\mathbf{a}_i^{pred})$. This should not present a great difficulty.

Another unfortunate property of E_{KL} is of a more practical nature. The measure requires knowledge about the conditional distribution $p(w|I)$, but only the matrix $\mathbf{A}^{gt}$ is available. The distribution can be estimated only very poorly as there is only one observation available – the corresponding row of the matrix. In [1] the distribution is taken to be the ML estimate $1/K$ for the words appearing on the row and zero otherwise. This leads to a measure that does not discriminate well between common and uncommon words.

For these reasons and because of the incorrect fixed annotation length assumption, we believe E_{KL} not to be well suited for measuring annotation quality, although it clearly gives an indication whether an annotation method is sensible at all.

3 PicSOM system for image classification

The PicSOM system is a framework for research on content-based image retrieval. A more detailed description of the framework can be found in e.g. [5]. As the name implies, PicSOM uses the Self-Organising Map (SOM) [4] as its basic image indexing method, although other clustering methods are also supported. The SOM is a powerful tool for exploring huge amounts of high-dimensional data. It defines an elastic, topology-preserving grid of points that is fitted to the input space. The distribution of the data vectors over the SOM forms a two-dimensional discrete probability density. From the same image data several qualitatively different distributions can be obtained by using different feature extraction techniques.

3.1 Multiple Self-Organizing Maps

The PicSOM system is fundamentally based on the simultaneous use of several parallel SOMs, trained with different feature

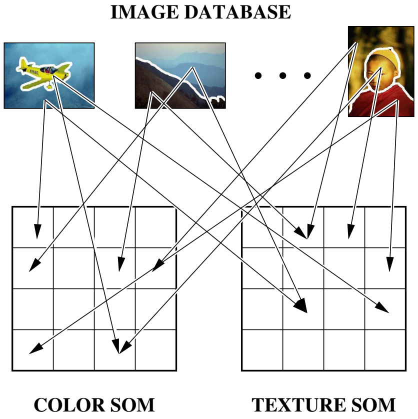

Figure 1. An example of using two parallel SOM indices for segmented images. The color and texture SOMs are trained with image segments and each segment is connected to its BMU on each SOM.

data, in image analysis. The features usually comprise of statistical visual descriptors such as the MPEG-7 content descriptors. The SOM feature indices can be constructed either for whole images or certain subobjects, such as image segments.

After the unsupervised training of the SOMs, the map units form data-driven representations of the feature spaces. Images and their segments are connected to SOM units that thus become their representantations on the SOMs. The connecting is done by locating the best-matching unit for each image or segment on each SOM. As a result, the different SOMs impose different similarity relations on the images and the system thus inherently uses multiple features for image analysis. An illustration with two parallel SOMs trained for image segments is presented in Figure 1.

3.2 Image similarity assessment

In the image classification task the system is provided with a number of training example images. The images are partitioned into two classes: they either do or do not possess some property. In this case, the class determining property is the existence of a certain keyword (e.g. “sky”) in the annotation of an image. In the following we denote the images belonging to the class relevant, the other images are called non-relevant. In the first phase the relevance classifications are propagated to the segments of the respective images. As the next step, the SOM units are awarded a positive score for every relevant image segment mapped in them resulting in an attached positive impulse. Likewise, the non-relevant segments result in negative scores. By normalising separately the positive and negative scores to sum to unity, we obtain a zero-sum sparse value field on every SOM in use.

Due to the topology preservation of the SOM, segments that are similar according to a specific visual feature are located near each other on the corresponding SOM surface. Therefore, we are motivated to perform spatial kernel smoothing on the relevance information. Figure 2 illustrates how the positive and negative responses are first mapped on a SOM surface and then expanded in the convolution.

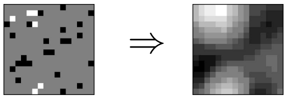

Figure 2. An example of a SOM surface being kernel smoothed. On the left, relevant and non-relevant segments of images are shown with white and black marks, respectively. On the right, the convolution result, where relevance information is spread around the centers.

