IDIAP RESEARCH REPORT

A Discriminative Approach for the Retrieval of Images from Text Queries

David Grangier [1] Florent Monay [2]
Samy Bengio [3]

IDIAP–RR 06-15

March 2006

[1] IDIAP, CP 592, 1920 Martigny, Switzerland, grangier@idiap.ch
[2] IDIAP, CP 592, 1920 Martigny, Switzerland, monay@idiap.ch
[3] IDIAP, CP 592, 1920 Martigny, Switzerland, bengio@idiap.ch

IDIAP Research Institute **www.idiap.ch**
Rue du Simplon 4 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

IDIAP Research Report 06-15

A Discriminative Approach for the Retrieval of Images from Text Queries

David Grangier Florent Monay Samy Bengio

March 2006

Abstract. This work proposes a new approach to the retrieval of images from text queries. Contrasting with previous work, this method relies on a discriminative approach: the parameters are selected in order to minimize a loss related to the ranking performance of the model, i.e. its ability to rank the relevant pictures above the non-relevant ones when given a text query. In order to minimize this loss, we introduce an adaptation of the recently proposed *Passive-Aggressive* algorithm. The generalization performance of this approach is then compared with alternative models over the *Corel* dataset. These experiments show that our method outperforms the current state-of-the-art approaches, e.g. the average precision over *Corel* test data is 21.6% for our model versus 16.7% for the best alternative, Probabilistic Latent Semantic Analysis.

1 Introduction

Several organizations, such as advertising companies or publishers, need tools to efficiently access and organize large collections of pictures. For instance, Getty Images proposes to its customers to browse and search more than 30 million images. This paper focuses on one of the tools needed by such organizations: a system that retrieves pictures from text queries. Given a picture collection P and a text query q, the goal of such a system is to rank the pictures of P such that the pictures relevant to q appear above the others. In order to perform such a ranking, a scoring function F which assigns a real value $F(q,p)$ to any picture/query pair (p,q) is used: given a query q, the pictures of P are ranked by decreasing scores.

In the ideal case, such a function F would always rank relevant pictures above non-relevant ones, i.e. F would satisfy,

$$\forall q, \forall p^+ \in R(q), \forall p^- \notin R(q), F(q,p^+) - F(q,p^-) > 0, \tag{1}$$

where $R(q)$ is the set of pictures relevant to query q.

In the following, we propose a learning procedure to identify a scoring function close to this ideal property, relying on a set of training data D_{train}. For that purpose, we first introduce a parameterized function F_w and a loss $L(F_w, D_{train})$ related to (1). A *Passive-Aggressive* (Crammer et al., 2003) approach is then adopted to identify the parameter vector w^* which minimizes $w \rightarrow L(F_w, D_{train})$. This model is referred to as Passive-Aggressive Model for Image Retrieval (PAMIR) in the following.

The proposed model contrasts with previous approaches that mostly rely on generative models and likelihood maximization (Barnard et al., 2003; Jeon et al., 2003; Monay & Gatica-Perez, 2004). In fact, the optimization of a loss related to the final retrieval performance is a key aspect of PAMIR. Our experiments over the *Corel* data show the advantage of this discrimative approach (see Section 5) and PAMIR is reported to outperform various models, such as Cross Media Relevance Model, CMRM (Jeon et al., 2003), Cross Media Translation Table, CMTT (Pan et al., 2004), or Probabilistic Latent Semantic Analysis, PLSA (Monay & Gatica-Perez, 2004). For instance, when the *SIFT* features are employed (see Section 3), PAMIR yields 16.0% average precision which should be compared to 12.3% for PLSA, the best alternative (see Section 5).

The remainder of this paper is organized as follows: Section 2 introduces PAMIR, Section 3 presents the features extracted to represent texts and images, Section 4 briefly describes the related work and Section 5 reports the experiments and results. Finally, Section 6 draws some conclusions.

2 The PAMIR Model

In this section, we first introduce the notation used, we then describe the parameterization of F_w and the loss $L(\cdot,\cdot)$, we finally explain how the *Passive-Aggressive* learning algorithm is applied.

2.1 Notation

In the following, we face two types of data: pictures and texts. Both of them are represented as vectors. The picture vector space is referred to as $\mathcal{P}$ while the text vector space is referred to as $\mathcal{T}$. Before describing our model, it should further be added that $\mathcal{T}$ is a subset of $\mathbb{R}^T$, where T is the vocabulary size. The i^{th} component of a vector $t \in \mathcal{T}$ is referred to as the weight of term i in text t. A detailed description of both text and picture representations is given in Section 3.

