FAST IMAGE AUTO-ANNOTATION WITH VISUAL VECTOR APPROXIMATION CLUSTERS

Hervé Glotin and Sabrina Tollari

Université du Sud Toulon-Var
Laboratoire LSIS - Equipe INCOD
F-83957 La Garde cedex, France
{glotin, tollari}@univ-tln.fr

ABSTRACT

The paper proposes a simple novel technique to automatically determine a set of keywords that describe the content of an image. The images are segmented in 'blobs', which are approximatively classified using discretized features space. This results in a small number of visual Vector Approximation Clusters (VAC), which allows to train the joint probability table of the visual features and the textual annotations from a training data set. Futhermore a simple Bayes model is used to determine the probability that a keyword describes a test image. The paper includes an experimental evaluation on COREL database. We compare our approach with state of the art auto-annotation methods using the same database, words set and scoring method. Results show that our simple method give similar results than state of the art models.

Keywords: image auto-annotation, CBIR, Vector Approximation Clusters, split entropy, COREL, word prediction, classification, high-dimensional problem.

1. INTRODUCTION

The need for efficient content-based image retrieval has increased in many application areas such as biomedicine, military, and Web image classification and searching. The problem is a highly important issue in MM IR. Many approaches have been devised and discussed over more than a decade. While the technology to search text has been available for some time, the one to search images and videos is much more challenging. Most of the image content based retrieval systems require the user to give a query based on image concepts, but in general people would like to construct semantic queries using textual descriptions. Some systems aim to enhance image word research using visual information [18], anyway one needs a fast system that robustly auto-annotates large un-annotated image databases.

The general idea of auto-annotation systems is to associate a class of 'similar' images with similar keywords, which reduces the problem to index a new image to the task of determining its class. It has been pursued in various approaches, such as neural networks, statistical classification etc. The presented approach is somewhat similar but very simple. The paper presents a novel approach to image indexing. The images are submitted to a feature analysis process, resulting for each image in a segments (called blobs) set. This induces a partition of the image set. Then each blob is approximatively classified using a thresholded discretized features space. This results in a small number of visual Vector Approximation Clusters (VAC), which allows to train the joint probability table of the visual features and the textual annotations from a training data set. The main advantage of the VAC method is that it dramaticaly reduces the time and memory cost of any scaning algorithm of the high-dimensional visual space. Then from a training set of images annotated with keywords a correlation between image approximated classes and keywords is derived. Thus, identifying the classes of each blob of a new image allows to assign the keywords attributed to this class as well. The originality of the model is to use a fast visual vector approximation clustering and a simple naive bayes approach, but to generate competitive auto-annotation compared to state of the art algorithms. We call our model DIMATEX (for Dichotomic IMAge TEXt annotation).

Experiments are driven on the COREL database: COREL images samples with their reference manual annotations are shown in figure 1.COREL lexicon has about 250 different words. Our model is trained on 7000 labelled images, and tested on 3000 images. Results of correct annotation are presented for different thresholded visual features. Two types of experiments are made: the first one (E1) involves a visual space clustering which is stable over all the visual dimensions. The second experiment (E2) involves an adaptive threshold clustering which splits each dimension according to the maximization of local feature information gain.

Experiments prove the viability of our approach relatively to other methods tested on the same database: we

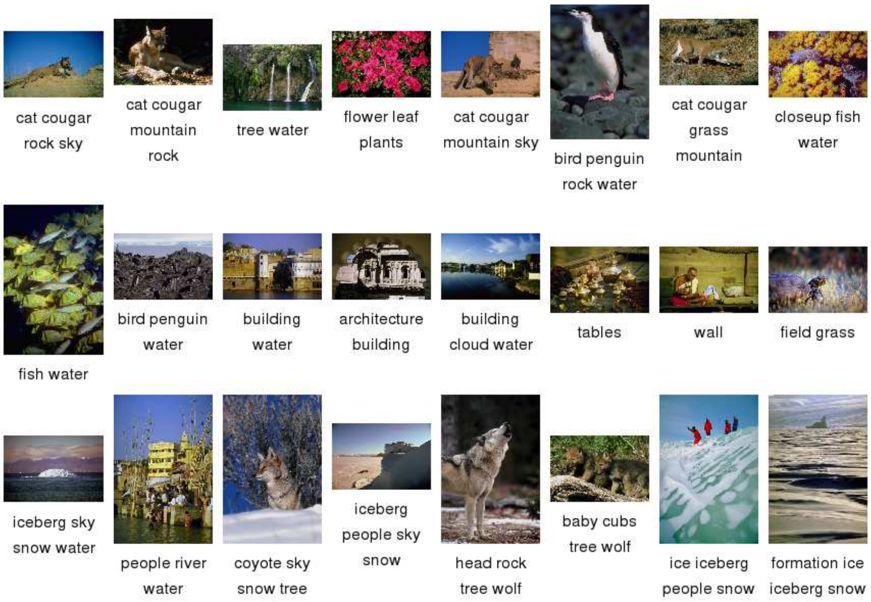


Fig. 1. Image examples from COREL database with their manual annotation. Lexicon is about 250 different words.

