Can ambiguous words be helpful in image-understanding systems?

Huiling Zhou, Jiwei Hu and Kin Man Lam
Centre for Signal Processing, Department of Electronic and Information Engineering,
The Hong Kong Polytechnic University, Hong Kong
E-mail: {12900240r, 07901714r, enkmlam} @polyu.edu.hk

***Abstract*— A semantic gap always decreases the performance of the mapping for image-to-word, which is an important task in image understanding. Even efficient learning algorithms cannot solve this problem because: (1) of a lack of coincidence between the low-level features extracted from the visual data and the high-level information translated by human, and (2) an ambiguous word may lead to a wrong interpretation between low-level and high-level information. This paper introduces a discriminative model with a ranking function that optimizes the cost between the target word and the corresponding images, while simultaneously discovering the disambiguated senses of those words that are optimal for supervised tasks. Experiments were conducted using two datasets, and results show quite a promising result when compared with existing methods.**

I. INTRODUCTION

Nowadays, an increasing number of people are interested in taking photos and browsing pictures on the Internet using computers or mobile devices. It will be of great convenience if we can find a target image in a short time without knowledge of the tag of that image. In recent years, more and more Internet companies have begun researching content-based image retrieval (CBIR) instead of keyword-based image retrieval. However, it is always not easy for machines to automatically find the target images based on image content. While image retrieval has been an active topic [1, 2, 3] over the last two decades, a more challenging and interesting field is automatic concept recognition from the visual features of images. The challenge is primarily due to the well-known semantic-gap problem, which means that the extracted low-level features can hardly represent or reflect the high-level knowledge of humans. If we can successfully use some text keywords to reflect the content of every single image, image management and browsing will become much easier. Image annotation is the right approach to solve this problem, as its purpose is to assign relevant keywords to a given image so as to represent its content.

Recent annotation frameworks can be grouped based on three models: generative models, discriminative models, and nearest-neighbor-based models. Since discriminative models [4, 5] can generally yield a superior performance to generative models in some respects, we therefore emphasize discriminative models in this paper. Discriminative models aim to learn a separate classifier for each single label, and use the classifier to decide whether or not a query image belongs to that class. However, if a label (word) has ambiguous senses, the great variation in related training samples will degrade the annotation results.

Fig.1. Different images annotated by the same word *Apple*.

Although many queries might be considered to have a single sense only in terms of their dictionary definition, they can actually have several senses in terms of images. Figure 1 shows two different images which are annotated with the same word *Apple*. This means that the label *Apple* has ambiguous senses. Despite the tasks of word-sense disambiguation having been well studied in the field of natural language processing, the problem of polysemy with images is widely recognized as a difficulty for image-understanding systems. Recently, several methods have been proposed for word-sense disambiguation using text and images jointly. Loeff et al. [6] performed spectral clustering in both the text and the image domains and evaluated how well the clusters matched different senses. Saenko et al. [7] indicated that the method in [6] fails to give a sense label to images. Wan et al. [8] made use of the knowledge from the Internet to find the senses of a word. Then, LDA is used to explore latent visual topics and to learn a model of the wiki-senses in the latent space. Barnard et al. [9] proposed to model the joint probability of words and image regions, which requires prior segmentation. However, in this paper, we simply divide images into 5 sub-windows in our algorithm, rather than performing segmentation.

This paper proposes a method to learn a discriminative model using knowledge of the KNN-based distance measurement for image annotation. At the same time, we aim to discover the disambiguated senses of the word that are

optimal for supervised tasks. After that, we will learn a ranking function for each specific sense, and a combined ranking function for each single class (word).

The remainder of this paper is organized as follows. In Section II, we give an overview of our method. Then, we present our proposed method in detail in Section III. The experiment set-up and results, and a conclusion, are given in Sections IV and V, respectively.

II. Model Parameterization

Following our previous work [10], we divide the image instances into 5 patches (including 2×2 non-overlapping windows and the window at the image center, all of the same size), and then embed our patches of training samples into a hypergraph [11] and find the exemplars of each class by solving the graph. We then compute the feature difference between the training samples and the exemplars, and formalize the similarity vector. After that, we define a ranking function for every sense of every single class under the assumption that each single class has several senses.