2.2 Model Parameterization

The parameterization of PAMIR is inspired by approaches developed for text retrieval, i.e. the task of retrieving *text* documents from *text* queries. In this case, documents are generally ranked with respect to their inner product with the submitted query (Baeza-Yates & Ribeiro-Neto, 1999). In other words,

the scoring function is

$$F^{text} : \mathcal{T} \times \mathcal{T} \rightarrow \mathbb{R}, \text{ where } F^{text}(q,d) = \sum_{i=1}^{T} q_i \cdot d_i. \quad (2)$$

We would like to adopt a similar approach to assign a score $F(q,p)$ to any pair (q,p) consisting of a text query $q \in \mathcal{T}$ and a picture $p \in \mathcal{P}$. For that purpose, we first introduce a mapping $f_w : \mathcal{P} \rightarrow \mathcal{T}$ that assigns a text vector $f_w(p) \in \mathcal{T}$ to any picture $p \in \mathcal{P}$ and we then compute the score of any query/picture pair (q,p) as,

$$F_w(q,p) = F^{text}(q, f_w(p)).$$

In the following, we restrict ourselves to mappings f_w of the form,

$$f_w : \mathcal{P} \rightarrow \mathbb{R}^T \text{ where } f_w(p) = (w_1 \cdot p, \ldots, w_T \cdot p)$$

and $w = (w_1, \ldots, w_T) \in \mathcal{P}^T$.

2.3 Ranking Loss

As mentioned in the introduction, we would ideally like to identify the parameters w such that F_w verifies all constraints in (1). However, we are only given a finite training set,

$$D_{train} = ((q_1, p_1^+, p_1^-), \ldots, (q_n, p_n^+, p_n^-)),$$

where for all k, q_k is a text query (i.e. $q_k \in \mathcal{T}$), p_k^+ is a picture relevant to q_k (i.e. $p_k^+ \in R(q_k)$) and p_k^- is a picture non-relevant to q_k (i.e. $p_k^- \notin R(q_k)$). Hence, we would like to select w relying on D_{train} data such that F_w ensures good generalization performance. In other words, w should be chosen such that F_w is likely to satisfy the constraints (1) for unseen data. For that purpose, a first approach would be to identify F_w such that all training constraints are satisfied, i.e.

$$\forall k, \quad F_w(q_k, p_k^+) - F_w(q_k, p_k^-) > 0. \quad (3)$$

However, to ensure better generalization, we propose to select w such that,

$$\forall k, \quad F_w(q_k, p_k^+) - F_w(q_k, p_k^-) \geq \epsilon_k \quad (4)$$

where $\forall k, \epsilon_k > 0$. This equation can then be rewritten as,

$$\begin{aligned} &\forall k, \quad l(w; (q_k, p_k^+, p_k^-, \epsilon_k)) = 0, \\ \text{where} \quad &l(w; (q_k, p_k^+, p_k^-, \epsilon_k)) = \\ &\qquad \max\left\{0, \epsilon_k - F_w(q_k, p_k^+) + F_w(q_k, p_k^-)\right\}. \end{aligned}$$

This means that for all k, we would like the score $F_w(q_k, p_k^+)$ to be greater than $F_w(q_k, p_k^-)$ by at least a *margin* of ϵ_k. This *margin* criterion is all the more attractive in our case since it has been applied to different text retrieval tasks with success, e.g. (Joachims, 2002).

In the following, the choice of ϵ_k is performed according to two alternative policies. First, we set all ϵ_k to 1, i.e. $\forall k, \epsilon_k = 1$, which is the strategy generally adopted for classification SVM (Joachims, 2001). This choice is referred to as *constant-*ϵ. Second, in the case where each training picture is further associated with a text caption, i.e. a set of keywords describing the main objects in a picture, we also introduce the following strategy,

$$\forall k, \quad \epsilon_k = \max(\epsilon, F^{text}(q_k, c_k^+) - F^{text}(q_k, c_k^-)),$$

where ϵ is a positive number and (c_k^+, c_k^-) refers to the captions of the pictures (p_k^+, p_k^-). This means that, in this case, the difference $F_w(q_k, p_k^+) - F_w(q_k, p_k^-)$ should at least be as high as the difference outputted by a text IR system relying on manually-produced captions. This second strategy is referred to as *caption-*ϵ in the following.

