Joint Image and Word Sense Discrimination for Image Retrieval

Aurelien Lucchi[1,2] and Jason Weston[1]

[1] Google, New York, USA
[2] EPFL, Lausanne, Switzerland

Abstract. We study the task of learning to rank images given a text query, a problem that is complicated by the issue of multiple senses. That is, the senses of interest are typically the visually distinct concepts that a user wishes to retrieve. In this paper, we propose to learn a ranking function that optimizes the ranking cost of interest and simultaneously discovers the disambiguated senses of the query that are optimal for the supervised task. Note that no supervised information is given about the senses. Experiments performed on web images and the ImageNet dataset show that using our approach leads to a clear gain in performance.

1 Introduction

Ranking an image database given a text query (i.e. the image retrieval task) has found growing importance due to its popularization through internet image search engines. Users' text queries typically contain only one or two words so their intent is not always clear, often leaving several possible interpretations, or senses, that should be retrieved. This provides a machine learning system with two related difficulties: first to automatically understand what are the possible senses of the text query. Secondly, to understand for any given image in the database which sense, if any, is relevant to decide whether to retrieve it.

Apart from being of academic interest, we believe that a surprisingly large number of random queries can be positively impacted by image and word sense discrimination. Our reasoning is the following: although many queries might be considered to have one sense in terms of their dictionary definition, they can often have several senses in terms of images. For example, the query "France" has relevant images from several distinct senses: images of the French flag, maps of the country, images of cities (such as Paris) and images of monuments (such as the Eiffel tower) to name a few.

In this work, we propose a model that tries to explicitly learn the senses of a given query that optimizes an image ranking cost function jointly over all senses. The training data for our image ranking task is of the standard types: either click-through based (how many users clicked on a given image for a given query) or human-annotated (labeled as to whether this image is relevant for this query). Hence, our data does not contain any information about the number or type of senses per query. We thus propose to learn the senses in the following way: model the desired ranking function as containing S components for the S senses, and

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part I, LNCS 7572, pp. 130–143, 2012.

optimize the overall ranking loss for all senses jointly (and also optimize S). We show that this approach provides improved ranking metrics over systems that do not model the senses both for random queries and particularly for queries with multiple senses. Simultaneously, our work provides an interpretable model where we can examine the senses that have been learned.

2 Related Work

Work on image retrieval covers a large number of methods. Almost all systems comprise of two steps: choosing a feature representation for the data (i.e. how to represent the images) and then a learning scheme to connect the query text to the images, as represented by their features. One of the simplest methods to understand, that also happens to bring very reasonable performance, is that of large margin rank-based learning of a linear model in the chosen image feature space, such as implemented by the PAMIR system [1] which uses online gradient updates. PAMIR was shown to perform well compared to several other models such as SVM [2], PLSA [3] and Cross-Media Relevance Models [4]. Nearest neighbor models [5] and metric learning models combined with nearest neighbor have more recently been proposed [6]. Finally, a neural network approach that avoids an explicit feature map design step by learning the feature representation was described in [7]. To our knowledge, none of the above systems try to learn a model that takes into account the senses of the query text explicitly (although there are some such systems, described below).

In the field of natural language processing, the tasks of word sense disambiguation and word sense discrimination (focusing on text alone) are well studied. Word sense disambiguation is perhaps the more well known task of selecting the sense of a word given its context from a predefined set of possibilities. Word sense discrimination, which is more related to our goals in this work, involves dividing the usages of a word into different meanings *given only unannotated corpora*. Word sense disambiguation typically uses a supervised signal of some kind, either from resources like WordNet [8], or from labeled text (e.g. Senseval[1] data). Unsupervised word sense discrimination typically tries to cluster senses using context, for example [9,10]. Word sense discrimination has also been applied to the task of information retrieval for text documents, see e.g. [11]. See [12] or [13] for an overview of this field.

The problem of polysemy with images is widely recognized as being difficult [14], [15] and still largely unsolved. Recently, several authors proposed methods using both text and images for word sense disambiguation. The authors of [16] performed spectral clustering in both the text and image domain and evaluated how well the clusters matched different senses. However, as recognized in [17], this method fails to assign a sense label to images. Instead, [17] proposed to compute the likelihood of a particular word sense for a given web image but their method requires the use of a dictionary model which is additional information that we do not consider in this paper. Another approach,

[1] `http://www.senseval.org/`

proposed by [18], uses Wikipedia to find the senses of a word. Their method uses LDA to learn latent visual topics and then learns a model of the wiki-senses in the latent space. The authors of [19] also propose to use the context words in documents to infer the senses of images, something we do not consider here. Finally, [20] proposes to model the joint probability of words and image regions, which requires labels associated to each region, which is again something not available in our setup.

The model introduced in the next section proposes to treat different word senses as latent variables. Although the optimization could be done with Latent Structured SVM [21], this would require an expensive loss-augmented inference step to find the most violated constraint over the entire set of negative images. Other latent models like Multiple instance ranking (MIRank) [22] could also be used. However, given the large number of training images used in our experiments, we found the stochastic gradient descent procedure described in section 3 to be very well suited, mostly because it is easy to implement, highly scalable, and can be trained online.