In [10], we have described the details of how to construct the exemplars of each class. An exemplar represents the properties of the label class concerned, and we generate a new feature vector by computing the difference between the training samples and the exemplars. This new feature vector is called a delta-vector (Δ-vector). Assume that we have k exemplars for each class; the Δ-vectors obtained from the image I and the k exemplars form a Δ-matrix, i.e. $[|\Delta_1|, |\Delta_2|, \ldots |\Delta_k|]^T$. We choose 5 as the optimal number of exemplars for each class, which was determined by experiment in [10]. Thus, we set k equal to 5 in this paper. Specifically, an exemplar image is denoted as I_e^n, and a Δ-vector is obtained by comparing a training image to an exemplar as follows:

$$\Delta = \left| \Upsilon(I^m) - \Upsilon(I_e^n) \right| , \tag{1}$$

where I^m represents the m^{th} image in the dataset, and $\Upsilon(I)$ represents the feature set of image I. Now, the following formulation is used to obtain the output labels t from a query image I_q.

$$\begin{aligned} t^* &= \arg\max_t p(t \mid I_q) \\ &= \arg\max_t p(t \mid \Delta) \\ &= \arg\max_t \sum_i p(t \mid \Delta_i). \end{aligned} \tag{2}$$

Thus, the annotation task is performed in a new feature space composed of the difference vectors Δ_i instead of in the original image feature space.

Since we assume that each class label has several senses, we need to learn a ranking function for each sense. Thus, we have:

$$f_{t,s}(x,i) = w_{t,s} \cdot \Delta_{x,i} , \tag{3}$$

where x is an input image, $w_{t,s}$ are the parameters of the model, $\Delta_{x,i}$ is the feature difference between the image x and the i^{th} exemplar, and s is the s^{th} sense for class label t. The ranking function returns a real-valued output that measures the degree of the match between a label t and the image-feature difference $\Delta_{x,i}$, where a large value means a better match. Since we use Δ instead of image features, we can have the ranking function for an input image simply by using a maximal function as follows:

$$f_{t,s}(x) = \max_i f_{t,s}(x,i) . \tag{4}$$

When we have found the most suitable exemplar for the s^{th} sense of label c, we can then combine those scores to give an overall relevance match, independent of the senses:

$$f_t(x) = \max_{s \in S(t)} f_{t,s}(x) = \max_{s \in S(t)} (\max_i (w_{t,s} \cdot \Delta_{x,i})) . \tag{5}$$

Since we do not know how many senses $S(t)$ there are for each class, we can use the cross-validation method to find the optimal number for each class label t. Moreover, we do not know which sense should be related to an image. We can solve this problem by fixing the number of senses and setting a margin M between the score of positive samples and negative samples as follows:

$$\max_{s \in s(t)} f_{t,s}(x^+) > \max_{s \in s(t)} f_{t,s}(x^-) + M \tag{6}$$

Since M can be a small value, in this work we set M=0.5 empirically. After that, we have converted the problem of equation (2) into an optimization problem as follows:

$$\begin{aligned} &\min \sum_{t, x^+ \in \chi_t^+, x^- \in \chi_t^-} \xi(t, x^+, x^-) \\ &\text{subject to} \\ &\max_{s \in s(t)} f_{t,s}(x^+) > \max_{s \in s(t)} f_{t,s}(x^-) + M - \xi(t, x^+, x^-) \\ &\forall t, x^+ \in \chi_t^+, x^- \in \chi_t^-, \ \| w_{t,s} \| \le C \quad \forall t, s \ \xi(t, x^+, x^-) \ge 0. \end{aligned} \tag{7}$$

We introduce the slack variables ξ, which measure the ranking error in this model. We also add a constraint C to regularize the weighting parameter. Since different class labels have different numbers of senses, we can learn the parameters independently per label t. In the next section, we will introduce how to learn the weighting parameter $w_{t,s}$.

III. Learning Senses for Ambiguous Word

In Algorithm 1 (shown below), we describe the details of our learning algorithm. With the positive samples, our exemplar can make the feature difference small; with the negative samples, the same exemplar can make the feature difference large. This makes our training samples much purer than if using simple features only. When we learn the weighting

parameters, we use the classifier to output the score of each patch of a query image with respect to the exemplars of the corresponding class, and then add the scores of the 5 patches. Finally, we choose 5 and 10 classes with the highest scores, and select the labels of these classes as the final labels of the query image. Our model explores the different senses for ambiguous words, which helps to narrow the gap between the high-level information and the low-level features. We will give the experiment design and results in the next section.

Algorithm 1:

for each label t **do**

Input: training data with labels and corresponding exemplars for each label, $x^+ \in \chi_t^+, x^- \in \chi_t^-$, exemplars of label t : I_e^t.

Initialize the weights $w_{t,s}$, randomly with mean 0 and standard deviation $1/\sqrt{D}$ for all the labels t and s.

for each $s(t)$ ($s(t)$=1,…,5) **do**

Pick a positive sample x^+ randomly and all exemplars of the corresponding label t, find the exemplar i which makes: $\max_i f_{t,s}(x,i)$

Pick a positive example x^+ randomly and compute the Δ-vectors between x^+ and the exemplar i

Let $s^+ = \arg\max_{s \in S(t)}(w_{t,s} \cdot \Delta_{x^+,i})$.