show the relatively good accuracy of the system and its potential in linguistic indexing of photographic images. Results show a better word prediction (between 25% to 50% of gain of correct words) compared to direct priors word emission model, for an annotation generating from 1 to 10 words. These are similar results than the state of the art models running on the same database.

The first section of this paper presents related work. The second section presents our method of dichotomic clustering of each segment of each image. The third section gives the simple bayesien model which predicts the keywords of an image using the joint distribution of textual and visual random variables. In the next section, experiments done on COREL allow fair comparisons to state of the art methods, showing that DIMATEX gives similar performance to other algorithm. We detail commun auto-annotation results with [9] in the last section, and we discuss of DIMATEX improvements in the concluding section.

2. RELATED WORK

One major issue in content based information retrieval on visual database is the high dimension of the visual space, leading to heavy word prediction models. One of these is the multi-modal extension to Latent Dirichlet Allocation (LDA), used in [3], in combination with Hofmann's hierarchical clutering/aspect model and translation model. Other models use an AHC algorithm, or a Latent Semantic Analysis models [9, 4]. Another kind of a heavy model recently developed is the two-dimensional multiresolution hidden Markov models (2-D MHMMs) [7]. Some other works are using relevance-based language models [6]. This system is quite robust, but still needs smoothed maximum likelihood estimates, which can yield to a long training stage.

On the other hand, some recent techniques have been proposed for the creation of index of very large database: the Vector Approximation file (VA-file) approach (see [19] and [4] for a good introduction). VA-file aims to reduce the number of dimension for a given index problem by a vector approximation. Recently VA-file has been extended to relevance feedback retrieval systems [13]. Indeed, many data partitioning index methods perform poorly in high dimensional space and do not support relevance feedback retrieval. The VA-File approach overcomes some of the difficulties of high dimensional vector spaces, but cannot be applied to relevance feedback retrieval using kernel distances in the data measurement space. Heisterkamp and Peng introduce kernel VA-File that extends VA-File to kernel-based retrieval methods. A key observation is that kernel distances may be non-linear in the data measurement space but is still linear in an induced feature space. It is this linear invariance in the induced feature space that enables KVA-File to work with kernel distances. An efficient approach to approximating vectors in an induced feature space is presented in [13] with the corresponding upper and lower distance bounds. Thus an effective indexing method is provided for kernel-based relevance feedback image retrieval methods.

In this paper, we propose a method inspired from kernel VA-file allowing to build words and visual features joint distributions table, after image segmentation and approximative segments clustering using discretized features space.

3. VISUAL VECTOR APPROXIMATION CLUSTERS

Trying to tackle with the high-dimension visual space problem, DIMATEX splits each visual feature in two classes. Human visual classification has been shown to be robust to such hard features classification [11]. Therefore, we propose to build dichotomic clusters, based on a discriminant threshold applied on each visual feature. The relatively low number of possible clusters allows to build a direct codebook containing the joint probability of visual and textual random variable, leading to a simple and fast auto-annotation system.

Let I be an image. Let $\{w_1, \cdots, w_n\}$ be the keywords set associated with the document I. Each image can be segmented into different visual segments (called blobs) b_i which belong to the set of segments $\{b_1, \cdots, b_m\}$ of the image I. Each segment b can have different kinds of physicals characteristics (such as textures, forms, colours...) which can be represented by a vector of D dimensions.

We can suppose that values for each visual dimension are limited. So that, we can split each dimension into y intervals. We number each interval from 0 to $y-1$. It's easy now to classify a visual segment b, one just needs to find the approximate vector of b in the visual vector. Finally, there are y^D possible clusters and the algorithm just needs $(y-1)*D$ comparisons to classify a blob. The cluster name of a segment is then the concatenation of the interval numbers for that segment. For example, let visuals characteristics be red, green and blue ($D=3$). Each dimension values are in $[0, 255]$. We split all visual space into two equal intervals ($y=2$): interval 0 $[0, 127]$, interval 1 $[128, 255]$. The vector $v_1 = \{10, 212, 198\}$ is in the cluster named 011.

