Learning to Retrieve Images from Text Queries with a Discriminative Model

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 presents a discriminative model for the retrieval of pictures from text queries. The core idea of this approach is to minimize a loss directly related to the retrieval performance of the model. For that purpose, we rely on a ranking loss which has recently been successfully applied to text retrieval problems. The experiments performed over the *Corel* dataset show that our approach compares favorably with generative models that constitute the state-of-the-art (e.g. our model reaches 21.6% mean average precision with Blob and SIFT features, compared to 16.7% for PLSA, the best alternative).

1 Introduction

Several application domains, such as stock photography providers or web search engines, need tools to search large collections of pictures from text queries. In most commercial applications, these tools generally rely on some manually-produced text associated with each picture and then apply text retrieval techniques over such texts. Although effective, this approach has a major drawback: its human annotation step is a costly process, moreover it often results in incomplete and subjective annotations. In order to circumvent this limitation, several automatic annotation techniques have recently been proposed, e.g. [1–5]. Automatic image annotation is generally performed relying on a generative model that aims at estimating the distribution of words given any picture from a training set of annotated images. Such models include, for instance, Cross-Media Relevance Models (CMRM) [3], Latent Dirichlet Allocation (LDA) [5] or Probabilistic Latent Semantic Analysis (PLSA) [6].

In this paper, we introduce an alternative to these approaches. The proposed model, Passive-Aggressive Model for Image Retrieval (PAMIR), relies on discriminative learning. This means that the model parameters are not selected to maximize the likelihood of some annotated training data; they are instead selected to maximize the retrieval performance of the model over a set of training queries. This has several advantages when compared to generative approaches: from a theoretical point of view, it is attractive to solve the targeted problem directly instead of solving the more complex problem of data generation [7]. From a practical point of view, discriminative methods have been highly successful in several domains and our experiments also confirm this advantage (for single word queries, PAMIR attains 30.7% mean average precision with Blob+SIFT features compared to 24.5% for the second best model, PLSA).

The remainder of this paper is organized as follows: Section 2 introduces our approach, Section 3 presents the features used to represent text queries and images, Section 4 briefly describes previous related research. Section 5 reports the experiments and results. Finally, Section 6 draws some conclusions.

2 Passive-Aggressive Model for Image Retrieval

In this section, we first define the ideal goal that an image retrieval model F is targeting, which allows us to define a training loss L related to this objective. Then, we introduce the parameterization of our model F_w and we explain the optimization procedure adopted to select the parameters w^* that minimize L over a given training set D_{train}.

2.1 Ranking Loss

Before introducing the ranking loss, we should first recall the ideal goal of an image retrieval system. Given a set of pictures P and a query q, such a system should rank the pictures of P such that the pictures relevant to q appear above the non-relevant ones. In order to address such a problem, a scoring function F that assigns a real value $F(q,p)$ to any query/picture pair (q,p) is generally introduced [8]. Given a query q, a retrieval system then simply computes the scores $\{F(q,p), \forall p \in P\}$ and ranks the pictures of P by decreasing scores. The effectiveness of such a system is hence mainly determined by the choice of an appropriate function F. In fact, optimal retrieval performance would be achieved if F satisfies

$$\forall q, \forall p^+ \in R(q), \forall p^- \notin R(q), F(q,p^+) > F(q,p^-), \quad (1)$$

where $R(q)$ refers to the pictures of P which are relevant to q. In other words, if F satisfies (1), the retrieval system will always rank the relevant pictures above the non-relevant ones.

Hence, our learning problem is to identify a function F which is likely to satisfy (1) for any unseen queries and pictures, given only a limited amount of training data D_{train}. For that purpose, we need a loss function L such that the selection of a function F minimizing $F \rightarrow L(F; D_{train})$ ensures that F also yield good retrieval performance over unseen data. In fact, such a loss has recently been introduced in the text retrieval literature [9, 10] and we propose to apply it to our image retrieval problem. This loss, referred to as the *ranking loss* in the following, assumes that we are given a set of training triplets,

