A Discriminative Approach for the Retrieval of Images from Text Queries

David Grangier[1,2], Florent Monay[1,2], and Samy Bengio[1]

[1] IDIAP Research Institute, Martigny, Switzerland,
firstname.lastname@idiap.ch,
[2] Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland

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 model: 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 pictures. 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 discriminative approach 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). The *Passive-Aggressive* algorithm [1] 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 [2–4], see Section 4. In fact, the optimization of a loss related to the final retrieval performance is a key aspect of PAMIR and our experiments over the *Corel* data show the advantage of this discrimative approach (see Section 5). PAMIR is reported to outperform several models, such as Cross Media Relevance Model, CMRM [3], Cross Media Translation Table, CMTT [5], or Probabilistic Latent Semantic Analysis, PLSA [4] for various feature extraction setups. 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 pictures, 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 this problem, 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 $\mathcal{T}$. 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 [6]. In other words, the scoring function is

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

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.

$$\forall k, \quad F_w(q_k, p_k^+) - F_w(q_k, p_k^-) > 0. \tag{2}$$

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

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

where $\epsilon > 0$. This equation can then be rewritten as,

$$\forall k, \quad l(w; (q_k, p_k^+, p_k^-)) = 0,$$
$$\text{where } \ l(w; (q_k, p_k^+, p_k^-)) = \max\left\{0, \epsilon - F_w(q_k, p_k^+) + F_w(q_k, p_k^-)\right\}.$$

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 (in the following, we arbitrarily set $\epsilon = 1$ since any positive value would lead to the same optimization problem). This *margin* criterion is inspired from the ranking SVM approach, which has successfully been applied to text retrieval [7]. Our model is however different from ranking SVM in both its parameterization and its optimization procedure [1]. In fact, we use the online Passive-Aggressive minimization algorithm which does not rely on quadratic optimization like ranking SVM, allowing PAMIR to scale to large constraint sets (e.g. there are $\sim 10^8$ constraint triplets in the training data presented in Section 5).

2.4 Training Procedure

Our goal is to minimize the loss

$$L(w; D_{train}) = \sum_{k=1}^{n} l(w; (q_k, p_k^+, p_k^-)). \tag{3}$$

For that purpose, we adapt the *Passive-Aggressive* (PA) algorithm, originally introduced for classification and regression problems [1], to minimize this retrieval loss. For this minimization, the algorithm constructs a sequence of weight vectors $(w^0, \dots, w^m)$ according to the following iterative procedure: the first vector is set to be zero, $w^0 = 0$ and, at the i^{th} iteration, the weight w^i is selected according to the i^{th} training example and the previous weight w^{i-1},

$$w^i = \operatorname*{argmin}_w \frac{1}{2} \|w - w^{i-1}\|^2 + C \cdot l(w; (q_i, p_i^+, p_i^-)). \tag{4}$$

This means that, at each iteration, we select the weight w^i as a trade-off between minimizing the loss on the current example $l(w; (q_i, p_i^+, p_i^-))$ and remaining close to the previous weight vector w^{i-1}. The *aggressiveness* parameter C controls this trade-off. Adopting an approach similar to [1], it can be shown that the solution of problem (4) is

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

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

$$w = \operatorname*{argmin}_{w \in \{w^0, \dots, w^m\}} L(w; D_{valid}).$$

The hyperparameter C has also been selected to maximize the performance over D_{valid}. The proof that the above procedure actually minimizes the loss (3) is not included here due to space constraint but can easily be inferred from the proof given in [1].

3 Text and Picture Representations

This section describes the representations used for text and pictures.

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

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

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 [6], 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. It should be noted that the definition of *idf* assumes that the training pictures are labeled with a caption. This is the case for the *Corel* data used in our experiments (see Section 5). However, were such captions to be unavailable, it would still be possible to compute *idf* relying on another textual corpus, such as an encyclopedia.

3.2 Picture Representation

Similarly to previous work (See Section 4), the *visterm* approach has been used for picture representation. The main idea of this approach is to define different classes of image regions, referred to as the *visual vocabulary*, which then allows the representation of each picture p as a histogram over this vocabulary. In practice, vocabulary definition is performed automatically through the following 3-step process: first, regions of interests are detected from each training picture; second, each extracted region is assigned a vector describing its visual properties; third, the vocabulary is built through k-means clustering of the training region descriptors. Finally, any picture p (either from train or test set) is assigned the histogram,

$$p = (vtf_{p,1}, \ldots, vtf_{p,V}), \tag{5}$$

where V is the visual vocabulary size and $vtf_{p,i}$ is the number of regions of p that belongs to the i^{th} visual vocabulary cluster. In our case, we used two types of visterms, either individually or jointly.