We will show that low visual space dimension gives enough good results allowing fast and efficient systems. If the cluster of a test image is no more present in the model, then one can search for the nearest cluster and merge it in.

In order to generate dichotomic cluster on the visual space, one can simply split in two equal demi-space a visual space dimension with a stable threshold value $\theta = 0.5$. This value is closed to the mean over all words and all features, of the threshold values where the features probability density functions given labelled or not by a word are getting inversed. Futur theoretical studies will formalize this criterion. This method is called E1 in the experiment section.

A second method (E2) involves an adaptive threshold clustering which splits each dimension according to the maximization of local feature information gain. Indeed, the optimal choice of the discriminant value of 2 logical clusters is given in [14] and [11]. If a threshold θ_{dim} splits a data set χ into two subsets χ^- and χ^+, then the split entropy $e_{split}(\chi, \theta_{dim})$ is the sum of the two sub-entropies weighted by the frequency of the respective subsets:

$$e_{split} = \frac{|\chi^-|}{|\chi|} e(\chi^-) + \frac{|\chi^+|}{|\chi|} e(\chi^+). \qquad (1)$$

The discriminating power of a threshold can then be measured as the difference between the two entropy values, before and after split. This is called the information *gain*:

$$gain(\theta_{dim}) = e(\chi) - e_{split}(\chi, \theta_{dim}) \qquad (2)$$

and represents the reduction in the quantity of information needed to describe the class labels of the data. This method will be labelled E2 in the experiment section and will be compared to E1.

4. THE DIMATEX AUTO-ANNOTATION MODEL

Let C be a collection of un-annotated images. Each $I \in C$ is represented by a set of blobs generated as explained in the experiment section: $I = \{b_1, \cdots, b_m\}$.

Therefore the images are submitted to a feature analysis process, resulting in a vector of segments (or blobs) for each image: for the given set of pixels representing I, all b_i generate a partition.

In this section we develop a formal model that allows us to automatically assign meaningful keywords to an un-annotated image I.

We assume that there exists a training collection T of annotated images, where each image $J_j \in T$ has a dual representation in terms of both words and blobs:

$$J = \{b_1, \cdots, b_m; w_1, \cdots, w_n\}, \qquad (3)$$

where $\{w_1, \cdots, w_n\}$ represents the words in the image caption. Note that m and n may differ from image to image, because we do not assume that there is an underlying one to one alignment between the blobs and the words in an image, as in [6, 2, 9].

4.1. Training the joint visual-textual distribution

DIMATEX models as much simple as possible the joint probability of observing the word w and the blobs $b_1, \cdots, b_m$ in the same image. Thus, for each word w, and each blob b_i belonging to its unique visual cluster c_k, we have:

$$P(w, c_k|T) = P(w, b_i|T). \qquad (4)$$

Because all images J_j of T make a uniform partition of T, we have $P(J_j|T) = \frac{1}{|T|}$ and because there is a one to one relation between visual cluster c_k of b_i and b_i, we have:

$$\begin{array}{rcl} P(w, c_k|T) & = & P(w, b_i|T) \\ & = & \sum_j P(w, b_i|J_j, T) P(J_j|T) \\ & = & \frac{1}{|T|} \cdot \sum_j P(w, b_i|J_j, T) \\ & = & \frac{1}{|T|} \cdot \sum_j P(w|J_j, b_i, T) P(b_i|J_j, T). \end{array} \qquad (5)$$

We set $P(w|J_j, b_i, T) = 1$ if w annotates J_j, 0 else. We estimate $P(b_i|J_j, T)$ as the relative surface of b_i over all blobs of J_j. Thus:

$$P(b_i|J_j, T) = \frac{area(b_i)}{\sum_{l=1}^{m} area(b_l)} \quad (6)$$

where $area(b_i)$ is the number of pixels of b_i in J_j of T.

4.2. Using DIMATEX for image auto-annotation

Using previous joint distributions, one can estimate the most accurate words for the auto-annotation of each image. We need to estimate $P(w|I)$ for every word w in the vocabulary. The probability of drawing the word w is best approximated by the conditional probability of observing w given that we previously observed $b_1, \cdots, b_m$. We produce automatic annotation for new images by simply picking a desired number of words that have the highest probability under $P(w|I)$ and use those words for the annotation.