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

where, for all k, p_k^+ is a picture relevant to query q_k and p_k^- is a picture non-relevant to query q_k, and is defined as follows:

$$\begin{aligned} L(F; D_{train}) &= \sum_{k=1}^{n} l(F; q_k, p_k^+, p_k^-) \\ &= \sum_{k=1}^{n} \max(0, 1 - F(q_k, p_k^+) + F(q_k, p_k^-)). \end{aligned}$$

This means that minimizing L favors the selection of functions F such that, for all k, the score $F(q_k, p_k^+)$ is greater than $F(q_k, p_k^-)$ by at least a *margin* of 1 (the choice of 1 is arbitrary here and any positive constant would lead to the same optimization problem). This notion of *margin* is a key aspect of this criterion and has shown to yield good generalization performance when applied over different text retrieval tasks [9, 10].

2.2 Model Parameterization

In this section, we describe a family of parameterized functions F_w that are suitable for our task. This parameterization is inspired from text retrieval, i.e. the retrieval of *text* documents from *text* queries. In this case, documents and queries are generally represented with *bag-of-words* vectors, i.e. each text item t is assigned a vocabulary-sized vector in which the i^{th} component is a weight related to the presence or absence of term i in t (see Section 3 for a detailed description). Each query/document pair (q, d) is then assigned a score corresponding to the inner product of their vector representation [8], i.e.

$$F^{text}(q, d) = q \cdot d = \sum_{i=1}^{T} q_i \cdot d_i,$$

where T is the vocabulary size.

In our case, we adopt a similar approach and we compute the score of a picture/query pair (q, p) according to:

$$F_w(q, p) = F^{text}(q, f_w(p)) \tag{2}$$

where f_w is a linear mapping from the picture space $\mathcal{P}$ to the text space $\mathcal{T} = \mathbb{R}^T$. In other words, f_w is defined as,

$$\forall p \in \mathcal{P}, f_w(p) = (w_1 \cdot p, \ldots, w_T \cdot p)$$

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

2.3 Passive-Aggressive Loss Minimization

As mentioned above, our goal is to identify the parameters w^* that minimizes $w \to L(F_w; D_{train})$. For that purpose, we rely on the *Passive-Aggressive* minimization algorithm[3] [11]. This algorithm iteratively constructs a sequence of weights $w^0, \ldots, w^m$ according to the following procedure: the first vector is set to be zero ($w^0 = 0$) and, at any iteration $i > 0$, we select the weight w^i as a trade-off between remaining close from the previous weight w^{i-1} and satisfying the i^{th} training constraint,

$$w^i = \arg\min_w \|w - w^{i-1}\|^2 + C \cdot l(F_w; q_i, p_i^+, p_i^-). \tag{3}$$

where C is the *aggressiveness* hyperparameter that controls this trade-off. This problem (3) can then be solved analytically [11], leading to:

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

After the last training iteration m, the best weight w^* is selected among $w^0, \ldots, w^m$ according to some validation data D_{valid}: $w^* = \arg\min_{w \in \{w^0, \ldots, w^m\}} L(F_w; D_{valid})$. The two hyperparameters, i.e. the aggressiveness C and the number of iterations m, are selected by cross-validation.

3 Text and Visual Features

This section describes the features used to represent text queries and pictures.

3.1 Text Features

The queries are represented with *bag-of-words* vectors, i.e. each query q is represented with a vocabulary sized vector,

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

where q_i is the weight of term i in q and T is the vocabulary size. Each term weight q_i is assigned according to the *normalized tf idf* weighting, i.e.

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

where $tf_{i,q}$ refers to the number of occurrences of i in q and $idf_i = -log(r_i)$, r_i being the fraction of training captions in which i occurs.

[3] The proof that the *Passive-Aggressive* algorithm actually minimizes the loss $L(F_w; D_{train})$ is not reported here due to space limitation but can be easily infered from [11].

3.2 Visual Features

Similarly to previous work, e.g. [1, 6], we adopt a *bag-of-visterms* representation for pictures. In this framework, the representation of a picture p is assigned according to a 4-step process. In a first step, salient regions of p are detected. Then, each region is described with a feature vector. Each of these feature vectors is then mapped to a single discrete value according to a codebook (in general, this codebook is built through k-means clustering of the set of feature vectors extracted from all training images). The picture p is then represented as an histogram over the codebook, i.e.

$$p = (vtf_{p,1}, \ldots, vtf_{p,V}), \tag{4}$$

where V is the codebook size and $vtf_{p,i}$ is the number of regions in p whose vector is mapped to the i^{th} codebook value.