Because of the normalisation the response values of the parallel SOMs are mutually comparable. The total qualification value for each image segment can then be obtained by simply summing the corresponding responses. For images, their segment-wise values are further summed to form the image-level qualification values. On SOMs that correspond well to the keyword annotations concentrated positive scores amplify each other. On worse SOMs, nearby positive and negative scores interfere destructively. The parallel content descriptors and indices do not therefore need explicit weighting.

4 Converting classifier outputs to annotations

Using the training set, the PicSOM system is trained to act as a classifier and to output a discrimination value that reflects the likelihood of a given image to be associated with an individual keyword. A separate classifier is trained to discriminate each keyword in the vocabulary. We apply each classifier to the test set images and call the resulting list of discrimination values the classifier outputs.

We require from the classifier outputs that among the values output by a single classifier, increasing values are associated with larger likelihoods of the corresponding keyword. We do not require the values from different classifiers to be comparable. We assume the classifier outputs to be similar in the test and training sets, i.e. similar classifier output values for a training set and a test set image correspond to approximately equal likelihood of the keyword.

The annotation procedure processes each keyword separately. For each keyword, the classifier outputs for the test set are sorted and a number of top-scoring images are annotated with the keyword. The score threshold is selected so that a performance measure is maximised in the training set. In the following we will consider the selection of the optimal threshold for the NS and DTMI measures. In both cases the threshold can be selected to be the score of one of the images that are annotated with the keyword in the training set (positive example image). For the performance measures under consideration the thresholds for different keywords do not interact. We can thus select a threshold separately for each keyword classifier.

4.1 Normalised score

We compute the expected training set performance s_{NS} for each of the candidate thresholds, the expectation being over the empirical distribution of the annotation length w in the training set images. If the threshold was the score of the i:th positive image, the expected normalised score would be

$$s_{\mathrm{NS}}(i) = E_w\left[\frac{i}{w} - \frac{\mathrm{rank}(i) - i}{W - w}\right], \tag{14}$$

where $\mathrm{rank}(i)$ is the position of i:th positive image in the sorted classifier output. The threshold is set to the score of the positive image maximising this performance measure.

The approximation introduced by taking the expectation over the annotation lengths w could be avoided since we know the exact number of annotations for each training set image. By doing the approximation we lose the information of different keywords having different distributions of number of words in the annotations they appear in. On the positive side, we now have enough data to robustly estimate the global annotation length distribution, as opposed to small number of samples to estimate the distributions for the individual keywords. The tests we performed indicated that the effect of the approximation to s_{NS} is minor.

4.2 Termwise mutual information measures

The threshold selection procedure is similar to the normalised score case. The same threshold values can be used for optimising both the DTMI measures as for the correct-polarity predictions the measures coincide. For the training set the performance maximising threshold always results in a correct-polarity prediction. For this reason, we can calculate simply the mutual information.

In this case we evaluate a keyword's contribution to the train set performance exactly. We ignore the unconditional entropy of the training set since it is constant for all thresholds. The DTMI score for the threshold being at i:th example image is then

$$\begin{aligned} s_{\mathrm{DTMI}}(i) &= -\sum_y p_{\mathrm{emp}}(Y{=}y|i)\sum_x p_{\mathrm{emp}}(X{=}x|Y{=}y,i) \\ &\quad \times \log_2 p_{\mathrm{emp}}(X{=}x|Y{=}y,i). \end{aligned} \tag{15}$$

Here X and Y denote the presence of a keyword in the annotation in the training set ground truth and prediction, respectively. p_{emp}'s are the empirical probabilities in the training set:

$$\begin{aligned} p_{\mathrm{emp}}(Y=1|i) &= \frac{\mathrm{rank}(i)}{N} && (16)\\ p_{\mathrm{emp}}(X=1|Y=1,i) &= \frac{i}{\mathrm{rank}(i)} && (17)\\ p_{\mathrm{emp}}(X=1|Y=0,i) &= \frac{N_{\mathrm{pos}} - i}{N - \mathrm{rank}(i)}. && (18) \end{aligned}$$

5 Experiments

5.1 Data sets

For testing we use subsets of the Corel image database, which has often been used in the experiments in the literature, e.g. [1, 2, 3, 6, 7]. Unfortunately, many of these experiments use performance measures different from ours. Anyway, we will reproduce the settings of some of the experiments. In the Corel database the images are equipped with keyword annotations. For the sake of comparison we use the preprocessing of the database supplied by [1] where the annotations are filtered to contain only keywords that appear frequently enough.