As $\{b_1, \cdots, b_m\}$ is a partition of I, we calculate the probability that word w annotated I knowing T:

$$P(w|I, T) = \sum_{i=1}^{m} P(w|b_i, I, T) P(b_i|I, T) \quad (7)$$

where $P(b_i|I, T)$ is estimated by:

$$P(b_i|I, T) \simeq P(b_i|I) = \frac{area(b_i) \in I}{\sum_{l=1}^{m} area(b_l) \in I}. \quad (8)$$

If the associated visual cluster of b_i is c_k then:

$$P(w|b_i, I, T) = P(w|c_k, I, T) \simeq \frac{P(w, c_k|T)}{P(c_k|T)} \quad (9)$$

where $P(w, c_k|T)$ is calculated using the trained joint visual-textual distribution sets by (5).

4.3. Scoring

As in [3], we measure the system performance using the Normalised Score. We allow the model to predict K words, $0 < K < 11$. Then, the score E_{model} is the Normalized Score [2, 9]:

$$NS = \frac{right}{h} - \frac{wrong}{N - h} = sensibility + specificity - 1 \quad (10)$$

where $right$ is the number of correct predicted words, $wrong$ is the number of wrong predicted words, h is the number of words in the reference, N is the vocabulary size.

In all results reported for segmentation, feature choice, and region merging, we express word prediction relative to that for the empirical word distribution i.e. the frequency table for the words in the training set. Let be E_{priors} this empirical model score, then we display results as the relative gain to the prior model based to the empirical word density:

$$gain = 100 * \frac{E_{model} - E_{priors}}{E_{priors}}. \quad (11)$$

This measure reduces variance due to varied test sample difficulty.

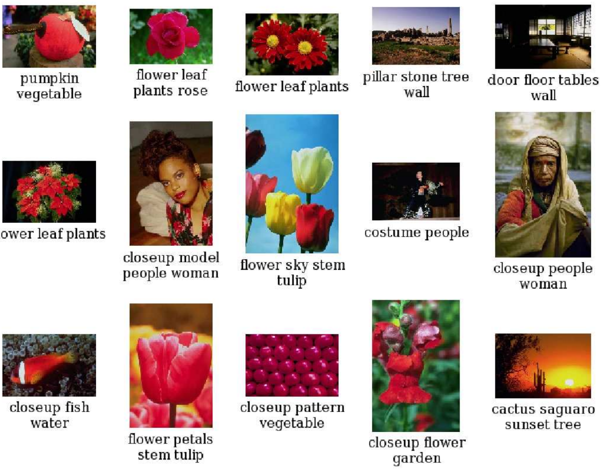


Fig. 2. Some training images which have one segment in the cluster 0000110110001 for FLABT and experiment E1.

5. EXPERIMENTS

5.1. Corpus

The corpus is COREL database [12]. For our experiments, it is made of 10000 images with approximately 100000 segments. Each image is labelled by an average 3.6 words by image and has an average of 10 visual segments. In order to make fair comparisons, we use the same data and the same features, as previous state of the art experiments (Computer Vision Group of University of California (Berkeley) and Computing Science Department of University of Arizona as described in [2, 1]). Each image is segmented using normalized cuts [15]. This segmentation method has the tendency to produce small or unstable regions. Thus, only the 10 largest regions in each image are selected. For this database, the order of any region is defined as its decreasing size rank.

Each region is described by a set of 46 features. Size is represented by the portion of the image covered by the region. 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. Colour is represented using the average and standard deviation of (R,G,B) and (L,a,b) over the region. In the case of RGB, the 3 extracted bins are: S=R+G+B, r=R/S, g=G/S,

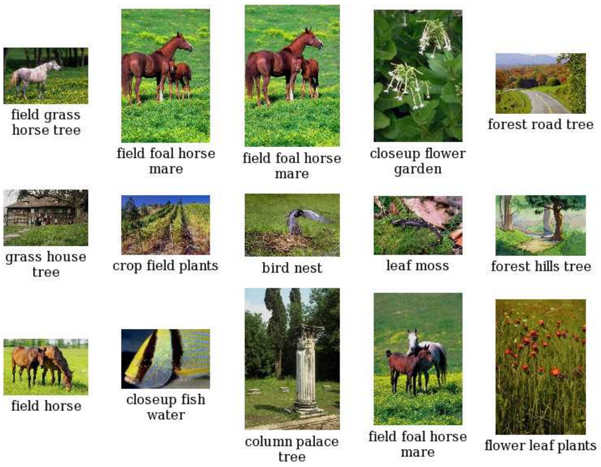


Fig. 3. Some training images which have one segment in the cluster 0111010110000 for FLABT and experiment E1. Images 2 and 3 are the same because two segments of this image are in this cluster.

named rgS. Texture and shapes are extracted as in [2]. Authors has chosen these features to be computable for any image region, and be independent of any recognition hypothesis.