In our case, we use two alternative types of visterm representation, i.e. Blob and Scale Invariant Feature Transform:

Blobs are based on the visual properties of large color-uniform regions. In this case, the salient regions are detected through a normalized cut algorithm, each region is then described by a 36-dimensional vector describing colors (18), texture (12) and shape/location (6). Region quantization is then performed according to the k-means clustering of the training regions. More details about these features can be found in [1].

SIFTs are based on the distribution of edges in regions located around salient points of the image. In this case, the salient regions are detected with a Difference-of-Gaussians detector, and each region is then described according to a 128-bin edge histogram. Like for Blobs, region quantization is also performed according to the k-means clustering of training regions. More details about these features can be found in [12].

SIFTs and Blobs have also been used jointly in our experiments. In this case, a single histogram per picture is obtained by concatenating the Blob and SIFT histograms.

Like for text representation, we do not use the vtf vector (4) directly, we instead use a representation similar to the *normalized tf idf* weighting[4], i.e.

$$p_i = \frac{vtf_{i,p} \cdot vidf_i}{\sqrt{\sum_{j=1}^{V} (vtf_{j,q} \cdot vidf_j)^2}},$$

where $vidf_i = -log(vr_i)$, vr_i referring to the fraction of training regions mapped to the i^{th} codebook vector.

[4] Due to space limitation, we do not report the preliminary experiments over validation data highlighting the advantage of this weighting strategy over standard vtf histograms.

4 Related Work

Most of the previous work in image retrieval from text queries focussed on an intermediate step, image auto-captioning, the underlying idea being to apply text retrieval techniques over the automatically inferred captions. The goal of such approaches is hence not to optimize directly a criterion related to retrieval performance but to find the most probable caption given a picture. In this context, several models have been introduced in the last decade and the following describes three of them: we present Cross-Media Relevance Model (CMRM) [3], Cross-Media Translation Table (CMTT) [4] and Probabilistic Latent Semantic Analysis (PLSA) [6]. Other models, such as Latent Dirichlet Analysis [5] or Hierarchical Mixture Model [2], could also have been present in this section. However, due to space constraints, we decided to focus on the models that have shown to be the most effective over the benchmark *Corel* dataset [1].

4.1 Cross-Media Relevance Model

The core idea of CMRM [3] is to estimate the joint probability of a term t and a test picture p^{test} 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 image 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 image, leading to:

$$P(t, p^{test}) = \sum_{p^{train} \in D_{train}} P(p^{train}) \cdot P(t | p^{train}) \prod_{i=1}^{m} P(v_i | p^{train}) \qquad (5)$$

The probability $P(p^{train})$ is then assumed to be uniform over D_{train}, while $P(t|p^{train})$ and $P(v_i|p^{train})$ are estimated through maximum likelihood with Jelinek-Mercer smoothing [3]. The probability $p(t|p^{test})$ is then simply inferred from (5) using Bayes rule. Although simple, this method has shown to yield good performance over the standard *Corel* dataset [3].

4.2 Cross-Media Translation Table

The CMTT approach is inspired from cross-lingual retrieval techniques [4]. Given a term t and a picture p^{test}, CMTT estimates $p(t|p^{test})$ according to a translation table, containing the similarities $sim(t, v)$ between any textual term t and any visterm v:

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

$v_1, \ldots, v_m$ being the visterms of p^{test}. The translation table is built from the training set D_{train} according to the following methodology: in a first step, each term t and each visterm v is represented by a $|D_{train}|$ dimensional vector in which each component i is the $tf \cdot idf$ weight of term t (or visterm v) in the i^{th} training example. The vectors of all terms and visterms are then represented as a matrix, $M = [t_1, \ldots, t_T, v_1, \ldots, v_k]$, and Singular Value Decomposition (SVD) is then applied over this matrix as a noise removal step, yielding $M' = [t'_1, \ldots, t'_T, v'_1, \ldots, v'_k]$. The similarities between a visterm v and a term t are then computed according to:

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

CMTT has been successfully applied to the Corel data. In particular, the application of SVD has shown to improve noise robustness. However, cosine similarity only allows to model simple term/visterm relationships. This limitation has been circumvented with the introduction of more complex models, like PLSA.