3 Model

We design our model for the standard image retrieval task which is defined as follows: we are given a set of text-based queries and a set of training data associated with each query. The training data consists of positive training images $x^+ \in \mathcal{X}_q^+$ that are relevant to the query q and negative training images $x^- \in \mathcal{X}_q^-$ that are irrelevant. The goal is, given a query, to rank a set of (previously unseen) test images such that relevant images are ranked at the top of the list, above the irrelevant ones.

Even though no information about the senses of a query (and hence the senses of the relevant images) is given in the training data, a query such as "jaguar" has at least two kinds of relevant images: images of cars, and of *Panthera* (big cats). The key aspect of our method is that we want to model that phenomenon. For that goal, we define a ranking function, per query, and per sense:

$$f_{q,s}(x) = W_{q,s} \cdot x \tag{1}$$

where q is the query, $x \in R^D$ is an input image (represented with features in a D-dimensional space), $W_{q,s}$ are the parameters of the model and s is the s^{th} sense for this query. The ranking function returns a real-valued output that measures the degree of match between a query q and an image x, where a large value means a higher match.

For a given query, after we scored an image in terms of its relevance match with respect to each sense, we then combine those scores to give an overall relevance match, independent of sense:

$$f_q(x) = \max_{s \in S(q)} f_{q,s}(x) = \max_{s \in S(q)} W_{q,s} \cdot x \tag{2}$$

where $S(q)$ is the number of semantic classes (and hence hyperplanes) that are used for the given query q (i.e., the number of discovered senses is variable

depending on the query). Our justification for using the max in Eq. 2 is because if an image is relevant with respect to any one of the senses (and typically it is only relevant for one, e.g. the "jaguar" example above), then it is indeed relevant for the query. Also, note that using a sum instead of a max would gain nothing as that is the same as a single hyperplane with parameters equal to the sum of the hyperplanes $W_q^{sum} = \sum_{s \in S(q)} W_{q,s}$. Finally, we then rank the entire set of images by their matching scores using $f_q(x)$.

To train our model we need to deal with two of its non-standard properties: (i) we do not know which sense an image belongs to, so the hyperplane it should be assigned to is unknown without the max function; and (ii) we do not know how many total senses $S(q)$ there are for each query. We solve problem (ii) by cross-validation, trying each value of $S(q)$ (in our experiments, we consider up to 5 possible senses) and selecting the one that does best. To solve problem (i) we directly train our model, with a fixed number of senses $S(q)$, so that the maximum sense score of a positive image is greater than the maximum sense score for a negative image, plus some margin:

$$\max_{s \in S(q)} f_{q,s}(x^+) > \max_{s \in S(q)} f_{q,s}(x^-) + 1 \quad \forall x^+ \neq x^-. \tag{3}$$

We also consider regularizing the weight vectors by enforcing the following constraints: $||W_{q,s}|| \leq C, \ \forall q, s$ where C is a constant whose value is determined empirically. That is, our overall optimization problem is: minimize

$$\sum_{q, x^+ \in \mathcal{X}_q^+, x^- \in \mathcal{X}_q^-} \xi_{(q,x^+,x^-)}$$

subject to

$$\max_{s \in S(q)} f_{q,s}(x^+) > \max_{s \in S(q)} f_{q,s}(x^-) + 1 - \xi_{(q,x^+,x^-)}$$

$$\forall q, x^+ \in \mathcal{X}_q^+, x^- \in \mathcal{X}_q^-, ||W_{q,s}|| \leq C, \qquad \forall q, s$$

$$\xi_{(q,x^+,x^-)} \geq 0, \qquad \forall q, x^+, x^-.$$

where the slack variables ξ measure the margin-based ranking error per constraint. As all the parameters are actually decoupled between queries q, we can learn the parameters independently per query (and hence train in parallel). We choose Stochastic Gradient Descent (SGD) as the optimization method, giving us one more free parameter to optimize over, the learning rate λ.[2] The steps for training our system, which we call IMAX, are given in Algorithm 1.

3.1 Analysis of the Senses Learned by the Model

After learning we can analyze the word/image senses our model has learned. For any given query, we can first look at the value $S(q)$ that is learned (i.e.,

[2] SGD has known convergence properties when decreasing the learning rate over time. In our work, we keep the learning rate fixed, which has still been shown to converge to within a certain distance of the optimal [23].