Blobs describes the visual properties of large, color-homogeneous regions. In this case, region detection is performed with a normalized cut algorithm and the region descriptors are 36-dimensional vectors summarizing color (18), texture (12) and shape (6) information of the region, see [8].

SIFTs describes edge properties of areas around salient points of the picture. In this case, region detection is performed with a difference-of-Gaussian detector and region descriptors consist of edge histograms, see [9].

Blob+SIFT visterms have also been combined through the concatenation of their histograms.

Like for the text features, we also applied the normalized $tf \cdot idf$ weighting to visterm histograms, i.e. each picture p is represented with:

$$p = (p_1, \ldots, p_V), \text{ where } p_i = \frac{vtf_{p,i} \cdot vidf_i}{\sqrt{\sum_{j=1}^{V} vtf_{p,i} \cdot vidf_j}} \tag{6}$$

where $vidf_i = -log(vr_i)$ with vr_i referring to the fraction of training pictures containing at least one region mapped to the i^{th} cluster. Space limitation prevents us from reporting the results of the experiments over validation data that concluded on the superiority of this weighting compared to (5).

4 Related Work

The previous work in image retrieval from text queries mainly focused on an intermediate step, *image auto-captioning*. This task consists in estimating the likelihood of a textual annotation, or caption, given an unannotated picture. Given a query q, such a model then allows the user to retrieve the pictures for which q is the most likely. In this context, several models such as Cross-Media Relevance Models (CMRM) [3], Probabilistic Latent Semantic Analysis (PLSA) [4] or Latent Dirichlet Allocation (LDA) [2] have been proposed. These model hence learn a captioning model from a set of training picture that have been manually annotated. Even if such approaches are leading to state-of-the-art performance, it could seem questionable to focus on an intermediate annotation problem when the final goal is to solve a retrieval problem. It would be more appropriate to adopt a discriminative approach and directly optimize a loss related to the retrieval performance of the model. However, to the best of our knowledge, no discriminative approaches have been proposed in the context of image retrieval prior to this work. Previous discriminative approaches have only focussed on categorization ranking problems (e.g. [10, 11]), i.e. the task of ranking unseen pictures with respect to queries or categories *known* at training time. This task is hence different from a true retrieval task in which a *new* query (i.e. any set of vocabulary words) can be submitted.

In absence of discriminative alternatives, this section will therefore focus on the non-discriminative approaches that have shown to be the most effective over the benchmark *Corel* dataset: Cross-Media Relevance Model (CMRM) [3], Cross-Media Translation Table (CMTT) [5] and Probabilistic Latent Semantic Analysis (PLSA) [4]. The proposed PAMIR approach will then be compared to these models in Section 5.

4.1 Cross-Media Relevance Model

In order to estimate the probability of a term t given a picture p^{test}, $P(t|p^{test})$, CMRM [3] estimates the joint probability $P(t, p^{test})$ and then relies on Bayes rule. The joint probability $P(t, p^{test})$ is estimated as its expectation over the training pictures,

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

The picture p^{test} is considered as a set of discrete features or visterms (see Section 3), i.e. $p^{test} = \{v_1, \ldots, v_m\}$, which means that:

$$P(t, p^{test}) = \sum_{p^{train} \in D_{train}} P(p^{train}) \cdot P(t, v_1, \ldots, v_m|p^{train}).$$

Terms and visterms are then assumed to be independent given a training picture, leading to:

$$P(t, p^{test}) = \sum_{p^{train} \in D_{train}} P(p^{train}) \cdot P(t|p^{train}) \prod_{i=1}^{m} P(v_i|p^{train})$$

The probabilities $P(t|p^{train})$ and $P(v_i|p^{train})$ are then estimated through maximum likelihood estimates, smoothed with the *Jelinek-Mercer* method. Although simple, this approach has shown to yield good performance over the standard *Corel* dataset [3].