4.3 Probabilistic Latent Semantic Analysis

PLSA, initially introduced for text retrieval [13], has recently been applied to image retrieval problems [6]. This model assumes that the observation of a picture p and a term t are independent conditionally to a discrete latent variable $z_k = \{z_1, \ldots, z_K\}$,

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

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, which has shown to be more effective empirically [6].

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

5 Experiments and Results

In this section, we first present the experimental setup and then discuss the results.

5.1 Experimental Setup

The experiments presented in this section have been performed over the *Corel* dataset, following the setup introduced in [1]. This dataset consists of $5,000$ captioned images which are split into $4,500$ development images and 500 test images. The image captions are manual annotations, based on a vocabulary of 179 words.

As a feature extraction step, we extracted Blob and SIFT visterms relying on a codebook built through k-means clustering of the development pictures (see Section 3). For PAMIR training, we split the development set into a $4,000$ image train set ($L(F_w, D_{train})$ is minimized over this set, see Section 2) and a 500 image validation set (the number of iterations m and the aggressiveness parameter C are selected relying on this set). Since no retrieval queries were available as such for this *Corel* data, we used as queries all subsets of the 179 words which have at least one relevant image 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". Such queries have already been used in previous work, e.g. [3, 14]. Table 1 and Table 2 summarize image and query set statistics.

In order to assess PAMIR effectiveness, we used mean average precision (mAP), the standard evaluation measure in Information Retrieval benchmarks [8]. For any query, average precision is defined as the average of the precision (i.e. the percentage of relevant pictures) measured at each ranking position where a relevant picture appears and mAP corresponds to the mean of average precision

Table 3. Mean 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**

over the Q_{test} set. For the sake of comparison, we also report the performance of CMRM, CMTT and PLSA that we trained and evaluated according to the same setup.

5.2 Overall Performance

Table 3 reports the mean average precision for the CMRM, CMTT, PLSA and PAMIR models when Blobs, SIFTs and their combined representation are used. The PAMIR model achieves the best retrieval performance for the three image representations, with a significant improvement according to the Wilcoxon signed-rank test at 95% confidence over the three other models for SIFTs and Blobs+SIFTs (this outcome is indicated by bold values in the tables). Although it does not contain any color information, the SIFT representation leads to a more accurate ranking of the test images for the PLSA and PAMIR models than the Blob representation (27% and 34% relative improvement respectively). This might be explained considering the difference between the two representations, which not only relies in the region descriptors, but also in the number, and the size of the considered regions (see Section 3). While the Blobs representation only consists of a maximum of ten regions, the average number of regions sampled per image with the Difference of Gaussians point detector is around 230 in our dataset. The SIFT representation therefore presents richer statistics than the Blob representation, and these statistics seems better exploited by PLSA and PAMIR.

The two representations are complementary, and their combination interestingly achieves a higher score than the Blob or SIFT representation individually for the CMRM, PLSA, and PAMIR models. The relative improvement of Blobs+SIFTs over SIFTs is 41% for the CMRM model, 36% for the PLSA model, and 35% for the PAMIR model. Only CMTT fails to take advantage of the combination, and achieves a similar performance with the Blobs+SIFTs and Blobs. The poor performance of CMTT model over SIFTs might explain the difference. The PAMIR model does take the best benefit of the combined representation, and outperforms the second best model, PLSA, with a 29% relative improvement. These results justify the combination of a small set of large, color-based regions (Blobs) with a larger set of small, texture-based regions (SIFTs) to represent an image.

A majority of studies [1, 6, 4, 2] evaluates the retrieval performance based on single-word queries. We therefore compare the four models using the three image representations for the subset of single-word queries in Table 4. On this set of

Table 4. Mean 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**

single-word queries, the CMTT model achieves the best performance when the Blob representation is used, and the PAMIR model performs the best image ranking for the SIFT and Blob+SIFT representation. The relative increase in performance with respect to PLSA, the second best model, is 39% and 25% for SIFTs and Blobs+SIFTs respectively.