Algorithm 1. IMAX training algorithm

for each query q **do**
  Input: labeled data $x^+ \in \mathcal{X}_q^+$ and $x^- \in \mathcal{X}_q^-$ (specific for query q).
  Initialize the weights $W(q,s)$ randomly with mean 0 and standard deviation $\frac{1}{\sqrt{D}}$ for all q and s.
  for each $S(q)$ to try (e.g. $S(q) = 1, \ldots, 5$) **do**
    repeat
      Pick a random positive example x^+.
      Let $s^+ = \text{argmax}_{s \in S(q)} W_{q,s} \cdot x^+$.
      Pick a random negative example x^-.
      Let $s^- = \text{argmax}_{s \in S(q)} W_{q,s} \cdot x^-$.
      if $f_{q,s^+}(x^+) < f_{q,s^-}(x^-) + 1$ **then**
        Make a gradient step to minimize:
        $|1 - f_{q,s^+}(x^+) + f_{q,s^-}(x^-)|_+$, i.e:
        Let $W_{q,s^+} \leftarrow W_{q,s^+} + \lambda x^+$.
        Let $W_{q,s^-} \leftarrow W_{q,s^-} - \lambda x^-$.
        Project weights to enforce constraints $||W_{q,s}|| \leq C$:
        for $s' \in \{s^+, s^-\}$ **do**
          if $||W_{q,s'}|| > C$ **then**
            Let $W_{q,s'} \leftarrow C W_{q,s'} / ||W_{q,s'}||$.
          end if
        end for
      end if
    until validation error does not improve.
  end for
  Keep the model (with the value of $S(q)$) with the best validation error.
end for

the number of senses that are chosen). Secondly, we can rank the database of images by each sense submodel of the model, i.e. we can produce a ranked list of images for each $f(q,s)$. Presumably, each submodel should be identifying a different sense/aspect of the query concept. Finally, in order to further understand the actual word sense that each submodel is identifying we propose the following technique. We first find the top N most similar hyperplanes to $W(q,s)$ by measuring $\frac{W(q,s) \cdot W(q',s')}{||W(q,s)|| \cdot ||W(q',s'))||}$ for all q' where $q \neq q'$ and all $s' \in S(q')$. That is, by measuring the cosine similarity between hyperplanes we find the most similar concepts from individual senses from other queries. Then, we can return the query text for each of the top-scoring q'. The idea is that if we find multiple senses for the query "jaguar" then one of the senses should be similar to queries like "lion" and "tiger" and the other sense should be similar to queries like "Audi" and "BMW". If so, we are truly "discovering" the senses of the query.

4 Experimental Results

We conducted experiments on two datasets: a proprietary web dataset collected via user clicks on images for given queries, and the ImageNet dataset [24].

4.1 Baselines

Linear Ranker. Our main baseline is a large margin ranking model in the style of PAMIR [1] which was shown to perform well compared to SVM, PLSA and other methods on image ranking tasks. PAMIR uses the passive-aggressive weight updates, but for ease of comparison we follow the process of Algorithm 1 (i.e., the same algorithm except that $S(q) = 1$ for all q).

Max-Avg relaxation. We relax the IMAX algorithm by minimizing:

$$\sum_{q,r\in S(q),x^+\in\mathcal{X}_q^+,x^-\in\mathcal{X}_q^-} \xi_{(q,r,x^+,x^-)}$$

$$\text{subject to}$$

$$\max_{s\in S(q)} f_{q,s}(x^+) > f_{q,r}(x^-) + 1 - \xi_{(q,r,x^+,x^-)},$$

$$\forall q, r \in S(q), x^+ \in \mathcal{X}_q^+, x^- \in \mathcal{X}_q^-.$$

$$||W_{q,s}|| \leq C, \quad \forall q, s.$$

$$\xi_{(q,r,x^+,x^-)} \geq 0, \qquad \forall q, r, x^+, x^-.$$

That is, we have the same algorithm as IMAX except that the max operation over negative examples is not present in the constraints, but instead one separate constraint per sense is used. The purpose of this baseline is to show the importance of the max operation during the training stage.

Avg-Avg Relaxation. We considered the further simplification of removing the max operations altogether. This leaves us with the following optimization problem: minimize

$$\sum_{q,r\in S(q),r'\in S(q),x^+\in\mathcal{X}_q^+,x^-\in\mathcal{X}_q^-} \xi_{(q,r,r',x^+,x^-)}$$

$$\text{subject to}$$

$$f_{q,r}(x^+) > f_{q,r'}(x^-) + 1 - \xi_{(q,r,r',x^+,x^-)},$$

$$\forall q, r \in S(q), r' \in S(q), x^+ \in \mathcal{X}_q^+, x^- \in \mathcal{X}_q^-.$$

$$||W_{q,s}|| \leq C, \quad \forall q, s,$$

$$\xi_{(q,r,r',x^+,x^-)} \geq 0, \qquad \forall q, r, r', x^+, x^-.$$

Without any max operations during training at all, the learning of the parameters for each sense becomes decoupled and this is equivalent to learning an ensemble of $S(q)$ rankers.

4.2 Image Representation

We used the same feature representation adopted in [25]. Various spatial [26] and multiscale color and texton histograms [27] are combined for a total of

about 5×10^5 dimensions. The descriptors being somewhat sparse by nature (around $50,000$ non-zero weights per image), we perform Kernel PCA [28] on the combined features using the intersection kernel [29] to produce a 100 dimensional input vector. [25] showed that training on these features for the related task of image annotation outperforms sparse bag-of-visual term features, e.g. as used in [1]. In fact, we tried some preliminary experiments on our own task and came to the same conclusions, which is not surprising as the tasks are quite related.