For visual description of the images we use segmentations and features given in [1]. The segmentations are produced with the *normalised cuts* algorithm [8]. At most ten largest segments are kept for each image. The visual content of the segments is described by a set of 40 features, some of which are highly redundant. The features include position, area, three elementary shape descriptors, averages and standard deviations in RGB, CIE L*a*b* and RSG colour spaces, and 16 measures of texture.

The first of the used image subsets is exactly the same as in [3] and contains 9883 images. This image set is divided, similarly, in the training set of 6961 and the test set of 2922 images. We denote this data set the Glotin data. The keyword vocabulary is slightly preprocessed, mostly to remove inconsistencies in the keywords. The resulting vocabulary size is 267 words.

In [1] and others, somewhat smaller subsets of the Corel image database are used. We run the experiments for two of these subsets (subsets 006 and 008 in [1]). The training set sizes for the subsets are 5192 and 5266 images, the test set sizes 1737 and 1724 images. The vocabulary is limited to contain only keywords that occur frequently enough in the training data. The vocabulary sizes are 162 and 168 words. We denote these data sets the Barnard data. There are two reasons for the use of these data sets. First of all, we can directly compare the performance of our system against the published figures of the other methods on this data set. Secondly, we can evaluate the effect of slightly different image sets on performance and thus get a better understanding of the comparability of various published results.

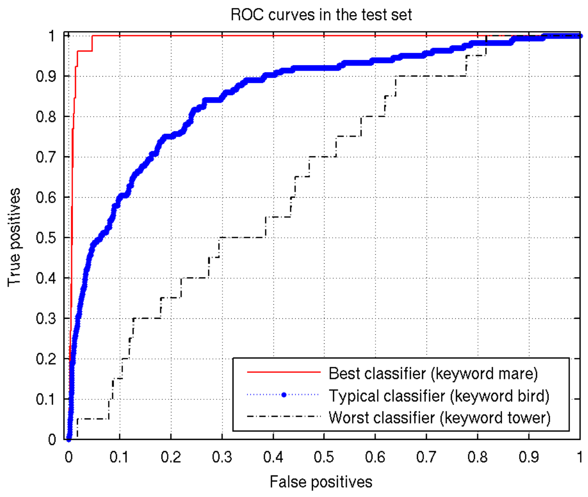

Figure 3. ROC curves of some classifiers in the test image set. The performance of the classifiers is evaluated using the area-under-curve (AUC) criterion. Keywords appearing less than 20 times in the test set were excluded from this comparison. The typical classifier was selected by binning the AUC values and selecting a classifier from the most typical bin.

5.2 Classifier training

For classifier training the features were divided into ten partly overlapping sets. The feature values were normalised so that the standard deviations of the features in a set were roughly equal. For each set of features a SOM presentation was trained in an unsupervised manner. Both test and training sets were used to train the SOM's. We consider this to be justifiable, as we are not using the labels of the test set, only the statistics of the test inputs. This is not even likely to affect the results in this case, as the statistics of the test and the training set are very similar.

We then use the standard PicSOM machinery to train classifiers for each keyword. We use all the training images annotated with the relevant keyword as set of positive example images, and all the other training images as the set of negative examples. We did not do extensive search to select the best combination among the available features to train the classifiers. We chose the features that performed best in the experiments [3] as a starting point and did some local search in that neighbourhood. In addition to starting feature set of shape and CIE L*a*b* features, the final feature sets also included position and some of the texture features. It is not likely that we have found the optimally performing set of features, neither are the other classifier parametres likely to be optimal. However, the performance of the resulting classifiers seems reasonably good. Figure 3 shows the receiver operating characteristic (ROC) curves of the best, the worst and a typical classifiers for not too infrequent keywords for a certain set of classifier training parametres.

5.3 Results

Even though we have above questioned the justification for using the NS score in the evaluation of annotation performance, we use it for the comparison of various existing methods for the reason that it simply is readily available for many methods.