Comparing Tables 3 and 4, one should remark that the performance is higher for single-word queries. This result can be explained by the number of relevant pictures per query. The subset of 179 single-word queries has a higher average number of relevant images (9.4) than the set of all $2,241$ queries (2.4). This means that these queries correspond to an easier retrieval problem [8], that naturally results in higher mean average precision values. Moreover, the words appearing in queries with many relevant pictures occur more frequently in the training data, allowing the model to achieve better generalization performance. The influence of the number of relevant images on PAMIR results is shown in Table 5. The single-word queries are divided into three sets, defined by the number of relevant images per query. The mean of the average precision of the queries within each range indicates that the average precision is higher for queries with more relevant documents, which confirms the above explanation.

We showed that the PAMIR model takes the best advantage of the Blobs+SIFTs combination, outperforming the PLSA-based generative model and other approaches significantly. The good performance of the PAMIR model justifies the use of a ranking-based criterion for retrieval applications.

5.3 Per-query Performance

The mean average precision measure summarizes the overall retrieval performance of a model in a single number, but does not indicate the per-query performance. To have a more complete evaluation, we compare the average precision of each single-word query obtained with the PAMIR model over the different types

Table 5. Mean average precision (mAP) in percent obtained with PAMIR for Blobs+SIFTs for three sets of single-word queries defined by the number of relevant images.

query range	# queries	mAP (%)
$0 < \#rel.pic. \leq 2$	47	15.5
$2 < \#rel.pic. \leq 7$	69	26.7
$7 < \#rel.pic.$	63	46.5

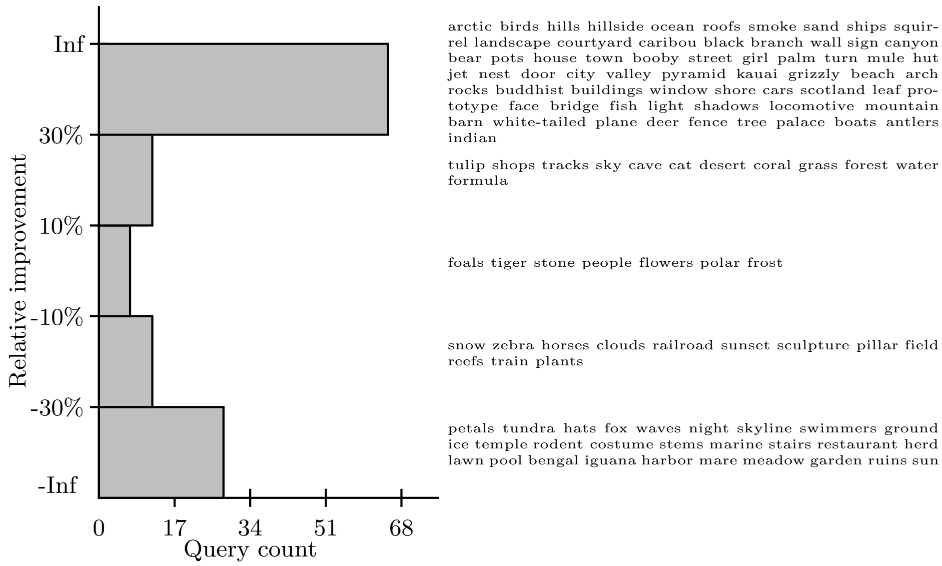

Fig. 1. Histogram of the relative increase of the average precision of single-word queries between the SIFTs and Blobs representations. The words corresponding to each bin are shown on the right.

of image representation. We also propose to compare the results of PAMIR and PLSA, the second best model, on a per-query basis.