2.4 Passive-Aggressive Training

The *Passive-Aggressive* (PA) algorithm is an iterative minimization algorithm (Crammer et al., 2003). In our case, we apply this algorithm to minimize the loss

$$L(w; D_{train}) = \sum_{k=1}^{n} l(w; (q_k, p_k^+, p_k^-, \epsilon_k)). \tag{5}$$

For that purpose, we iteratively construct a sequence of weight vectors $(w^0, \ldots, w^m)$. The first vector is set to be zero, $w^0 = 0$. At the i^{th} iteration of the algorithm, the weight w^i is selected according to the i^{th} training example and the previous weight w^{i-1},

$$w^i = \arg\min_w \frac{1}{2}\|w - w^{i-1}\|^2 + C \cdot l(w; (q_i, p_i^+, p_i^-, \epsilon_i)). \tag{6}$$

This means that, at each iteration, we select the weight w^i as a trade-off between minimizing the loss on the current example $l(w; (q_i, p_i^+, p_i^-, \epsilon_i))$ and remaining close to the previous weight vector w^{i-1}. The *aggressiveness* parameter C controls this trade-off. It can be shown (Crammer et al., 2003) that the solution of problem (6) is

$$\begin{aligned} & w^i = w^{i-1} + \tau_i v_i, \\ \text{where} \quad & \tau_i = \min\left\{C, \frac{l(w^{i-1}; (q_i, p_i^+, p_i^-, \epsilon_i))}{\|v_i\|^2}\right\} \\ \text{and} \quad & v_i = -(q_1(p_k^+ - p_k^-), \ldots, q_T(p_k^+ - p_k^-)). \end{aligned}$$

At the end of the iterative process, the best weight among $\{w^0, \ldots, w^m\}$ is selected according to some validation data D_{valid}, i.e.

$$w = \arg\min_{w \in \{w^0, \ldots, w^m\}} L(w; D_{valid}).$$

The hyperparameter C has also been selected to maximize the performance over D_{valid}. The proof that the above procedure actually minimizes the loss (5) is given in (Crammer et al., 2003).

3 Text and Picture Representations

This section describes the representations used for text and images.

3.1 Text Representation

As mentioned before, textual data are represented with vocabulary-sized vectors, e.g. a query q will be assigned the vector

$$q = (q_1, \ldots, q_T),$$

where q_i is the weight of term i in the query q and T is the vocabulary size. This type of vector is often referred to as *bag-of-words* vector since this representation does not take word ordering into account. In our case, the term weights correspond to the popular $tf \cdot idf$ representation with Euclidean normalization (Baeza-Yates & Ribeiro-Neto, 1999), i.e. given $t \in \mathcal{T}$,

$$t_i = \frac{tf_{i,t} \cdot idf_i}{\sqrt{\sum_{j=1}^{T} (tf_{j,t} \cdot idf_i)^2}}$$

where the term frequency $tf_{i,t}$ corresponds to the number of occurrences of term i in t and the inverse document frequency idf_i is defined as $idf_i = -log(r_i)$, r_i being the fraction of training picture captions containing term i. This weighting scheme is referred to as *tf-idf-norm* in the following.

3.2 Image Representation

Similarly to previous work focussing on image retrieval from keywords (see Section 4), discrete image features have been used. Two alternative types of features have been extracted from the images: *blobs* and *Scale Invariant Feature Transform* (SIFT) key-points. These two types of features have been used individually and then jointly in our experiments (see Section 5).

Blobs These features are based on the visual properties of large, color-homogeneous regions of the pictures. They have been introduced in the context of image auto-annotation (Duygulu et al., 2002), and have then been widely used for image retrieval, e.g. (Barnard et al., 2003; Monay & Gatica-Perez, 2004). They are extracted from the pictures according to a 3-step process. In the first step, the pictures are segmented into regions using a normalized cut algorithm, each region is then represented with a 36-dimensional vector describing color (18), texture (12) and shape/location (6) information. K-means clustering is then applied to the set of vectors describing the regions of the training pictures, resulting into B region clusters. Finally, each picture p is represented as a histogram over the region clusters, i.e.

$$p^b = (tf^b_{1,p}, \ldots, tf^b_{B,p}),$$

where $tf^b_{i,p}$ denotes the number of regions of p which belong to cluster i. As for text vectors, this representation is then used to obtain *tf-idf-norm* vectors, i.e. for each picture p,

$$p^{b-norm} = (p^{b-norm}_1, \ldots, p^{b-norm}_B),$$

where $\forall i = 1, \ldots, B$,

$$p^{b-norm}_i = \frac{tf^b_{i,p} \cdot idf^b_i}{\sqrt{\sum_{j=1}^{B} (tf^b_{j,p} \cdot idf^b_i)^2}}. \tag{7}$$

In this case, idf^b_i is defined as $-log(r^b_i)$ where r^b_i is the fraction of training pictures containing at least one region of the i^{th} cluster.