4.2 Cross-Media Translation Table

The CMTT model borrows its parameterization from cross-lingual retrieval techniques [5]. In this case, textual terms and visterms are considered as words originating from two different languages and CMTT constructs a translation table containing the similarities $sim(t, v)$ between any pair of term/visterm (t, v). This translation table is then used to estimates $p(t|p^{test})$ for any term t and any picture:

$$P(t|p^{test}) = \frac{w_{t,p_{test}}}{\sum_{i=1}^{T} w_{i,p_{test}}}, \text{ where } w_{t,p_{test}} = \sum_{i=1}^{m} sim(t, v_i),$$

$v_1, \ldots, v_m$ being the visterms of p^{test}. The translation table is computed from the training data D_{train} according to the following process: in a first step, each term i and each visterm j is represented by a $|D_{train}|$ dimensional vector, t_i or v_j, in which each component k is the weight of term i (or visterm j) in the k^{th} training example (the weighting scheme used here is $tf \cdot idf$, as defined in Section 3). As a noise removal step, the matrix $M = [t_1, \ldots, t_T, v_1, \ldots, v_V]$ containing all term and visterm vectors is approximated with a lower rank matrix, $M' = [t'_1, \ldots, t'_T, v'_1, \ldots, v'_K]$, through Singular Value Decomposition (SVD). The similarity $sim(i, j)$ between a term i and a visterm j is then defined as

$$sim(i, j) = \frac{\cos(t'_i, v'_j)}{\sum_{k=1}^{V} \cos(t'_i, v'_k)}.$$

Like CMRM, this method has also been evaluated over the *Corel* corpus [5], where it has shown to be effective. The use of SVD has notably shown to improve noise robustness. However, CMTT has also some limitations, the main one being that cosine similarity 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 [4], have been applied.

4.3 Probabilistic Latent Semantic Analysis

PLSA, introduced for text retrieval [12], has recently been applied to image retrieval [4]. This model assumes that the observation of a picture p and a term t in its caption are independent conditionally to a discrete latent variable $z_k = \{z_1, \ldots, z_K\}$,

$$P(p, t) = P(p) \sum_{k=1}^{K} P(z_k|p) P(t|z_k), \tag{7}$$

where K is a hyperparameter of the model. A similar conditional independence assumption is also made for visterms,

$$P(p, v) = P(p) \sum_{k=1}^{K} P(z_k|p) P(v|z_k).$$

In this framework, the different parameters of the model, i.e. $P(z_k|p)$, $P(t|z_k)$, $P(v|z_k)$ are trained through the Expectation Maximization (EM) algorithm. In fact, a modified version of EM is applied such that the latent space is constrained toward the text modality. This yields a latent space that better models the semantic relationships between pictures. Once parameter fitting over the set of training pictures is performed, it is still needed to infer $P(z_k|p), \forall k$, for any unnatotated test picture p. This estimation is performed to maximize the test picture likelihood, keeping $P(v|z_k), \forall(v, k)$ to the values estimated during training. After this step, (7) can then be used to infer $P(p, t)$ for any test picture/term pair (p, t). Similarly to CMRM, Bayes rule is applied to compute $P(t|p)$ from $P(p, t)$. This PLSA model has shown to be effective empirically, especially when the latent space is constraint toward the text modality as explained in [4].

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[3] consists of 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. In this work, we used a $5,000$-picture subset of *Corel*. This subset has been defined in [8]: it contains 4500 development pictures (P_{dev}) and 500 test pictures (P_{test}). This split has been widely used in the literature, e.g. [5, 4], and has hence become a kind of benchmark to compare image retrieval algorithm. In our case, we further split the development set into a $4,000$-picture train set (P_{train}) and a 500-picture validation set (P_{valid}), which allows us to perform model training and hyperparameter selection on different subsets.

Relevance data has been defined relying on picture captions, as explained in [3]: a picture p is considered as relevant to a query q if and only if its caption contains all the terms of q. The query sets Q_{train}, Q_{valid} and Q_{test} are then defined as the set of all queries which have at least one relevant picture among P_{train}, P_{valid} and P_{test} respectively. The statistics for the three picture/query sets, i.e. $D_{train} = (P_{train}, Q_{train})$, $D_{valid} = (P_{valid}, Q_{valid})$ and $D_{test} = (P_{test}, Q_{test})$ are summarized in Table 1 and Table 2. The PAMIR model has then been trained and evaluated relying on these data with the following setup: parameter fitting has been first performed over D_{train} (i.e. the training criterion is optimized over this set) and the hyperparameters (i.e. the aggressiveness C and the number of iterations m) have been selected relying on D_{valid}. Finally, D_{train} and D_{valid} have been used jointly to re-train the model with its selected hyperparameters. Model evaluation has then been performed over D_{test}, as explained in the next section. The alternative models CMRM, CMTT and

[3] *Corel* data are available at `http://www.emsps.com/photocd/corelcds.htm`.

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

PLSA have also been trained and evaluated according to the same setup for the sake of comparison.