5.2. Clustering

First, we normalized the features so that the distribution of each features on the train set is between 0 and 1 for 90% of the data. We use for this pre-processing an MLE fitting to gamma distribution (using matlab statistic toolbox) and we shift the distribution interval into the interval $(0, 1)$. Then, we mean four by four the 16 dimensions of texture to obtain 4 average textures. Finally, from the 46 features of the data, we keep only 3 dimensions of forms (F), the 6 dimensions of LAB and the 4 dimensions of mean textures(T).

We make experiments for LAB only (LAB, dim=6), for forms and LAB (FLAB, dim=9) and for forms, LAB and texture (FLABT, dim=13).

Features	LAB	FLAB	FLABT
Number of dimensions	6	9	13
Number of possible clusters	64	512	8192
Number of actual clusters	64	502	4419

Tab. 1. Resume of experiment E1.

In first experiment $(E1)$, we set $\theta_{dim} = 0.5$ for each dimension in order to have comparison results (see table 1). In second experiment $(E2)$, we split each dimension in two intervals according to the gain of information as explain in the end of section 3. We obtain 2^D possible clusters, combination of each dichotomic clusters. According to these clusters, we classify and cumulate the TRAIN set segments in the matrices of the joint distribution of words and visual random variables. Theses matrices are also called codebook. We obtain less number of clusters than the possible number of clusters, because some clusters are empty. Figures 2, 3 and 4 show some training images from two different clusters.

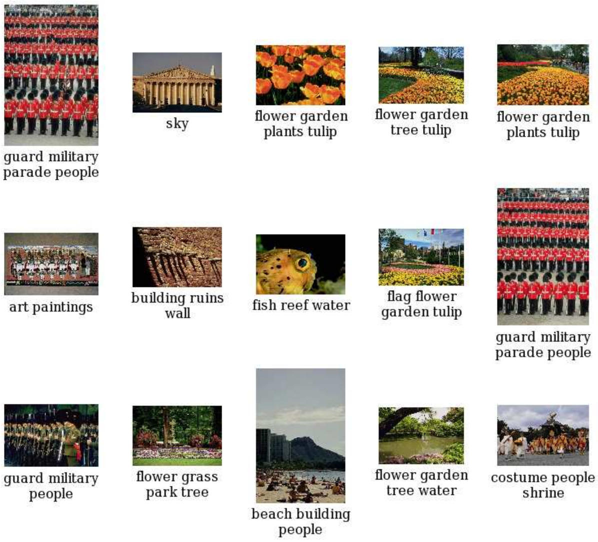


Fig. 4. Some training images from cluster 1111111111110 for FLABT and experiment E1.

5.3. Auto-annotation results

Experiments are conducted on 3000 images on COREL in the TEST set, and 7000 images in the TRAIN set. In order to make fair comparison using the gain measure we used similar sets as in [2, 9]. The results of correct annotation keywords by extracting the n best predicted words, are given for different feature dimensions and for original visual and textual joint distribution in figure 5.

The 6 dimensions refers to LAB only, 9 to F and LAB, 13 to F, LAB and T. The confidence interval is around 0.05 %. Results show the evidence that results with 10 best words are better than the one with 5 best ones. As a COREL annotation has in average 3,6 words, we compute the average of DIMATEX for FLAB with less that 4 words, and with more that 4. Then we compare in figure 6 the different gain measures of the state of the art model on the same data base [2, 9].

Average results on COREL database demonstrate a word prediction by DIMATEX between 25 and 50% of correct words for an annotation of 1 to 10 words, which is a performance similar to the state of the art much more complex models running on the same database.

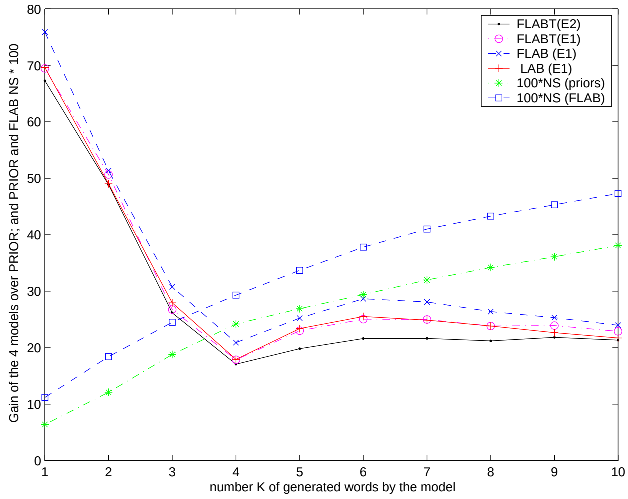


Fig. 5. Gain measures of DIMATEX model on the TEST SET, compared to a prior model, for different visual space features FLABT, FLAB, and LAB. Curves are givenfor E1 or E2 spliting methods. We plot also $100 \times NS$ measures for prior model and FLAB model which helps to explain the shapes of the gain curves. The curves shows that E1 performs better that E2, and that FLAB always out performs other features.