4.3 Experiments on Web Data

We had access to a proprietary database of images taken from the web, where for each (query, image) pair the number of anonymized user clicks have been recorded, an indicator that users believe this image is relevant for the given query. We randomly extracted a set of 565 queries and 96,812 images totaled over all queries that had been clicked at least 50 times. For each query, we split the corresponding set of images into three sets for training (60%), validation (20%) and test sets (20%). We selected a further 100,000 images at random from the set of all possible images to use as negative examples for any given query, which again is split into train, validation and test portions.

4.4 Experiments on ImageNet

The ImageNet dataset [24] (Spring 2010 release) contains 12,184,113 images and 17,624 synsets (categories) organized according to the semantic hierarchy of WordNet [8]. Multiple words can belong to each synset, e.g. "cell" belongs to a synset for cell phones and jail cells. We randomly chose 389 queries among the phrases that belong to multiple synsets, corresponding to a total of 216,486 images. For each query, we again split the set of images into training (60%), validation (20%) and test sets (20%). We again selected a further 100,000 images at random from the set of all possible images to use as negative examples for any given query, which is split into train, validation and test portions.

4.5 Evaluation Metrics

We report results averaged over all queries using two evaluation metrics: AUC and precision@k. The pairwise-ranking loss, or AUC, is measured per-query as: $\frac{1}{|\mathcal{X}^+||\mathcal{X}^-|} \sum_{x^+ \in \mathcal{X}_q^+, x^- \in \mathcal{X}_q^-} I(f_q(x^-) > f_q(x^+))$. Precision@k is calculated by first ranking all images (both positive and negative) for a given query, and then looking at the top k ranked images: $p@k(q) = \frac{\sum_{i=1,\ldots,k} I(x_i \in \mathcal{X}_q^+)}{k}$ where x_1 is the top ranked image, x_2 is in second position and so on.

4.6 Results

Main Ranking Results. A summary of the main ranking evaluation results of our experiments is given in Table 1. IMax outperforms Linear rankers on both

Table 1. Summary of IMAX test results compared to the baseline methods

	ImageNet		Web-data	
Algorithm	AUC	p@10	AUC	p@10
IMAX	7.7%	70.37%	7.4%	64.53%
LINEAR ranker	9.1%	65.60%	7.9%	60.21%
AVG-AVG relaxation	8.7%	66.46%	8.1%	58.93%
MAX-AVG relaxation	8.3%	67.99%	7.7%	62.61%

Table 2. AUC loss averaged over queries with the predicted number of senses S on (a) ImageNet and (b) Web data. For a small fraction of queries (6% for ImageNet and 15% for the Web data), IMAX predicts only one sense and hence gets the same error rate as LINEAR. For $S > 1$, IMAX outperforms LINEAR.

(a)

S	Num. queries	LINEAR AUC	IMAX AUC	Gain
1	23	4.67	4.67	+0.00
2	42	9.79	8.76	+1.03
3	68	9.63	8.44	+1.19
4	108	9.09	7.48	+1.61
5	118	9.52	7.70	+1.82

(b)

S	Num. queries	LINEAR AUC	IMAX AUC	Gain
1	71	7.24	7.24	+0.00
2	118	7.53	7.23	+0.30
3	107	8.11	7.59	+0.52
4	138	8.18	7.54	+0.64
5	131	8.22	7.67	+0.55

datasets for both metrics, and also outperforms the relaxed optimization problems that approximate it. The improvement on ImageNet (e.g. from 9.1% to 7.7% AUC) is larger than on Web Data (7.7% to 7.4% AUC) which can be explained by the fact that for ImageNet we deliberately selected ambiguous queries, whereas the Web Data are randomly sampled queries. The fact that our model makes an improvement over a broad range of queries seems quite encouraging, and we will analyze this further later in this Section.

Approximate Optimization Baselines. The two relaxed optimization problems that approximate IMAX perform worse. In fact, AVG-AVG, which avoids using the max function completely during training brings no gains compared to a LINEAR ranker, we believe it fails to recover the senses. The MAX-AVG relaxation on the other hand, which does use the max but only for positive examples, brings roughly half the gains of IMAX. This shows the importance of optimizing the right function, rather than approximating it. We note that the max function is often approximated by the sum (see e.g. the literature on multiclass classifiers [30,31]) but in our setting we believe the use of the max function is essential.