Figure 1 shows the relative increase in performance between SIFT and Blob representations. The histogram shows five ranges of relative improvement, with the corresponding queries on the right. In this figure, we consider only the queries that correspond to a minimum of 10% of average precision for one of the two representations in order to avoid unreliable measurements of relative improvement. Among the 124 resulting single-word queries, 65 are improved by more than 30% when SIFTs instead of Blobs are used. For instance, for queries like *house*, *town*, *street*, *city*, *arch*, *buildings*, *window*, and *bridge*, images are ranked with a higher average precision when represented with SIFTs instead of Blobs features. As these concepts naturally correspond to local edge structures, it seems consistent that SIFTs better capture the corresponding image content. It is more surprising that the average precision of single-word queries like *ocean* or *black* is improved by more than 30% when SIFTs instead of Blobs features are used. The opposite trend is also observed with other queries, for which the PAMIR model achieves a higher score with the Blob representation. For 28 queries, the relative gain obtained by using Blobs instead of SIFTs is over 30%. The ranking of color-based concepts like *sun*, *ice*, *night*, and *garden* is learned with a higher accuracy by the PAMIR model when images are depicted with Blobs rather than SIFT features, which seems consistent. The fact that the queries *temple* and *restaurant* are improved when Blobs instead of SIFTs are used is less intuitive.

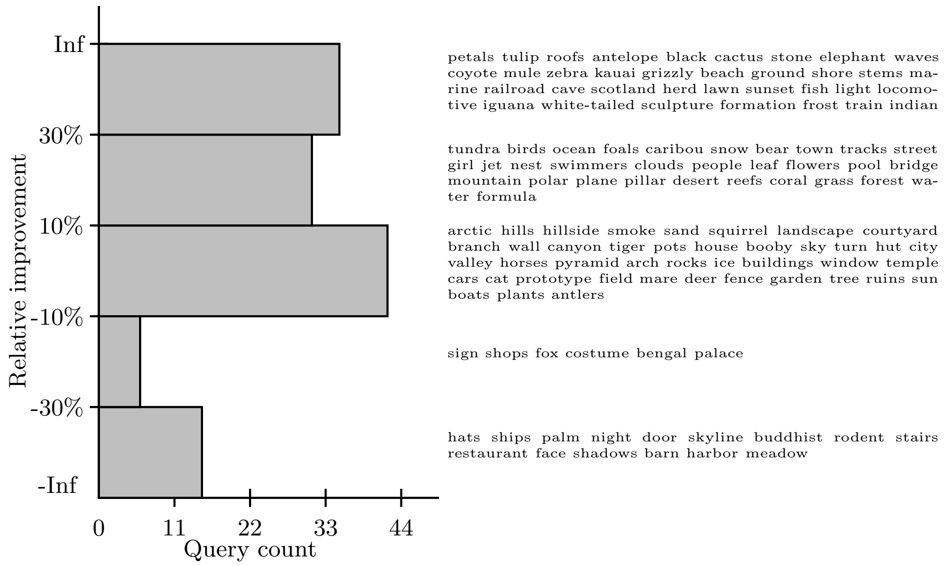

Fig. 2. Histogram of the relative increase of the average precision of single-word queries obtained with the Blobs+SIFTs representation and the best average precision achieved with the Blobs and SIFTs representations. The words corresponding to each bin are shown on the right.

As shown in Tables 3 and 4, the combination of the two representations improves the retrieval performance of the PAMIR model for all queries on average. To have an indication of the difference in performance at query level, Figure 2 shows the histogram of the relative improvement in average precision obtained with the Blobs+SIFTs over the best average precision obtained between Blobs and SIFTs individually. Note that this second performance is only theoretical, given that the *best* representation is chosen on the test data for each query. As for Table 1, only single-word queries resulting in a 10% minimum average precision with one of the two representations are considered. While 42 out of the 129 considered queries do not significantly benefit from the combined representation, with a relative difference between 10% and -10%, the average precision of 66 words is improved by more than 10% when Blobs+SIFTs is used. Moreover, the increase is over 30% for 35 single-word queries. The words *ocean* and *black*, that were surprisingly better represented by SIFT instead of Blob features (see Table 1), achieve a higher average precision when the SIFTs representation is completed with the Blobs features. This confirms the intuition that these specific queries should benefit from some color-based visual information. Although the best representation between SIFTs and Blobs has been selected a-posteriori for this evaluation, only 21 queries suffer a performance loss greater than 10% when using the combination instead of this unrealistic individual feature setup. This result hence highlights the complementarity of Blob and SIFT representation.