Figure 4 shows the NS score resulting from always predicting a fixed number of keywords with highest *a priori* probabilities, regardless of the actual pictorial content. This can be regarded as the baseline method to beat for a method to be useful at all. We see that the different data sets have different levels of *a priori* performance. However, the higher values are probably mainly explained by the fact that in the larger databases the additional keywords mostly appear in the infrequent keyword end of the vocabulary. Due to their low probability, the main effect of these keywords is the increase in the vocabulary size. This directly increases the NS scores by reducing penalty from false positives. On the other hand, the thresholds for predicting keywords in the optimal annotation are also lowered and the number of correctly predicted keywords increases. In the figure we can also see the magnitude of the effect of vocabulary preprocessing in the Glotin data.

The DTMI measures express the information gain from the annotation if the statistics of the keywords are already known. Any uniform annotation thus has the DTMI score zero. The relative “difficulty” of the data sets can be compared by the unconditional average entropies of the annotations, which are shown in the Tables 1 and 2. For comparison, the average termwise entropy for the Glotin data without vocabulary preprocessing is 15.55 bits. However, neither these numbers nor the NS scores of the prior distributions are able to give any hint on how much of the remaining uncertainty can realistically be predicted away by exploiting the image-word correlations. They merely indicate what kind of performance can be achieved by looking at the word statistics alone and ignoring the pictorial content. One can think of example cases with the same unconditional entropies (or *a priori* NS scores) where the image features would be either fully correlated with the keywords (e.g. the keywords graphically written in the image) or fully uncorrelated with the keywords (e.g. noise images).

It appears that the performance comparisons of annotation methods working on different data sets are going to be unreliable in any case. We thus try to avoid such comparisons here by employing several data sets. In [3] it is proposed that the effect of different data sets could be partly equalised by measuring the percentual NS improvement over fixed *a priori* annotations. This would also be used to equalise the performance measure of methods predicting a different number of keywords for the images. This equalisation method seems flawed. Implicit to the idea is the notion that it would be easier to improve over a good *a priori* performance than bad. Intuition tells otherwise. E.g. the improvement of 0.1 over *a priori* performance 0.4 is percentually larger than 0.1 improvement over 0.9, even though the predictor is doing a perfect job in the latter case. Also our annotation results for somewhat different but still rea-

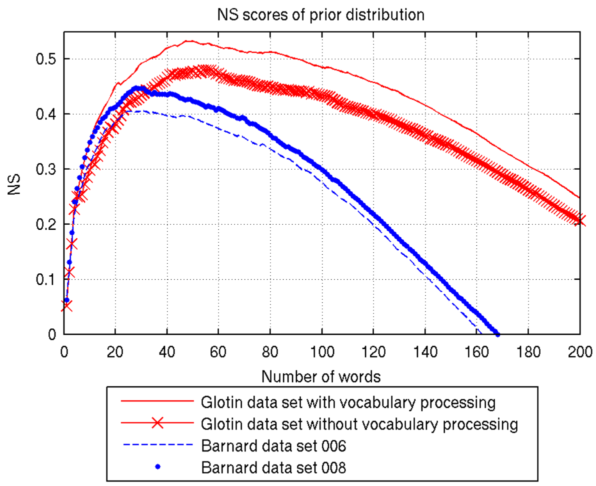

Figure 4. Normalised scores of fixed length annotations according to maximum training set *a priori* probability for various data sets.

sonably similar data sets point into this direction. The improvements ΔNS tend to get smaller as the *a priori* performance gets better. Also the equalisation of annotation lengths by this way appears unfair as there is no reason to assume the proportion remains (even approximately) the same for the same prediction method with a different number of predicted keywords. In fact, operating curves of the predictors in [1, 6] would hint the contrary.

Tables 1 and 2 show the prediction results for the Glotin and Barnard data, respectively. For our annotation methods, the subscripts NS and DTMI refer to predictors that are optimised for the respective measures. From [1, 3, 6], the best performing methods have been presented in the tables.