We show in figure 7 some test image sample with their original annotation versus the DIMATEX auto-annotation. We see that the model provides some words which are not in the original annotation, but could be appropriately attached to the image. This denotes the ability for DIMATEX to somehow handle ‘polysemy’.

6. DISCUSSION AND CONCLUSION

Exceeding the empirical density performance is required to demonstrate non-trivial learning. Doing substantially better than this on the Corel data is difficult. The annotators typically provide several common words (e.g. ‘sky’, ‘water’. ‘people’), and fewer less common words (e.g. ‘tiger’). This means that annotating all images with, say, ‘sky’, ‘water’, and ‘people’ is quite a successful strategy. Thus for this data set, the increment of performance over the empirical density is a sensible indicator. We see in figure 7 that DIMATEX give similar results than other algorithm. DIMATEX is quite fast: anotating 1000 images takes less than five minutes on a PIV bi-processor machine (without features extractions stage).

The fact that E2 is not improving results may come from the local feature optimisation of the threshold specific in each dimension using the split entropy. Actually, if one set the threshold to 0.5, it can reach by chance a better global optimisation clustering over all visual clusters on all visual features space. That is why we observe better result for E1 than for E2. Future work will consist in finding optimal threshold values across all features at once. We will study in detail this issue in futher studies on features pdf.

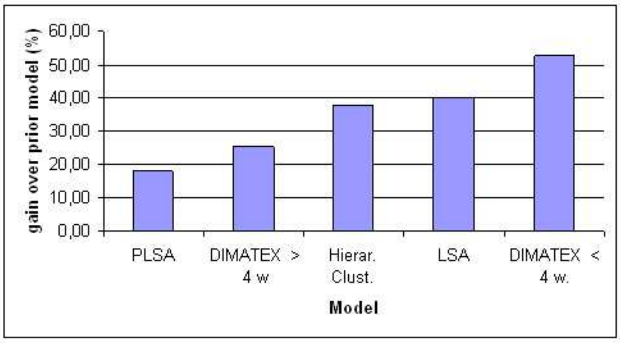


Fig. 6. This figure sumarises the gain results of DIMATEX over a ‘prior model’, and the gains for different state of the art model on the same task and the same database. LSA (Latent Semantic Analysis) [9], PLSA (Probabilistic LSA) [9], Hierarchical Clustering [2] and mean of DIMATEX less than 4 words and more or equal to 4 words prediction.