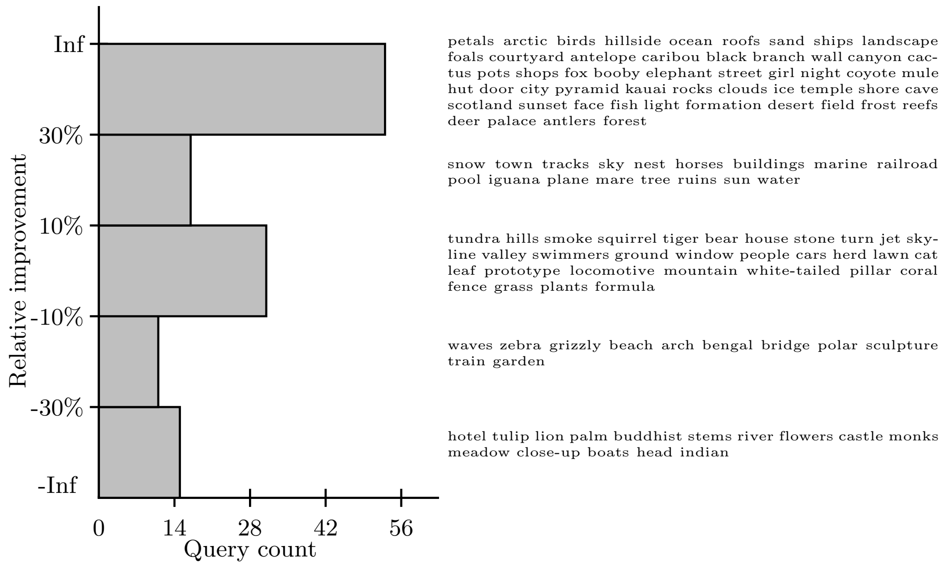

Fig. 3. Histogram of the relative increase of the average precision of single-word queries between the PAMIR and the PLSA model, using the Blobs+SIFTs representation.The words corresponding to each bin are shown on the right.

Keeping this combined feature setup, we propose to compare the performance of PAMIR with the best alternative, PLSA, on a per-query basis to have a deeper understanding of the difference between both models. Figure 3 shows the relative improvement in average precision for single-word queries between the PAMIR and the PLSA models, for the Blobs+SIFTs representation. Like for the above histograms, only the queries with a minimum average precision of 10% for one of the two models are considered to prevent unreliable measurements of relative improvement. This leads to 127 queries. Out of these, the ranking of 70 queries is improved by more than 10% when PAMIR instead of PLSA is used, while 26 queries only are better ranked by PLSA by more than 10%. The PAMIR model improves the ranking of 53 queries by more than 30% relative improvement. This further confirms the result of the Wilcoxon signed-rank test which concluded that PAMIR advantage is consistent over the query set. An illustration of the rankings obtained by PLSA and PAMIR is shown in Figure 4 for the queries *pillar* and *landscape*, which are respectively improved by more than 10% and 30% by the PAMIR model. Note that only the first five top-ranked images are shown, which does not necessarily reflect the whole ranking performance measured by the average precision measure. For the *pillar* query, both models retrieve relevant images in the top-five, except for the last image retrieved by PLSA. The second query shows the case where the ranking obtained by the PAMIR model is clearly better for the top-five images. The first two and the fourth images retrieved by PLSA are not related to the *landscape* concept, while only the fourth image