Evaluation Methodology The performance of PAMIR over the test data has been assessed according to standard IR measures [6]. For each test query $q \in Q_{test}$, the pictures 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 documents 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 [6], 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}. The alternative models (i.e. CMRM, CMTT and PLSA) have also been evaluated according to this methodology. The next section summarizes these results.

Table 3. 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**

Table 4. Model hyperparameters.

	C	m
Blobs	0.01	$1.75 \cdot 10^6$
SIFTs	0.001	$94.6 \cdot 10^6$
Blobs + SIFTs	0.01	$19.0 \cdot 10^6$

5.2 Experimental Results

Table 3 reports the AvgP results for all visual feature setups (see Section 3) while Table 4 reports the hyperparameters selected for these experiments. In all feature configurations, PAMIR is reported to outperform the other 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 [13] 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. The low number of visterms per picture (~ 9.5 on average, see Table 1) may explain the relatively good results of CMTT in the case of *Blobs*: we hypothesize that such a concise representation may only provide sufficient statistics to train highly constraint models, such as CMTT. On the contrary, the SIFT representation, where richer statistics are available (~ 230 visterms per picture on average, see Table 1), allows less constraint models, such as PAMIR or PLSA, to reach higher performance than CMTT.

As an alternative to AvgP, we also looked at the performance in terms of P10 and BEP, as explained in the previous section. Table 5 reports these results for the *Blob+SIFT* features[4]. 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 PAMIR. 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.

[4] 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.

	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, break even point and precision at top 10 (%) over *single-word* test queries for *Blob + SIFT* features.

	CMRM	CMTT	PLSA	PAMIR
AvgP	19.2	19.1	24.5	**30.7**
BEP	19.7	17.4	22.2	**27.2**
P10	17.8	17.9	21.3	**25.3**

Since several previous papers only reported results over single word queries (e.g. [5, 4]), 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. the 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 for the best feature configuration, i.e. *Blobs+SIFTs*. In this case, PAMIR outperforms the alternative approaches for all measures, this improvement being significant according to the Wilcoxon test at the 95% confidence level. The use of PAMIR is hence advantageous over the alternative models in both the case where the users focus on the first ranking positions (as shown by P10 results) and the case where the users are interested in the whole ranking (as shown by AvgP results).

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-Aggressive* algorithm [1] 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-homogeneous 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-discriminantive 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 [1], PAMIR could benefit from other Mercer kernels. In particular, recently proposed image kernels, such as [14], could be effective for our task.

Acknowledgments: This work has been performed with the support of the Swiss NSF through the MULTI project. It was also supported by the PASCAL European Network of Excellence, funded by the Swiss SER.

References

1. Crammer, K., Dekel, O., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. In: Neural Information Processing Systems (NIPS). (2003)
2. Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D.M., Jordan, M.I.: Matching words and pictures. Journal of Machine Learning Research (JMLR) **3** (2003) 1107–1135
3. Jeon, J., Lavrenko, V., Manmatha, R.: Automatic image annotation and retrieval using cross-media relevance models. In: ACM Special Interest Group on Information Retrieval (SIGIR). (2003)
4. Monay, F., Gatica-Perez, D.: PLSA-based image auto-annotation: constraining the latent space. In: ACM Multimedia. (2004) 348–351
5. Pan, J.Y., Yang, H.J., Duygulu, P., Faloutsos, C.: Automatic image captioning. In: International Conference on Multimedia and Expo (ICME). (2004) 1987–1990
6. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison Wesley, Harlow, England (1999)
7. Joachims, T.: Optimizing search engines using clickthrough data. In: International Conference on Knowledge Discovery and Data Mining (KDD). (2002)
8. Duygulu, P., Barnard, K., de Freitas, N., Forsyth, D.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: European Conference on Computer Vision (ECCV). (2002) 97–112
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (IJCV) **60**(2) (2004) 91–110
10. Tieu, K., Viola, P.: Boosting image retrieval. International Journal of Computer Vision (IJCV) **56**(1) (2004) 17 – 36
11. Wu, H., LuE, H., Ma, S.: A practical SVM-based algorithm for ordinal regression in image retrieval. In: ACM Multimedia. (2003)
12. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. Machine Learning **42** (2001) 177–196
13. Rice, J.: Rice, Mathematical Statistics and Data Analysis. Duxbury Press (1995)
14. Wallraven, C., Caputo, B.: Recognition with local features: the kernel recipe. In: International Conference on Computer Vision (ICCV). (2003)