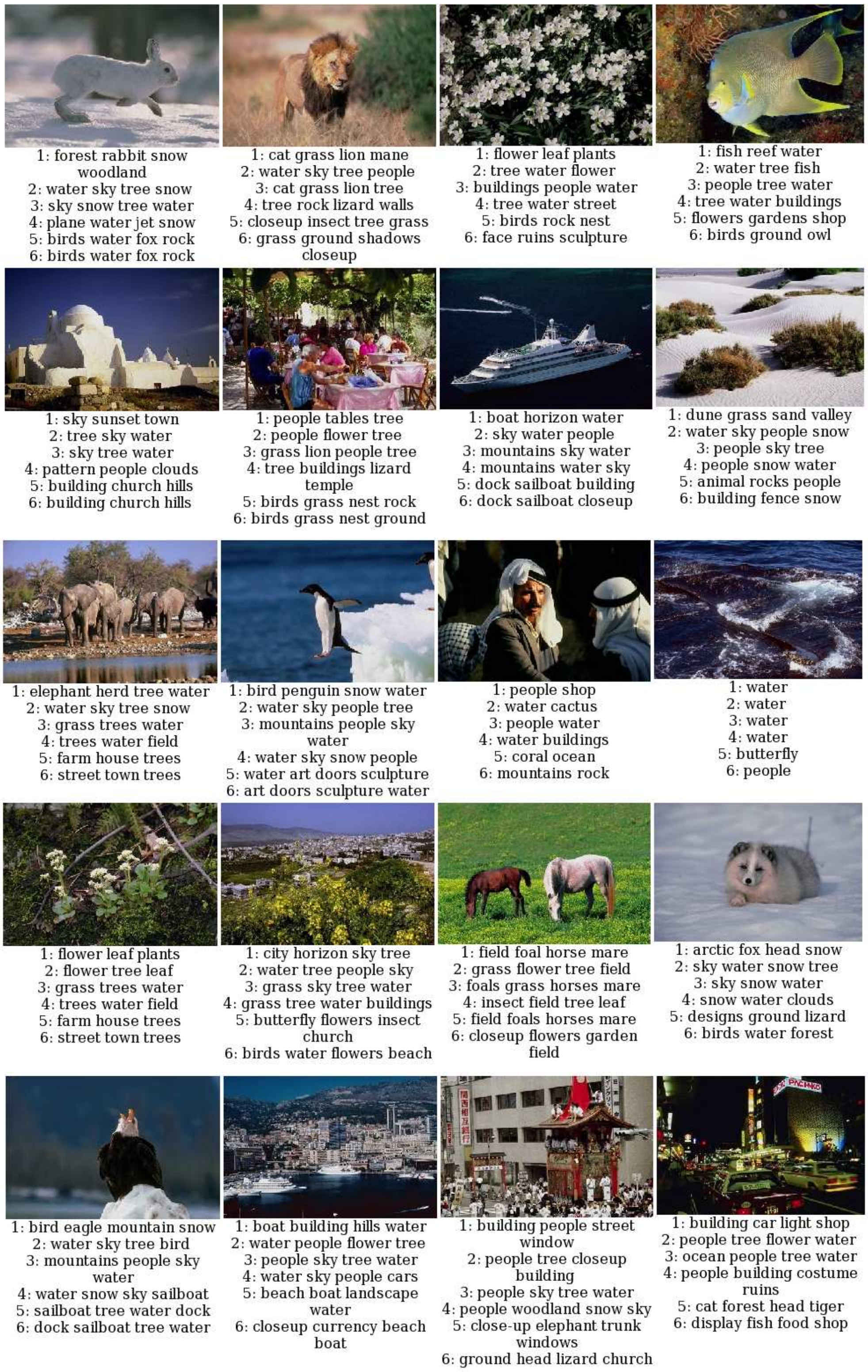
We demonstrated a simple visuo-textual mapping system which generates image auto-annotation with 25% to 50% correct words on a large reference database. According to the first experiment, correct annotation does not increases with the dimension features, so we could expect that the systems will perform better with a pre-processing discriminant features analysis.

Finally, according to current experiments, one can expect a high compression of the codebook size (reducing by 10 times), by applying the mutual information maximization, which generates new annotation errors. That would lead to a faster model. One could also try to get an adaptive number of emitted words using a threshold on the probabilities as in [9]. Another amelioration of DIMATEX will be tested running in a first stage a Factorial Discriminant Analysis on the visual feature set depending on each word [5, 17], in order to reduce the visual space. Indeed good results have been obtained on a ACH clustering systems [16] on the same database (gain of +37% for a reduction by 4 of the visual space dimension [17]).

7. REFERENCES

[1] K. Barnard. http://vision.cs.arizona.edu/kobus, 2003. Data used for “Matching Words and Pictures” [2].

[2] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. I. Jordan. Matching words and pictures. In *Journal of Machine Learning Research*, volume 3, pages 1107–1135, 2003.



1: forest rabbit snow woodland
2: water sky tree snow
3: sky snow tree water
4: plane water jet snow
5: birds water fox rock
6: birds water fox rock

1: cat grass lion mane
2: water sky tree people
3: cat grass lion tree
4: tree rock lizard walls
5: closeup insect tree grass
6: grass ground shadows closeup

1: flower leaf plants
2: tree water flower
3: buildings people water
4: tree water street
5: birds rock nest
6: face ruins sculpture

1: fish reef water
2: water tree fish
3: people tree water
4: tree water buildings
5: flowers gardens shop
6: birds ground owl

1: sky sunset town
2: tree sky water
3: sky tree water
4: pattern people clouds
5: building church hills
6: building church hills

1: people tables tree
2: people flower tree
3: grass lion people tree
4: tree buildings lizard temple
5: birds grass nest rock
6: birds grass nest ground

1: boat horizon water
2: sky water people
3: mountains sky water
4: mountains water sky
5: dock sailboat building
6: dock sailboat closeup

1: dune grass sand valley
2: water sky people snow
3: people sky tree
4: people snow water
5: animal rocks people
6: building fence snow

1: elephant herd tree water
2: water sky tree snow
3: grass trees water
4: trees water field
5: farm house trees
6: street town trees

1: bird penguin snow water
2: water sky people tree
3: mountains people sky water
4: water sky snow people
5: water art doors sculpture
6: art doors sculpture water

1: people shop
2: water cactus
3: people water
4: water buildings
5: coral ocean
6: mountains rock