Ranking Results And Sense Discrimination. On the Web Data, we observe a gain in terms of AUC compared to the LINEAR ranker on 59% of the queries, losses on 27% and draws on 13%. Breakdowns of the number of queries that are predicted S senses by IMAX, together with the AUC loss for only that subset

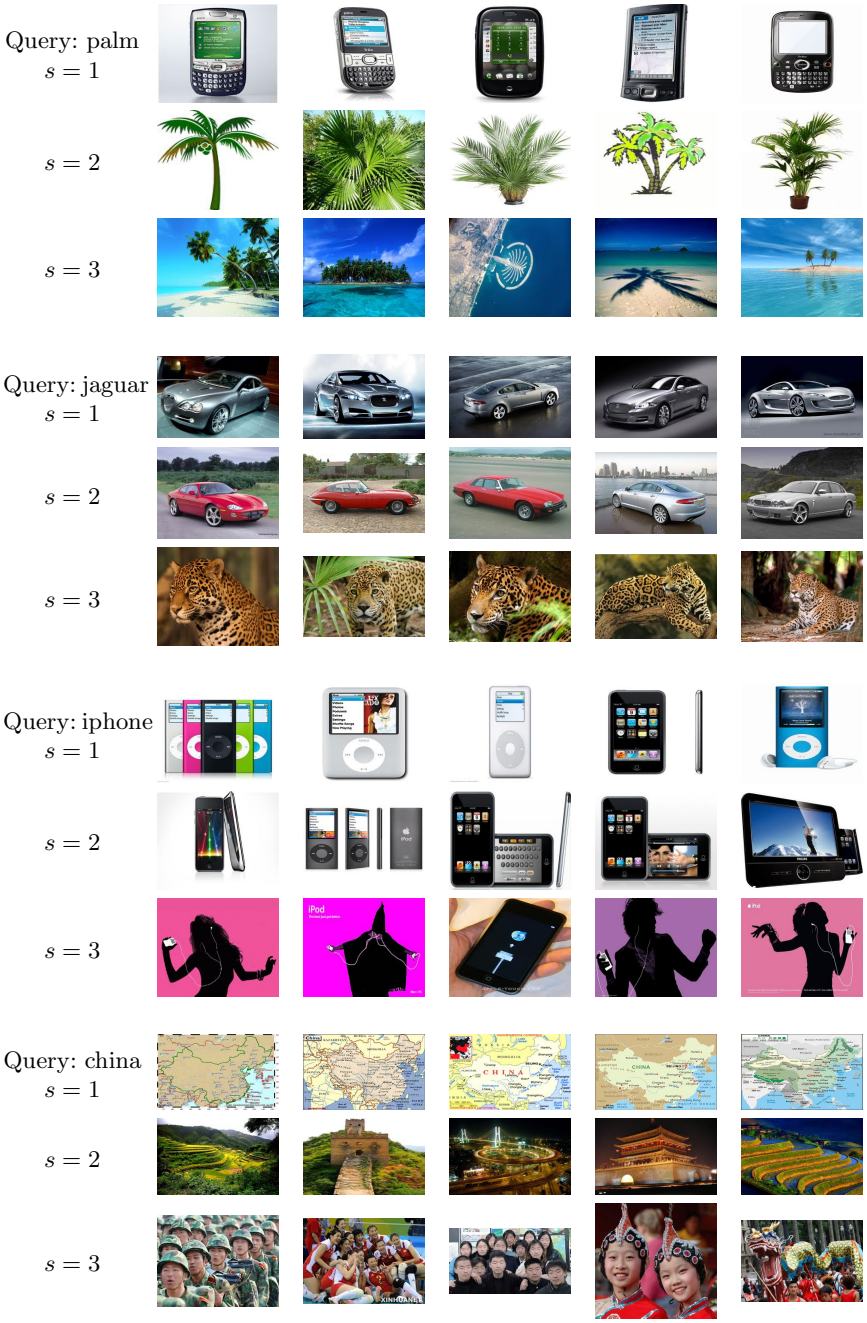

Fig. 1. The images returned by the IMAX ranking functions $f(q, s)$ for three discovered senses $s = 1, \ldots, 3$ of the queries "palm", "jaguar", "iphone" and "china" from Web Data. Different senses focus on different meanings, e.g. *panthera* and cars for "jaguar", phones and trees for "palm" or maps, scenes and people for "china".

Table 3. Nearest annotations learnt by IMAX for two queries: "jaguar" and "palm". IMAX clearly captures different senses, see Figure 1 for the corresponding images.

jaguar $s = 1$	jaguar logo, jaguar xf, mazda, jaguar xk, jaguar xj, chrysler 300m, jaguar xkr, porsche, toyota, hyundai, aston martin vanquish, e coupe, citroen metropolis, 911 turbo, mclaren mercedes, vw passat 2011, bugatti 2010.
jaguar $s = 2$	seat ibiza 2010, volkswagen polo, peugeot 308 cc, challenger 2010, tengerpart, citroen c4 tuning, iarna, polo 9n, yves tanguy, 308 cc, parachute, duvar sticker, asx, toyota yaris, seat toledo, seat ibiza st, honda accord coupe, hanna barbera, corolla 2011, cyd charisse.
jaguar $s = 3$	jaguar animal, bengal tiger, tigar, amur leopard, harimau, tiger pictures, gepard, tijgers, leopardos, bengal tigers, big cats, cheetah, tigre.
palm $s = 1$	blackberry, lg gd900, future phones, blackberry 9800, blackberry 9800 torch, smartphone, blackberry curve, nokia e, nokia phones, lg phones, cellulari nokia, nokia, nokia mobile phones, blackberry pearl, nokia mobile, lg crystal, smartphones.
palm $s = 2$	palmier, palm tree, coconut tree, money tree, dracaena, palme, baum, olive tree, tree clip art, tree clipart, baobab, dracena, palma, palm tree clip art, palmera, palms, green flowers, palm trees, palmeras.
palm $s = 3$	palmenstrand, beautiful beaches, playas paradisiacas, palms, beaches, lagoon, tropical beach, maldiverna, polinesia, tropical beaches, beach wallpaper, beautiful beach, praias, florida keys, paisajes de playas, playas del caribe, ocean wallpaper, karibik, tropical islands, playas.