SIFTs These features are based on the distribution of edge directions located in the neighborhood of salient points of the pictures (Lowe, 1999). Such features have shown to be effective for different computer vision tasks, such as object recognition (Lowe, 1999), or image categorization (Quelhas et al., 2005). The extraction of SIFT features relies on a 4-step process. In the first step, salient points, or key-points, are extracted from each picture. These points are detected as scale-space extrema using a difference-of-Gaussian detector. Each detected point is then described according to the distribution of edge directions in its neighborhood. The key-points of the training images are then clustered into S clusters using the K-means algorithm. As for blob features, each picture p is then represented as a histogram over the key-point clusters, i.e.

$$p^s = (tf^s_{1,p}, \ldots, tf^s_{S,p}),$$

where $tf^s_{i,p}$ denotes the number of key-points of p which belong to cluster i. Finally, each picture p is represented with a *tf-idf-norm* vector p^{s-norm}, see (7).

Combining Blobs and SIFTs The features are used jointly by concatenating the $tf \cdot idf$ vectors of blobs and SIFTs and then normalizing the resulting vector according to the $L2$-norm. For instance, a picture p is assigned the vector,

$$p^{b+s} = (\quad \begin{array}{l} tf^b_{1,p} \cdot idf^b_1, \ldots, tf^b_{B,p} \cdot idf^b_B, \\ tf^s_{1,p} \cdot idf^s_1, \ldots, tf^s_{S,p} \cdot idf^s_S \end{array} \quad),$$

which is then $L2$-normalized. It can be observed that this normalization *after* concatenation is not equivalent to concatenating the previously normalized $tf \cdot idf$ vectors of blobs and SIFTs. The late normalization was preferred since it led to better validation performance over *Corel* data (see Section 5).

4 Related Work

Contrary to our approach, most of the work in image retrieval from text queries focussed on non-discriminant approaches, e.g. (Barnard et al., 2003). In this context, a model assuming some dependencies between terms and discrete visual features is trained over a corpus of images annotated with text captions. The trained model then allows one to infer a conditional multinomial over the vocabulary, $\{P(t|p), \forall t = 1, \ldots, T\}$, for any non-captioned image p. The retrieval of images from text queries can then be performed through the application of text retrieval techniques over the inferred multinomials. More precisely, a text vector $c^p = (c^p_1, \ldots, c^p_T)$ is computed from $P(\cdot|p)$ for each picture p:

$$\forall t = 1, \ldots, T, c^p_t = \frac{P(t|p) \cdot idf_t}{\sqrt{\sum_{i=1}^{T} (P(i|p) \cdot idf_i)^2}}.$$

This c^p vector can then be compared to any query q according to the function $F^{text}(\cdot, \cdot)$, see (2).

In the following, we briefly describe the 3 main approaches relying on this methodology: Cross-Media Relevance Model (Jeon et al., 2003), Cross-Media Translation Table (Pan et al., 2004) and Probabilistic Latent Semantic Analysis (Monay & Gatica-Perez, 2004). Other methods, such as Latent Dirichlet Allocation, or Hierarchical Mixture Models (both presented in Barnard et al., 2003) could also have been described in this section. However, due to space limitation, we decided to focus on models which have shown to be the most effective according to the standard retrieval evaluation procedure (see Section 5). It should further be noticed that, although the presented models (CMRM, CMTT and PLSA) have been introduced to be used with the blob features (see Section 3), we adopt a generic notation for visual features as nothing prevents one from using other discrete features, such as SIFTs.

4.1 Cross-Media Relevance Model

CMRM (Jeon et al., 2003), is inspired by Cross-Lingual Relevance Model (Lavrenko et al., 2002), considering that the caption of a training image is the translation of its visual properties into words. Given a test picture p^{test}, CMRM infers the distribution $\{P(t|p), \forall t = 1, \ldots, T\}$ from its discrete visual elements, summarized in the tf vector,

$$p^{test} = (tf^v_{1,p^{test}}, \ldots, tf^v_{V,p^{test}}).$$

In this vector, $\forall i = 1, \ldots, V$, $tf^v_{i,p^{test}}$ is the number of elements of type i in p^{test} and V is the number of element types. For example, if the blob representation is used (see Section 3), we have $V = B$ and $tf^v_{i,p^{test}} = tf^b_{i,p^{test}}$.