The inclusion of the DIMATEX method for the Glotin data is not fully fair, as the method considers the problem of finding a fixed number of best keywords for each image. The number of keywords, ten, is clearly too low to give the highest possible NS score for this data. On the other hand, not even the Corel data itself has a fixed number of keywords for each image. Limiting the number of keywords in the annotations is a sensible goal by itself. However, if such a goal is desired, we think that the goal should be incorporated in the performance measure in order to be able to compare predictions with varying number of keywords.

Method	ΔNS	NS$_{\text{prior}}$	DTMI	$\sum_i H(\mathbf{a}_i^{gt})$
PicSOM$_{\text{NS}}$	**0.182**	0.533	2.23	14.97
PicSOM$_{\text{DTMI}}$	-0.016	0.533	**3.20**	14.97
DIMATEX [3]	**0.092***	0.348	**0.71**	14.97

Table 1. Results of annotating the Glotin data set. For the PicSOM methods, NS$_{\text{prior}}$ denotes the maximum of the NS score of fixed annotations over the number of predicted words. ΔNS denotes the NS increase over this number. The DIMATEX result (*) is for ten word prediction length and FLAB features. Thus, for sake of comparison, on the DIMATEX line NS$_{\text{prior}}$(*) is calculated for a fixed ten word annotation.

Method	ΔNS	NS$_{\text{prior}}$	DTMI	$\sum_i H(\mathbf{a}_i^{gt})$
PicSOM$_{\text{NS,006}}$	**0.262**	0.406	2.95	14.66
PicSOM$_{\text{DTMI,006}}$	0.115	0.406	**3.78**	14.66
PicSOM$_{\text{NS,008}}$	**0.213**	0.448	2.56	14.27
PicSOM$_{\text{DTMI,008}}$	0.070	0.448	**3.26**	14.27
Monay et al. [6]	**0.153**	0.383	-	-
Barnard et al. [1]	**0.179**	0.425	-	-

Table 2. Results of annotating the Barnard-like data sets. NS$_{\text{prior}}$ denotes the maximum of the NS score of fixed annotations over the number of predicted words. ΔNS denotes the NS increase over this number. The two sets of PicSOM results are for the two data sets 006 and 008. For [1, 6] the results are averages of ten and nine data sets, respectively. The result of Monay et al.is for their method LSA 2. The method of Barnard et al. has identifier binary-D-2-region-cluster in [1].

For our methods the DTMI and DTMI$_0$ figures are the same up to two decimal places. Thus, the effect of predictions of wrong polarity is negligible. In this case, using the normal definition of mutual information would not introduce large errors in performance measurement. However, for the sake of completeness and as a safeguard against prediction polarity, we consider DTMI to be the safer choice for this application.

For optimising the DTMI and NS performance measures it is worthwhile to predict more keywords than appear in the true annotations. The optimal number of excess keywords depends on the quality of predictions. The current quality of classifiers results in the optimal number to be around 40, whereas the number of keywords in the correct annotations is approximately 3.5. In general, optimising the NS measure requires somewhat more keywords to be predicted in our case than DTMI, 42 vs. 32 on average for the Glotin data set. The behaviour on individual keywords is dramatically different for the two methods. The NS measure encourages the prediction of the most common keywords much more often than DTMI. For rare keywords the opposite is the case.

In our approach the classifiers that result in good performance in sense of DTMI seem to have good performance also in the NS sense. Relative order of slightly differently performing classifiers may vary in the DTMI and NS senses. The similarity is quite natural since the classifier output determines the ordering of images in terms of likelihood of those images being associated with a keyword. Optimising the annotation in either NS or DTMI sense affects only the choice for threshold, i.e. the number of top images in the ordering that will be tagged with the keyword.

6 Discussion and conclusions

The image auto-annotation problem can always be seen as an optimisation problem. The choice of the criterion to be optimised will guide the development of the annotation methods into a certain direction. We argue that the information-theoretically inspired DTMI measure is a good and well-grounded candidate for such a criterion.

The quality of the annotations in the Corel database is not very good. The keywords that annotate an image are usually correct, but the main problem appears to be the missing annotations. For some of the keywords (e.g. “sky”,”snow”,”wall”), the decision to include the keyword in the annotations seems rather arbitrary, even though the corresponding objects appear in prominent positions in the images. It is questionable, whether the goodness of an automatic annotation system should be measured in terms of predicting such peculiarities of the manual annotation process.