Pick a negative example x^- randomly and compute the Δ-vectors between x^+ and the exemplar i

Let $s^- = \arg\max_{s \in S(t)}(w_{t,s} \cdot \Delta_{x^-,i})$.

if $f_{t,s}(\Delta_{x^+}) < f_{t,s}(\Delta_{x^-}) + 0.5$ **then**

Select a gradient step to minimize:

$\left|0.5 - f_{t,s}(\Delta_{x^+}) + f_{t,s}(\Delta_{x^-})\right|_+$, i.e:

Let $W_{t,s^+} \leftarrow W_{t,s^+} + \lambda \Delta_{x^+}$.

Let $W_{t,s^-} \leftarrow W_{t,s^-} - \lambda \Delta_{x^-}$.

Project the weights to enforce constraints:

$\| W_{t,s} \| \le C$:

for $s' \in \{s^+, s^-\}$ **do**

if $\| W_{t,s'} \| > C$ **then**

Let $W_{t,s'} \leftarrow C W_{t,s'} / \| W_{t,s'} \|$.

end if

end for

end if

until validation error does not decrease.

end for

Output final model with best validation error

end for

IV. Experiments and Results

Two benchmark image databases are used in our experiments. The first database used is Corel 5K, which contains 5,000 images comprising 4,500 training samples and 500 testing samples. Each image in the dataset is annotated with about 3.5 keywords on average, and the dictionary has a total of 374 words or labels. The other dataset used is Corel 30K, which is similar to Corel 5K except that it is substantially larger, containing 31,695 images and 5,587 words or labels.

Similar to [12], different feature descriptors, and a combination of these features, were used in our experiments. The following features are used for each image patch. We compute the Δ-vectors as the differences between the input patches and each of the exemplars.

(1) Color feature: RGB color moment (3×3 grid, color mean, variance, skewness for R, G, B),
(2) Edge histogram (edge-orientation histogram, Canny edge detector),
(3) Gabor wavelets transform (5 scales and 8 orientations, 3 moments for each sub-image),
(4) Local binary pattern (a 59-d LBP histogram), and
(5) GIST (a complex and popular feature descriptor).

To make a fair comparison with other state-of-the-art methods, we choose precision and recall as our evaluation criteria. The precision rate and recall rate for each test image are measured by comparing the annotated results to the ground-truth, and then the average precision and recall of all the test images are computed to form the final results. In addition to the mean precision rates (*P%*) and the mean recall rates (*R%*), the number of total keywords recalled (N^+) is also used as a performance index.

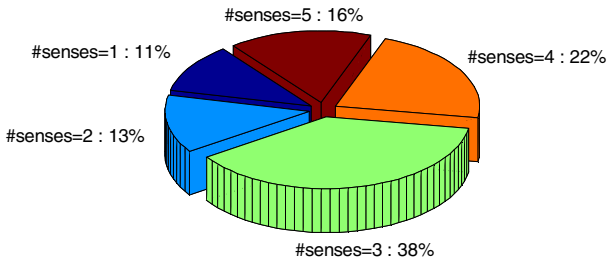

Fig.2. Percentage of classes achieving the best performance with a different number of senses (k=1,2,3,4,5).

Since the number of senses is unknown for each class, we have to learn the optimal number for the different classes. Figure 2 shows the percentage of classes choosing the optimal number (1, 2, 3, 4, or 5). This also means that when an optimal number is chosen, the discriminative classifier of the corresponding class achieves the best performance.

With the help of the ambiguous words, our new model discovers different senses for each class label. Figure 3 shows

the performance with and without considering the senses of the words, as a trade-off between precision and recall for this dataset.

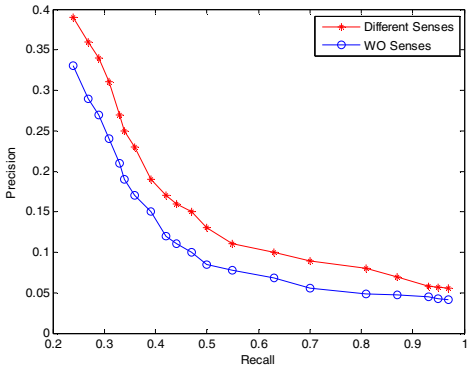

Fig. 3. Precision-recall plots generated by varying the number of keywords assigned to images with different numbers of senses and without considering senses.

Table 1 shows the performance of our proposed method versus some state-of-the-art methods. Having discovered different senses for ambiguous words, we can find an obvious improvement in all the three measurements compared with our previous work and with other state-of-the-art methods.