In a first step, the joint distribution of a term t and the visual elements of p^{test} is estimated by its expectation over the training images P_{train},

$$P(t, p^{test}) = \sum_{p \in P_{train}} P(p) \cdot P(t, p^{test}|p).$$

It is then assumed that terms and visual elements are independent given a training image p, leading to

$$P(t, p^{test}) = \sum_{p \in P_{train}} P(p) \cdot P(t|p) \prod_{v=1}^{V} P(v|p)^{tf^v_{v,p^{test}}}. \quad (8)$$

In this equation, the probability $P(p)$ is assumed to be uniform over the training set, i.e. $P(p) = 1/|P_{train}|$, while the probabilities $P(t|p)$ and $P(v|p)$ are estimated through maximum likelihood estimates, smoothed with the *Jelinek-Mercer* method. Relying on (8), $P(t|p^{test})$ can then be estimated through Bayes rule, i.e. $P(t|p^{test}) = P(t, p^{test})/P(p^{test})$. Although simple, this approach has shown to be more effective when compared to other approaches inspired by translation models, e.g. (Duygulu et al., 2002).

4.2 Cross-Media Translation Table

CMTT (Pan et al., 2004) aims at estimating the similarity between each term t and each visual feature v, $sim(t,v)$. These similarities are inferred from the co-occurrences of t and v in the training set P_{train}. More precisely, the following procedure is applied: first, the training pictures are represented as a $|P_{train}|$-by-V matrix D^v in which the (p,v) element corresponds to the $tf \cdot idf$ weight of the visual feature v in picture p. A similar representation is also adopted for the training captions, i.e. D^t is a $|P_{train}|$-by-T matrix in which the (p,t) element corresponds to the $tf \cdot idf$ weight of term t in the caption of picture p. These matrices are then concatenated,

$$D = [D^t D^v],$$

and Singular Value Decomposition (SVD) is applied to approximate D with a lower rank matrix,

$$D^{SVD} = [D^{t-SVD} D^{v-SVD}],$$

the objective of this step being to clean up noise. The similarity between a term t and a visual feature v is then computed according to the cosine of the corresponding columns of D^{SVD}, i.e.

$$sim(t,v) = cos(D^{t-SVD}_{\cdot,t}, D^{v-SVD}_{\cdot,v}).$$

These similarities are then used to estimate $p(t|p)$ for each test picture p,

$$p(t|p) = \frac{w_{t,p}}{\sum_{t'=1}^{T} w_{t',p}},$$

$$\text{where} \quad w_{t,p} = \sum_{v=1}^{V} tf^v_{v,p} \frac{sim(t,v)}{\sum_{i=1}^{V} sim(t,i)}.$$

This model has two main advantages compared to the CMRM model: first, it can benefit from effective term weighting techniques, such as $tf \cdot idf$. Second, it explicitly deals with noise in the annotation data through the use of SVD. However, CMTT has also some limitations, the main one being that the computation of the cosine of similarity between training occurrence patterns only allows to model simple relationships between terms and visual features. In order to circumvent this problem, approaches allowing to model more complex relationships, such as Probabilistic Latent Semantic Analysis have been applied (Monay & Gatica-Perez, 2004).

4.3 Probabilistic Latent Semantic Analysis

PLSA has been introduced in the context of text retrieval (Hofmann, 2001) and it has recently been extended to image retrieval (Monay & Gatica-Perez, 2004). This model introduces the following conditional independence assumption: "terms and visual features are independent from pictures conditionally to an unobserved discrete variable $z_k \in \{z_1, \ldots, z_K\}$ (z_k is called *aspect* variable and the hyperparameter K is referred to as the number of aspects)". In this framework, the probability of observing a term t or a visual feature v in a picture p follows

$$P(p,t) = P(p) \cdot \sum_k P(z_k|p) P(t|z_k), \tag{9}$$

$$P(p,v) = P(p) \cdot \sum_k P(z_k|p) P(v|z_k). \tag{10}$$

The different parameters of the model can be estimated as follows: first, the probabilities $P(p), P(z_k|p)$ and $p(t|z_k)$ for all $p \in P_{train}$ are estimated to maximize the training caption likelihood through the Expectation Maximization algorithm (EM). The probabilities $P(v|z_k), \forall v, k$ are then fitted to maximize the training picture likelihood (at this step $P(p), P(z_k|p)$ are kept fixed). For test pictures which have no caption, the following procedure is then applied: the probabilities $P(p), P(z_k|p)$ are estimated to

maximize the test picture likelihood, keeping $P(v|z_k), \forall(v,k)$ to the values estimated during training. After this step, (9) can then be used to infer $P(p,t)$ for any test picture/term pair (p,t). Similarly to CMRM, Bayes rule is then applied to compute $P(t|p)$ from $P(p,t)$.

This model has several strengths: the latent aspect assumption allows one to model more complex dependencies between term and visual features than the above presented models. Moreover, the use of multinomials allows for efficient training over large datasets which is not necessary the case for other latent models, e.g. Latent Dirichlet Allocation (Barnard et al., 2003). However, like the above approaches, PLSA also relies on a non-discriminative criterion (i.e. data likelihood) which may be suboptimal when targeting a specific task, i.e. text-based image retrieval in our case. In fact, our experiments clearly show that the proposed discriminative approach is indeed more effective than the above methods.