It would appear that the shortcomings of the database annotation start to seriously affect the performance evaluation only after the performance reaches a certain level. Based on the experimental evidence, the image-word correlations are by far the dominating effect also in the Corel database and the peculiarities of the annotations are only a second order effect. As long as the performance gain potential left in the “real” target of image-word correspondence dominates the secondary effects, it makes sense to use the database as a performance indicator. The problem is that it is very difficult to know how good is the limiting performance level in this database. We know assumption-free learning to be impossible [9]. On the other hand, with strong enough assumptions the learning problem becomes trivial. The question then is, how good performance can be achieved in the Corel database when exploiting reasonably general assumptions. We predict that the current results are still far below the maximum level. The performance can thus be usefully measured on this database as long as some caution is taken.

The achieved annotation results demonstrate that a relatively well performing auto-annotation system can be constructed in a straightforward manner given classifiers for the individual keywords. This provides a potential application for the kind of image similarity assessments the PicSOM system produces. Improvements in the classification performance will be directly converted to an increased performance in image auto-annotation. We have a number of ideas to this end. For example, in this paper we used the segmentations and features provided by [1]. Preliminary experiments indicate that some other feature extraction techniques could lead to better performance. On the other hand, there is a lot of room for improvement in the classifier building techniques themselves.

Another source for performance improvement is the procedure by which the classifiers are selected and their outputs converted into annotations. In this study we have assumed the different keywords to be independent. The assumption is not exactly correct. The annotation performance can probably be improved by taking the keyword interdependency into account. However, it appears likely that many of such dependencies would be quite strongly due to the specific annotation practices of the used database. In any case, the matter seems worth investigating.

In choosing the prediction thresholds we have employed the naïve assumption that the training set perfectly reflects the statistics of the test set. Thus we have ignored all small sample effects, and probably compromised some performance. It remains unknown how much performance could be gained by taking these effects into account. The effects are most pronounced for infrequent keywords and therefore the effect for a single keyword may be small. However, the number of such rare keywords is large.

Acknowledgements

We want to specially thank Sabrina Tollari and Hervé Glotin for providing access to their data and results in electronic form. Thanks also to Kobus Barnard et al. for setting their data publicly available. This work was supported by the Academy of Finland in the projects *Neural methods in information retrieval based on automatic content analysis and relevance feedback* and *New information processing principles*, the latter being part of the Finnish Centre of Excellence Programme 2000–2005.

References

[1] K. Barnard, P. Duygulu, N. Freitas, D. Forsyth, D. Blei, and M. I. Jordan. Matching words and pictures. *Journal of Machine Learning Research, Special Issue on Machine Learning Methods for Text and Images*, 3:1107–1135, February 2003.

[2] J. Fan, Y. Gao, H. Luo, and G. Xu. Automatic image annotation by using concept-sensitive salient objects for image content representation. In *Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 361–368, Sheffield, England, July 2004.

[3] H. Glotin and S. Tollari. Fast image auto-annotation with visual vector approximation clusters. In *Proc. of IEEE EURASIP Fourth International Workshop on Content-Based Multimedia Indexing (CBMI2005)*, June 2005.

[4] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer-Verlag, third edition, 2001.

[5] J. Laaksonen, M. Koskela, and E. Oja. PicSOM—Self-organizing image retrieval with MPEG-7 content descriptions. *IEEE Transactions on Neural Networks, Special Issue on Intelligent Multimedia Processing*, 13(4):841–853, July 2002.

[6] F. Monay and D. Gatica-Perez. On image auto-annotation with latent space models. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 275–278, Berkeley, CA, 2003.

[7] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. GCap: Graph-based automatic image captioning. In *Proceedings MDDE ’04, 4th International Workshop on Multimedia Data and Document Engineering*, Washington, DC, USA, July 2004.

[8] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, August 2000.

[9] D. H. Wolpert. The supervised learning no-free-lunch theorems. In *Proc. 6th Online World Conference on Soft Computing in Industrial Applications*, 2001.