Table 1. Performances based on the Corel5K dataset for some existing methods and our proposed method.

Methods	*P%*	*R%*	N^+
MBRM[13]	24	25	122
SML [1]	23	29	137
TGLM [14]	25	29	131
JEC [15]	27	32	139
LASSO[15]	24	29	127
TagProp[5]	33	42	160
Our previous work [10]	32	38	151
Proposed	35	43	164

A larger dataset is used to evaluate the performance of our framework. We follow the same procedure as in [16]: we choose only those words that are used as labels for more than 10 images in the Corel 30K dataset to form the semantic vocabulary. The average number of labels per image is about 3.6. To evaluate our method for large-scale annotation tasks, we compare our algorithm's performance with HPM [16], SML [1], and our previous work. The results are tabulated in Table 2. We can see that our proposed method outperforms most of the other methods. Although HPM (Given 1) is slightly better than ours, our performance is promising without the help of any prior knowledge (i.e. labels already known).

Table 2. Performance comparison on the Corel 30K dataset.

Method	SML	HPM, Given 0	HPM, Given 1	LFA [17]	Prop. Work
P%	12	10	16	13	15
R%	21	19	31	24	32
$Rate^+$	44.63%	46.21%	55.71%	49.89%	56.01%

V. CONCLUSIONS

This paper presents an idea for learning a discriminative classifier for each single label, while simultaneously exploring different senses for it. We have answered the question proposed in this paper: ambiguous words can be helpful in our image-understanding systems. Our discriminative model, with a ranking function, optimizes the problem and then solves it. Experimental results show that our method can achieve a promising performance, even when the dataset is large.

REFERENCES

[1] G. Carneiro, A.B. Chan, P.J. Moreno, N. Vasconcelos. Supervised Learning of Semantic Classes For Image Annotation and Retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (3) (2007) 394-410.

[2] G. Qiu. Indexing chromatic and achromatic patterns for content-based colour image retrieval, Pattern Recognition 35 (2002) 1675 – 1686.

[3] Yuchi Huang, Qingshan Liu, Shaoting Zhang, Metaxas, D. Netc. Image Retrieval via Probabilistic Hypergraph Ranking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (2010) 3376-3383.

[4] David Grangier and Samy Bengio, "A Discriminative Kernel-Based Model to Rank Images from Text Queries," *IEEE TPAMI*, vol.30, no.8, pp. 1371-1384, Aug. 2008.

[5] M.Grubinger, T.Mensink, J.Verbeek, and C.Schmid "Tagprop: Discriminative Metric Learning In Nearest Neighbor Models for Image Auto-Annotations," ICCV 2009, pp. 309-314.

[6] Loeff, N., Alm, C., Forsyth, D.: Discriminating image senses by clustering with multimodal features. In: ACL.2006, pp.547-554.

[7] Saenko, K., Darrell, T.: Filtering abstract senses from image search results. In:NIPS. (2009), pp. 1589-1597

[8] 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)

[9] Barnard, K., Johnson, M.: Word sense disambiguation with pictures. Artif. Intell.167 (2005) pp. 13-30.

[10] J. Hu, C. Sun, K.M. Lam, Learning a Discriminative Model for Image Annotation, in: Proceedings of Asia Pacific Information and Signal Processing Association, ASC 2011.

[11] Yuchi Huang, Qingshan Liu and etc, "Image Retrieval via Probabilistic Hypergraph Ranking," CVPR 2010, pp.3376-3383.

[12] Jianke Zhu, Steven C.H.Hoi, Michael R. Lyu and Shuicheng Yan, "Near-Duplicate Keyframe Retrieval by Nonrigid Image Matching," ACM Multimedia 2008.

[13] S.L. Feng, R. Manmatha, and V. Lavrenko, "Multiple Bernoulli Relevance Models for Image and Video Annotation." CVPR 2004, vol.2, pp.1002-1009.

[14] J. Liu, M. Li, Q. Liu, et al., "Image Annotation via Graph Learning," *Pattern Recogn*, vol.42, no2, pp.218-228, 2009

[15] Ameesh Makadia, Vladimir Pavlovic, Sanjiv Kumar. A New Baseline for Image Annotation, in: Proceedings of the European Conference on Computer Vision, (2008) 316-329.

[16] N. Zhou, W. Cheung, G. Qiu, X. Xue. A Hybrid Probabilistic Model for Unified Collaborative and Content based Image Tagging, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (7) (2011) 1281-1294.

[17] J.Hu, K.M.Lam, An Efficient Two-stage Framework for Image Annotation. Pattern Recognition 46(3): (2013) 936-947.