of the queries, are given in Table 2. One can see that the more senses IMAX predicts for a query, the bigger the gain over the LINEAR ranker. Although this effect tails off at $S = 5$ on the Web Data, it does not on ImageNet.

A further breakdown of results can be found in Tables 4 and 5, which show the AUC for training, validation and test sets. We report results both for the selected number of senses chosen by validation error (Best s) and by fixing s to be the same value for every query (note, "Best s" is variable per-query). Most of the gains are achieved from $s = 1$ to $s = 2$, although the error is still decreasing slightly even at fixed $s = 5$. Similar conclusions can be made about p@k from Tables 6 and 7. The top wins and losses per query are given in Table 8. On ImageNet, the top wins are much bigger in terms of gain than the worst losses are in terms of negative gain (loss). For the Web Data, the wins are clearly often multiple meaning queries such as "bass" (guitar, fish), "ape" (animal, footwear, vehicle), "axe" (deoderant, weapon) and "fox" (animal, Megan Fox). The next subsection provides a detailed analysis for some particular queries.

Analyzing the Senses Learned by the Model. Figures 1 and 2 show the images returned by the IMAX submodel ranking functions for four queries from Web Data ("jaguar", "palm", "iphone" and "china") and one query from ImageNet, "cell". We show the image ranking given by $f(q, s)$ for three of the senses $s = 1, \ldots, 3$ (the combined ranking $f(q)$ is a combination of these using the max

Table 4. AUC loss for IMAX (our method) and the baselines on the Web Data

	Data set	$s = 1$	$s = 2$	$s = 3$	$s = 4$	$s = 5$	Best s
IMAX	Training	5.2%	3.7%	3.0%	2.8%	2.6%	2.8%
	Validation	7.9%	7.7%	7.6%	7.5%	7.5%	6.9%
	Test set	7.9%	7.6%	7.6%	7.5%	7.5%	7.4%
MAX-AVG	Training	5.2%	4.3%	3.9%	3.8%	3.6%	4.1%
	Validation	7.9%	8.0%	8.0%	8.0%	8.1%	7.2%
	Test set	7.9%	7.8%	7.8%	7.8%	7.9%	7.7%
AVG-AVG	Training	5.2%	5.2%	5.3%	5.3%	5.3%	5.2%
	Validation	7.9%	7.9%	7.9%	7.9%	7.9%	7.5%
	Test set	7.9%	8.0%	8.0%	8.0%	8.0%	8.1%

Table 5. AUC loss for IMAX (our method) and the baselines on ImageNet

	Data set	$s = 1$	$s = 2$	$s = 3$	$s = 4$	$s = 5$	Best s
IMAX	Training	7.5%	5.8%	5.2%	4.9%	4.7%	4.9%
	Validation	9.1%	8.1%	7.9%	7.9%	7.8%	7.3%
	Test set	9.1%	8.2%	7.9%	7.9%	7.8%	7.7%
MAX-AVG	Training	7.5%	6.7%	6.4%	6.3%	6.1%	6.1%
	Validation	9.1%	8.7%	8.6%	8.6%	8.5%	7.9%
	Test set	9.1%	8.8%	8.7%	8.7%	8.5%	8.3%
AVG-AVG	Training	7.5%	7.3%	7.1%	7.1%	7.1%	7.1%
	Validation	9.1%	8.9%	8.9%	8.8%	8.7%	8.5%
	Test set	9.1%	8.9%	8.9%	8.8%	8.8%	8.7%

Table 6. Precision@10 results on Web Data. IMAX with $s = 1$ is equivalent to a LINEAR ranker.

	Data set	$s = 1$	$s = 2$	$s = 3$	$s = 4$	$s = 5$	Best s
IMAX	Training	82.5%	84.8%	85.5%	86.9%	86.5%	86.2%
	Validation	60.3%	63.1%	64.0%	64.7%	65.7%	73.8%
	Test	60.2%	63.1%	64.8%	65.1%	65.7%	64.5%
MAX-AVG	Training	82.5%	83.6%	83.4%	84.3%	83.6%	84.0%
	Validation	60.3%	61.0%	61.2%	61.0%	61.1%	70.4%
	Test	60.2%	61.7%	61.6%	61.6%	62.0%	62.6%
AVG-AVG	Training	82.5%	82.3%	82.2%	82.1%	82.7%	82.9%
	Validation	60.3%	59.7%	59.1%	59.1%	59.2%	65.5%
	Test	60.2%	58.8%	59.0%	58.5%	59.1%	58.9%

function, not shown). It is quite clear to see that each submodel has focused on a different sense or aspect for those queries. For the query "palm", the model learns phones, plants and beach images in the three different submodels. For the query "jaguar", the model learns animals, and two different car models / car model backgrounds. For "cell", it learns jail cells and two types of cell phones.