5 Experiments and Results

This section presents the experiments performed. The experimental setup is first described and the results are then discussed.

5.1 Experimental Setup

The Corel Dataset Our experiments are performed over pictures from the *Corel* database[1]. These pictures are photographs of various scenes such as bears in the wilderness, sunsets, air-shows, etc. Each picture is annotated with several keywords describing the main objects depicted. The subset of *Corel* used for these experiments contains $5,000$ pictures, which either belong to the 4500-picture development set (P_{dev}) or to the 500-picture test set (P_{test}). This split of the data originates from (Duygulu et al., 2002) and has been widely used the literature, e.g. (Pan et al., 2004; Monay & Gatica-Perez, 2004). For model training and hyperparameter selection, we further split the development set into a $4,000$-picture train set (P_{train}) and a 500-picture validation set (P_{valid}).

In addition to pictures, retrieval queries and the corresponding relevance assessments are also needed to train and evaluate our approach. As the *Corel* dataset does not provide such data, we generated queries and the corresponding relevance assessments from image captions. We defined the query set Q_{train} as the set containing all queries having at least one relevant picture in P_{train} according to the following rule: "a picture p is considered as relevant to a query q if and only if the caption of p contains all the words in q". The same procedure has also been applied to generate the sets Q_{valid} and Q_{test}.

Although automatic, this query generation process is based on manually produced captions and the resulting relevance information can thus be considered as reliable. In fact, there is no doubt that the pictures marked as relevant with our labeling technique are indeed relevant, e.g. if the words *beach, sky* are present in a caption, it can confidently be claimed that the corresponding picture is relevant to the queries *"beach", "sky"* and *"beach sky"*. The only problem that could affect our relevance data is due to the possible incompleteness of some captions, i.e. if a word is missing from a caption, the corresponding picture will wrongly be marked as non-relevant to all queries containing this word. This weakness is however not specific to our labeling process: e.g. *system pooling*, the semi-automatic technique used for labeling data for standard IR benchmarks, also tends to underestimate the number of relevant documents (Baeza-Yates & Ribeiro-Neto, 1999).

Once the queries are generated, we hence have three picture/query sets, i.e. $D_{train} = (P_{train}, Q_{train})$, $D_{valid} = (P_{valid}, Q_{valid})$ and $D_{test} = (P_{test}, Q_{test})$, see Table 1 and Table 2 for set statistics. PAMIR, CMRM, CMTT and PLSA are then trained and evaluated relying on these data, using the following procedure: in a first step, D_{train} is used for parameter fitting (i.e. the training criterion is optimized over this set) and D_{valid} is used for hyperparameter selection (i.e. we select the hyperparameters which maximize average precision over D_{valid}). In a second step, D_{train} and D_{valid} are used jointly to re-train each model with its selected hyperparameters. The models trained at this step are then evaluated over D_{test}, as explained in next section.

[1] *Corel* data are available at `www.fotosearch.com`.

Table 1: Picture Set Statistics.

	P_{train}	P_{valid}	P_{test}
Number of pictures	4,000	500	500
Number of blob clusters		500	
Avg. # of blobs per pic.	9.43	9.33	9.37
Number of SIFT clusters		1,000	
Avg. # of SIFTs per pic.	232.8	226.3	229.5

Table 2: Query Set Statistics.

	Q_{train}	Q_{valid}	Q_{test}
Number of queries	7,221	1,962	2,241
Avg. # of rel. pic. per q.	5.33	2.44	2.37
Vocabulary size		179	
Avg. # of words per query	2.78	2.51	2.51

Evaluation Methodology The performance of PAMIR over the test data has been assessed according to standard IR measures (Baeza-Yates & Ribeiro-Neto, 1999). For each test query $q \in Q_{test}$, the images of P_{test} have been ranked with respect to $\{F_w(q,p), \forall p \in P_{test}\}$. This ranking is then compared to the ideal case, i.e. the pictures relevant to q appear above the others, according to the following measures:

P10 Precision at top 10 pictures is defined as the percentage $Pr(10)$ of relevant pictures within the top 10 positions of the ranking. This measure hence corresponds to the percentage of relevant material that would appear in the first 10–result page of a search engine. Although it is easy to interpret, this measure tends to overweight queries with a large number of relevant pictures when averaging over a query set. In the case of such queries, it is easier to rank some relevant pictures within the top 10, simply because the relevance set is larger and not because of any property of the ranking approach.