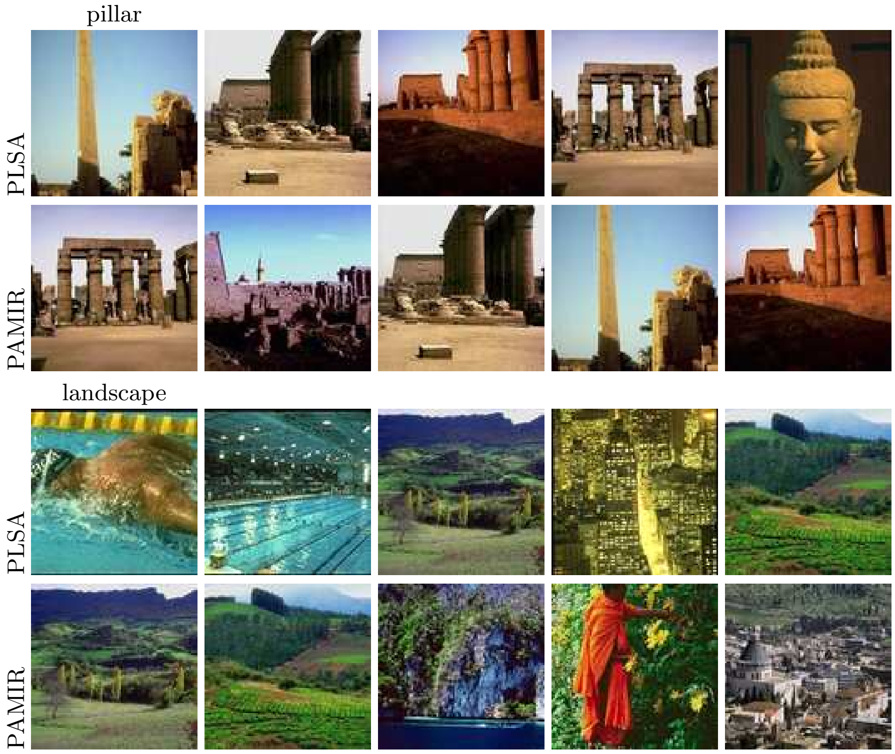

Fig. 4. First five images retrieved with the PLSA and the PAMIR models for the queries *pillar* and *landscape*.

retrieved by the PAMIR model is not a *landscape* image. These examples confirm the advantage of PAMIR over PLSA, showing the practical benefit of using a learning procedure appropriate to the image retrieval problem.

6 Conclusions

In this work, a discriminative model for the retrieval of pictures from text queries has been proposed. This model relies on the recently proposed Passive-Aggressive algorithm for training [11] and its parameters are selected to minimize a loss related to the ranking performance over a set of training queries. The choice of such a loss is motivated by recent work in the context of text retrieval [9, 10]. The experiments performed over the *Corel* dataset show that the advantage of discriminative approaches observed for text data translates to image retrieval: the proposed model PAMIR is reported to yield significantly better results than generative models that consistitute the state-of-the-art (e.g. PAMIR mean average precision is 21.6% when Blob and SIFT features are used, compared to 16.7% for PLSA, the second best model).

These results are promising and this work yield several possible future research directions. For instance, other parameterization could be investigated: as any passive-aggressive algorithm [11], the PAMIR model could rely on non-linear kernels, allowing the application of kernels which avoid the feature quantification step, such as [15]. Another extension of this work would be to modify PAMIR such that it could be applied over much larger datasets, where the application of any learning procedure, generative or discriminative, is challenging.

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

1. 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
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. Pan, J.Y., Yang, H.J., Duygulu, P., Faloutsos, C.: Automatic image captioning. In: International Conference on Multimedia and Expo (ICME). (2004) 1987–1990
5. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. Journal of Machine Learning Research **3** (2003) 993–1022
6. Monay, F., Gatica-Perez, D.: PLSA-based image auto-annotation: constraining the latent space. In: ACM Multimedia. (2004) 348–351
7. Vapnik, V.: Statistical Learning Theory. John Wiley and Sons, New York (1998)
8. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison Wesley, Harlow, England (1999)
9. Joachims, T.: Optimizing search engines using clickthrough data. In: International Conference on Knowledge Discovery and Data Mining (KDD). (2002)
10. Grangier, D., Bengio, S.: Exploiting hyperlinks to learn a retrieval model. In: NIPS Workshop on Learning to Rank. (2005) 12–17
11. Crammer, K., Dekel, O., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. In: Conference on Advances in Neural Information Processing Systems (NIPS). (2003)
12. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (IJCV) **60**(2) (2004) 91–110
13. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. Machine Learning **42** (2001) 177–196
14. Lavrenko, V., Manmatha, R., Jeon, J.: A model for learning the semantics of pictures. In: Conference on Advances in Neural Information Processing Systems (NIPS). (2003)
15. Wallraven, C., Caputo, B.: Recognition with local features: the kernel recipe. In: International Conference on Computer Vision (ICCV). (2003)