Table 7. Precision@10 results on ImageNet. IMAX with $s = 1$ is equivalent to a LINEAR ranker.

	Data set	$s = 1$	$s = 2$	$s = 3$	$s = 4$	$s = 5$	Best s
IMAX	Training	83.9%	86.2%	86.8%	87.1%	87.2%	87.2%
	Validation	65.7%	68.2%	69.1%	69.1%	69.4%	70.9%
	Test	65.6%	68.4%	69.2%	69.3%	69.4%	70.4%
MAX-AVG	Training	83.9%	84.8%	85.0%	85.6%	85.5%	85.9%
	Validation	65.7%	66.3%	66.7%	67.2%	67.0%	69.1%
	Test	65.6%	66.1%	66.9%	67.1%	67.1%	68.0%
AVG-AVG	Training	83.9%	84.3%	84.3%	84.3%	84.5%	84.6%
	Validation	65.7%	65.9%	66.1%	66.2%	66.3%	67.7%
	Test	65.6%	65.9%	66.0%	66.1%	66.2%	66.5%

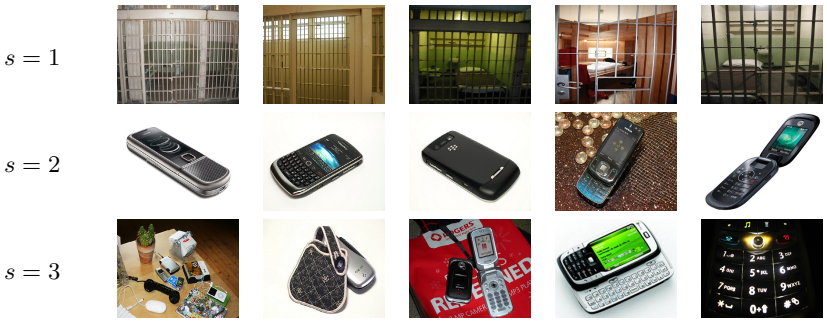

Fig. 2. The images returned by the IMAX ranking functions $f(q, s)$ for three discovered senses $s = 1, \ldots, 3$ of the query "cell" from ImageNet. One of the discovered senses focuses on jail cells, and the other two on different types of cell phone images.

Table 3 shows the nearest annotations for each discovered sense for "jaguar" and "palm" as described in Section 3.1. These results were obtained by training 100,000 rankers (for 100,000 random queries) and finding the closest queries for each sense (with the most similar learnt models). We believe the results show surprisingly good discovery of senses, considering the unsupervised nature of the task. For "jaguar", sense 3 is close to "jaguar animal", "bengal tiger" and many other *panthera*-related queries (more than 10). Sense 1, on the other hand, is close to "jaguar logo", "jaguar xf", "mazda" and "jaguar xk" – clearly a car model-related query. Similarly for "palm", sense 1 is related to "blackberry" and "smartphones", sense 2 is related to "'palm tree" and "coconut tree", and sense 3 focuses on "beautiful beaches", "tropical beach" and so on (i.e. images of the palm on a beach, rather than just the tree itself).

Table 8. Top 10 Best and Worst performing queries for IMAX on (a) ImageNet and (b) Web Data. We show the loss or gain in AUC of a LINEAR ranker compared to IMAX.

(a)

query	S	Gain	query	S	Gain
axe	4	+5.00	kim sun ah	4	-3.14
la barbie	4	+4.23	brontosaurus	3	-2.72
bass	5	+4.20	fotos para tumblr	2	-2.37
strange people	5	+3.97	tapety hd	5	-2.24
naga	3	+3.96	eduardo costa	5	-2.12
ape	5	+3.77	poni	4	-1.71
hart	2	+3.62	china	4	-1.65
broadway	4	+3.44	solo	3	-1.58
physiotherapy	3	+3.32	ibiza party	4	-1.56
fox	4	+3.36	vlada roslyakova	4	-1.53

(b)

query	S	Gain	query	S	Gain
stock	4	+8.89	hair	5	-2.80
rig	5	+8.44	return	3	-2.73
lap	2	+7.80	guard	2	-2.31
club	4	+7.36	stamp	2	-1.86
lock	5	+7.00	bond	5	-1.76
jack	4	+6.39	wash	3	-1.41
roller	5	+6.12	restraint	5	-1.41
capsule	4	+5.67	sweep	4	-1.29
head	3	+5.60	pull	4	-1.17
brass	5	+5.52	extension	3	-1.06

5 Conclusion

We have presented a novel method for determining the senses of word queries and the images that are relevant to each sense. We showed that our method improves ranking metrics compared to methods that do not model the senses. We obtain this improvement both on a set of random queries (on average), and particularly for queries that are known to be ambiguous. Simultaneously, our model is interpretable and we presented a method for discovering the senses that have been learnt. Finally, we should note that the method we presented is general and can be adapted to other tasks such as the "Google Goggles" type task of ranking results given an image. Our method could even be applicable outside of the vision domain such as in document retrieval.