1: water
2: water
3: water
4: water
5: butterfly
6: people

1: flower leaf plants
2: flower tree leaf
3: grass trees water
4: trees water field
5: farm house trees
6: street town trees

1: city horizon sky tree
2: water tree people sky
3: grass sky tree water
4: grass tree water buildings
5: butterfly flowers insect church
6: birds water flowers beach

1: field foal horse mare
2: grass flower tree field
3: foals grass horses mare
4: insect field tree leaf
5: field foals horses mare
6: closeup flowers garden field

1: arctic fox head snow
2: sky water snow tree
3: sky snow water
4: snow water clouds
5: designs ground lizard
6: birds water forest

1: bird eagle mountain snow
2: water sky tree bird
3: mountains people sky water
4: water snow sky sailboat
5: sailboat tree water dock
6: dock sailboat tree water

1: boat building hills water
2: water people flower tree
3: people sky tree water
4: water sky people cars
5: beach boat landscape water
6: closeup currency beach boat

1: building people street window
2: people tree closeup building
3: people sky tree water
4: people woodland snow sky
5: close-up elephant trunk windows
6: ground head lizard church

1: building car light shop
2: people tree flower water
3: ocean people tree water
4: people building costume ruins
5: cat forest head tiger
6: display fish food shop

Fig. 7. Some annotations automatically generated by DIMATEX and other systems. Manual annotations from COREL (1). Annotations automaticaly generated by DIMATEX (2). Annotations automaticaly generated by PLSAWORDS (3), PLSAWORDSFEATURES (4), DIRECT (5), and LSA (6) methods [10], extract from [8].

[3] Kobus Barnard, Pinar Duygulu, Raghavendra Guru, Prasad Gabbur, and David Forsyth. The effects of segmentation and feature choice in a translation model of object recognition. *Computer Vision and Pattern Recognition*, pages 675–682, 2003.

[4] Sid-Ahmed Berrani, Laurent Amsaleg, and Patrick Gros. Recherche par similarités dans les bases de données multidimensionnelles : panorama des techniques d'indexation. *Ingénierie des systèmes d'information (RSTI série ISI-NIS)*, 7(5-6):65–90, 2002.

[5] Hervé Glotin, Sabrina Tollari, and Pascale Giraudet. Approximation of linear fisher discriminant analysis for adaptive word dependent visual feature sets improving image auto-annotation. In *Proc. of Advanced Concepts for Intelligent Vision Systems (ACIVS2005), submitted*, Antwerp, Belgium, september 2005.

[6] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 119–126, 2003.

[7] Jia Li and James Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1075–1088, 2003.

[8] Monay. http://www.idiap.ch/ monay/acmm04, 2004.

[9] Florent Monay and Daniel Gatica-Perez. On image auto-annotation with latent space models. In *Proc. ACM Int. Conf. on Multimedia (ACM MM)*, pages 275–278, 2003.

[10] Florent Monay and Daniel Gatica-Perez. Plsa-based image auto-annotation: constraining the latent space. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 348–351, New York, NY, USA, 2004. ACM Press.

[11] Luis Miguel Moreira. *The use of boolean concepts in general classification contexts*. Thèse de doctorat, École polytechnique fédérale de Lausanne (EPFL), Lausanne, 2000.

[12] Henning Muller, Stéphane Marchand-Maillet, and Thierry Pun. The truth about corel - evaluation in image retrieval. In *The Challenge of Image and Video Retrieval (CIVR02)*, 2002.

[13] Jing Peng and Douglas R. Heisterkamp. Kernel indexing for relevance feedback image retrieval. In *Proc. of IEEE International Conference on Image Processing (ICIP-2003)*, pages 733–736, 2003.

[14] J. Ross Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., 1993.

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

[16] Sabrina Tollari. Filtrage de l'indexation textuelle d'une image au moyen du contenu visuel pour un moteur de recherche d'images sur le web. In *Actes d'ACM Confrence en Recherche d'Informations et Applications (CORIA'05)*, Grenoble, France, mars 2005.

[17] Sabrina Tollari and Hervé Glotin. Sélection adaptative des descripteurs visuels et usage de métadescripteurs contextuels dépendant du mot-clé pour l'indexation automatique d'images. In *Actes d'Atelier Métadonnées et Systèmes d'Information (MetSI'05) lié à INFORSID2005*, Grenoble, France, mai 2005.

[18] Sabrina Tollari, Hervé Glotin, and Jacques Le Maitre. Enhancement of textual images classification using segmented visual contents for image search engine. *Multimedia Tools and Applications*, 25(3): 405–417, march 2005.

[19] Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proc. of 24th Internationale Conference of Very Large Data Bases (VLDB)*, pages 194–205, 24–27 1998.