BEP Break-Even Point evaluates the precision at the top $|R(q)|$ pictures, $|R(q)|$ being the number of relevant pictures for the evaluated query q. This hence corresponds to the percentage $Pr(|R(q)|)$ of relevant document within top $|R(q)|$. It is also often called R-precision. Contrary to $P10$, this measure does not overweight queries with many relevant pictures.

AvgP Average Precision is the standard measure used for IR benchmark (Baeza-Yates & Ribeiro-Neto, 1999), and it corresponds to the average of the precision at each position where a relevant document appears, i.e.

$$AvgP = \frac{1}{|R(q)|} \sum_{d \in R(q)} Pr(rk_{d,q}),$$

where $rk_{d,q}$ is the rank of document d for query q.

The results of PAMIR are then reported according to the average of these measures over the set of test queries Q_{test}. For a sake of comparison, the alternative models presented in Section 4 have also been evaluated according to this methodology. The next section summarizes these results.

5.2 Experimental Results

In this section, we first report the hyperparameter values selected before presenting PAMIR results over test data. These results are then compared to those of the alternative models (see Section 4).

Hyperparameter Selection As already mentioned, the validation data have been used for hyperparameter selection, i.e. the selection of m the number of training iterations and C, the aggressiveness

Table 3: Model hyperparameters.

	C	m	Criterion
Blobs	0.01	$1.75 \cdot 10^6$	*caption-*ϵ
SIFTs	0.001	$94.6 \cdot 10^6$	*caption-*ϵ
Blobs + SIFTs	0.01	$19.0 \cdot 10^6$	*caption-*ϵ

Table 4: Average precision (%) for test queries.

	CMRM	CMTT	PLSA	PAMIR
Blobs	10.4	11.8	9.7	11.9
SIFTs	10.8	9.1	12.3	**16.0**
Blobs + SIFTs	14.7	11.5	16.7	**21.6**

parameter. These data have also been used to determine which of the *constant-*ϵ or *caption-*ϵ criterion is the most effective (see Section 2). The selected values are reported in Table 3. It should be noted that the *caption-*ϵ criterion has been preferred to the *constant-*ϵ criterion. However, this choice has only a slight effect on the validation performance (e.g. 16.5 for *caption-*ϵ vs 16.4 for *constant-*ϵ in the case of *SIFT* features). This means that satisfying performance may still be obtained if the *caption-*ϵ criterion cannot be used, i.e. in absence of available training captions.

Generalization Performance AvgP results for all visual feature setups (see Section 3) are reported in Table 4. As seen from the table, PAMIR outperforms all the other evaluated models, e.g. for the combination of *blob* and *SIFT* features, PAMIR yields 21.6% AvgP which corresponds to a relative improvement of 29% over the second best model (PLSA with 16.7% AvgP). In order to determine whether the PAMIR advantage observed on the average could be due to a few queries, we further compared PAMIR results with those of the alternative approaches for each of the $2,241$ queries and performed the Wilcoxon signed rank test (Rice, 1995) over these data. The test rejected this hypothesis with 95% confidence for both *SIFT* and *blob+SIFT* features (such a test outcome is indicated by bold numbers in the tables). In the case of *blob* features, the test concluded that PAMIR performance is similar to CMTT but better than the other models.

As an alternative to AvgP, we also looked at the performance in terms of P10 and BEP, as explained in previous section. Table 5 reports these results for the *blobs+SIFT* features[2]. These measurements confirm the superiority of PAMIR: for all measures, PAMIR yields significantly better results when compared to any alternative model among CMRM, CMTT and PLSA. Looking closely at Table 5, one could remark that the P10 values reported are quite low, e.g. only 0.88 relevant picture within top 10 for our method. These low values should however not be regarded as a failure of the models since the very low number of relevant pictures per query should also be considered (see Table 2). In fact, P10 cannot be higher than 20.2% for our Q_{test} set.

Since several previous papers only reported results over single word queries (Pan et al., 2004; Monay & Gatica-Perez, 2004), we also performed a set of experiments over this type of query. For that purpose, PAMIR has been trained and evaluated relying on the subsets of Q_{train}, Q_{valid} and Q_{test} containing only single word queries. These queries correspond to a more restrictive scenario, i.e. users are not given the possibility to submit multiple-word queries. Moreover, single-word queries generally have more relevant pictures than multiple-word queries, which makes the retrieval task easier (in our test data, each single-word query has 9.3 relevant pictures on average, compared to 2.4 for the whole query set). Table 6 reports the results of the experiments over single-word queries. In this case, PAMIR outperforms the alternative approaches for both *SIFT* and *blobs+SIFT* features, this improvement being significant according to Wilcoxon test. In the case of *blob* features alone, PAMIR is the second best model, after CMTT. This might underline that blob information is noisy and the

[2]We do not report the measurements for blobs and SIFTs individually due to space limitation.