References

1. Grangier, D., Bengio, S.: A discriminative kernel-based model to rank images from text queries. PAMI 30, 1371–1384 (2008)
2. Boser, B.E., Guyon, I., Vapnik, V.: A training algorithm for optimal margin classifiers. In: COLT, pp. 144–152 (1992)
3. Monay, F., Gatica-Perez, D.: On image auto-annotation with latent space models. In: ICMR, pp. 275–278 (2003)
4. Jeon, J., Lavrenko, V., Manmatha, R.: Automatic image annotation and retrieval using cross-media relevance models. In: SIGIR (2003)
5. Makadia, A., Pavlovic, V., Kumar, S.: A New Baseline for Image Annotation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 316–329. Springer, Heidelberg (2008)
6. Guillaumin, M., Mensink, T., Verbeek, J., Schmid, C.: Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In: ICCV, pp. 309–316 (2009)

7. Grangier, D., Bengio, S.: A Neural Network to Retrieve Images from Text Queries. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006, Part II. LNCS, vol. 4132, pp. 24–34. Springer, Heidelberg (2006)
8. Miller, G.A.: Wordnet: a lexical database for english. Commun. ACM 38, 39–41 (1995)
9. Pedersen, T., Bruce, R.: Distinguishing word senses in untagged text. In: EMNLP, vol. 2, pp. 197–207 (1997)
10. Purandare, A., Pedersen, T.: Word sense discrimination by clustering contexts in vector and similarity spaces. In: CoNLL, pp. 41–48 (2004)
11. Basile, P., Caputo, A., Semeraro, G.: Exploiting disambiguation and discrimination in information retrieval systems. In: WI/IAT Workshops, pp. 539–542 (2009)
12. Agirre, E., Edmonds, P.: Word Sense Disambiguation: Algorithms and Applications (Text, Speech and Language Technology), 1st edn. Springer (2007)
13. Navigli, R.: Word sense disambiguation: A survey. ACM Computing Surveys (CSUR) 41, 10 (2009)
14. Berg, T.L., Forsyth, D.A.: Animals on the web. In: CVPR (2006)
15. Schroff, F., Criminisi, A., Zisserman, A.: Harvesting Image Databases from the Web. PAMI 33, 754–766 (2011)
16. Loeff, N., Alm, C., Forsyth, D.: Discriminating image senses by clustering with multimodal features. In: ACL, pp. 547–554 (2006)
17. Saenko, K., Darrell, T.: Filtering abstract senses from image search results. In: NIPS, pp. 1589–1597 (2009)
18. Wan, K.W., Tan, A.H., Lim, J.H., Chia, L.T., Roy, S.: A latent model for visual disambiguation of keyword-based image search. In: BMVC (2009)
19. Chang, Y.-C., Chen, H.-H.: Image Sense Classification in Text-Based Image Retrieval. In: Lee, G.G., Song, D., Lin, C.-Y., Aizawa, A., Kuriyama, K., Yoshioka, M., Sakai, T. (eds.) AIRS 2009. LNCS, vol. 5839, pp. 124–135. Springer, Heidelberg (2009)
20. Barnard, K., Johnson, M.: Word sense disambiguation with pictures. Artif. Intell. 167, 13–30 (2005)
21. Yu, C.N.J., Joachims, T.: Learning structural svms with latent variables. In: ICML (2009)
22. Bergeron, C., Zaretzki, J., Breneman, C., Bennett, K.P.: Multiple instance ranking. In: ICML (2008)
23. Boyd, S., Mutapcic, A.: Subgradient methods. notes for ee364b, Stanford university (2007)
24. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR (2009)
25. Weston, J., Bengio, S., Usunier, N.: Wsabie: Scaling up to large vocabulary image annotation. In: IJCAI, pp. 2764–2770 (2011)
26. Grauman, K., Trevor, D.: The pyramid match kernel: Efficient learning with sets of features. JMLR 8, 725–760 (2007)
27. Leung, T., Malik, J.: Representing and Recognizing the Visual Appearance of Materials Using Three-Dimensional Textons. IJCV 43, 29–44 (2001)
28. Schoelkopf, B., Smola, A., Müller, K.R.: Kernel principal component analysis. In: Advances in Kernel Methods - Support Vector Learning, pp. 327–352. MIT Press (1999)
29. Barla, A., Odone, F., Verri, A.: Histogram intersection kernel for image classification. In: ICIP, pp. 513–516 (2003)
30. Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. JMLR 2, 265–292 (2001)
31. Zien, A., De Bona, F., Ong, C.S.: Training and approximation of a primal multiclass support vector machine. In: ASMDA (2007)