Table 5: Average precision, break even point and precision at top 10 over test queries (Q_{test}) for *Blob + SIFT* features. All measures are reported as percentage.

	CMRM	CMTT	PLSA	PAMIR
AvgP	14.7	11.5	16.7	**21.6**
BEP	10.5	5.9	10.5	**13.4**
P10	5.8	5.5	7.1	**8.8**

Table 6: Average precision (%) over single-word test queries.

	CMRM	CMTT	PLSA	PAMIR
Blobs	14.2	**17.2**	15.5	16.6
SIFTs	14.2	15.1	17.1	**23.8**
Blobs + SIFTs	19.2	19.1	24.5	**30.7**

use of a technique to improve noise robustness, such as SVD in the case of CMTT, can be beneficial.

The overall outcome of these experiments is hence positive, underscoring the benefit of using a discriminative approach to the problem of image retrieval from text queries.

6 Conclusions

In this paper, we proposed a discriminative approach to the retrieval of images from text queries. After introducing the model parameterization, we presented a margin loss adapted to this retrieval task. We then proposed an adaptation of the *Passive-Agressive* algorithm (Crammer et al., 2003) to identify the model parameters which minimize this loss.

Our model, PAMIR, has then been evaluated over the *Corel* dataset. These experiments have been performed relying on different visual features that describe color homogene regions or salient points of the images. The results have then been compared to those of state-of-the-art approaches, which rely on non-discriminant models. It has been observed that PAMIR outperforms the alternative approaches for most queries, e.g. for the most effective visual features, *Blobs+SIFTs*, the reported AvgP for PAMIR is 21.6% which should be compared to 16.7% for PLSA, the second best model.

The results of PAMIR are hence promising and need to be confirmed over other datasets. Furthermore, it would also be of a great interest to investigate on the use of non-linear kernels in PAMIR. In this work, we relied on the linear kernel over feature histograms to compare images. However, like any *Passive-Agressive* model (Crammer et al., 2003), PAMIR could benefit from other Mercer kernels. In particular, recently proposed image kernels, such as (Wallraven & Caputo, 2003), could be effective for our task.

Acknowledgments

This work has been performed with the support of the Swiss NSF through the NCCR–IM2 project. It was also supported by the PASCAL European Network of Excellence, funded by the Swiss OFES.

References

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. Harlow, England: Addison Wesley.

Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D. M., & Jordan, M. I. (2003). Matching words and pictures. *J. of Machine Learning Research (JMLR)*, *3*, 1107–1135.

Crammer, K., Dekel, O., Shalev-Shwartz, S., & Singer, Y. (2003). Online passive-aggressive algorithms. *Conf. on Advances in Neural Information Processing Systems (NIPS)*.

Duygulu, P., Barnard, K., de Freitas, N., & Forsyth, D. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. *European Conference on Computer Vision (ECCV)* (pp. 97–112).

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, *42*, 177–196.

Jeon, J., Lavrenko, V., & Manmatha, R. (2003). Automatic image annotation and retrieval using cross-media relevance models. *ACM Special Interest Group on Information Retrieval (SIGIR)*.

Joachims, T. (2001). *Learning to classify text using support vector machines*. Dordrecht, The Netherlands: Kluwer.

Joachims, T. (2002). Optimizing search engines using clickthrough data. *Int. Conf. on Knowledge Discovery and Data Mining (KDD)*.

Lavrenko, V., Choquette, M., & Croft, W. B. (2002). Cross-lingual relevance models. *ACM Special Interest Group on Information Retrieval (SIGIR)* (pp. 175–182).

Lowe, D. G. (1999). Object recognition from local scale-invariant features. *Int. Conf. on Computer Vision (ICCV)* (pp. 1150–1157).

Monay, F., & Gatica-Perez, D. (2004). PLSA-based image auto-annotation: constraining the latent space. *ACM Multimedia* (pp. 348–351).

Pan, J. Y., Yang, H. J., Duygulu, P., & Faloutsos, C. (2004). Automatic image captioning. *Int. Conf. on Multimedia and Expo (ICME)* (pp. 1987–1990).

Quelhas, P., Monay, F., Odobez, J. M., Gatica-Perez, D., Tuytelaars, T., & Gool, L. J. V. (2005). Modeling scenes with local descriptors and latent aspects. *Int. Conf. on Computer Vision (ICCV)* (pp. 883–890).

Rice, J. (1995). *Rice, mathematical statistics and data analysis*. Belmont, California: Duxbury Press.

Wallraven, C., & Caputo, B. (2003). Recognition with local features: the kernel recipe. *Int. Conf. on Computer Vision (ICCV)*.