Tagging over Time: Real-world Image Annotation by Lightweight Meta-learning

Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang
The Pennsylvania State University, University Park, PA 16802, USA
{datta, djoshi, jiali, jwang}@psu.edu

ABSTRACT

Automatic image annotation has been a hot-pursuit among multimedia researchers of late. Modest performance guarantees and limited adaptability often restrict its applicability to real-world settings. We propose *tagging over time* (T/T) to push the technology toward real-world applicability. Of particular interest are online systems that receive user-provided images and feedback over time, with user focus possibly changing and evolving. The T/T framework consists of a principled probabilistic approach to meta-learning, which acts as a go-between for a 'black-box' annotation system and the users. Inspired by *inductive transfer*, the approach attempts to harness available information, including the black-box model's performance, the image representations, and the WordNet ontology. Being computationally 'lightweight', this meta-learner efficiently re-trains over time, to improve and/or adapt to changes. The black-box annotation model is not required to be re-trained, allowing computationally intensive algorithms to be used. We experiment with standard image datasets and real-world data streams, using two existing annotation systems as black-boxes. Both batch and online annotation settings are experimented with. It is observed that the addition of this meta-learning layer produces much improved results that outperform best-known results. For the online setting, the T/T approach produces progressively better annotation with time, significantly outperforming the black-box as well as the static form of the meta-learner, on real-world data.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Linguistic processing

General Terms

Algorithms, Experimentation, Performance.

1. INTRODUCTION

The scale of the Web makes it essential to have automated systems for content management. A significant fraction of this content exists in the form of images, often with metadata unusable for meaningful search and organization. To this end, automatic image annotation or tagging is an important step toward achieving large-scale organization and retrieval. In the recent years, many new image annotation ideas have been proposed [2, 3, 6, 9, 12, 13, 16, 17, 18, 20, 24, 26]. Typical scenarios considered are those where batches of images, having visual semblance with training images, are statically tagged. However, incorporating automatic image tagging into real-world photo-sharing environments (e.g., Flickr, Riya, Photo.Net) poses unique challenges that have seldom been taken up in the past. In an online setting, where people upload images, automatic tagging needs to be performed as and when they are received, to make them searchable by text. On the other hand, people often collaboratively tag a subset of the images from time to time, which can be leveraged for automatic annotation. Moreover, time can lead to changes in user-focus/user-base, resulting in continued evolution of user tag vocabulary, tag distributions, or topical distribution of uploaded images.

In online systems, e.g., Yahoo! Flickr [10], collaborative image tagging, also referred to as *folksonomic tagging*, plays a key role in making the image collections organizable by semantics and searchable by text [23]. This effort can go a long way if automated image annotation engines complement the human tagging process, taking advantage of these tags and addressing the inherent scalability issues associated with human-driven processes. Traditionally, annotation engines have considered the batch setting, whereby a fixed-size dataset is used for training, following which it is applied to a set of test images, in the hope of generalization. A realistic embedding of such an engine into an online setting must tackle three main issues: (1) Current state-of-the-art in annotation is a long way off from being reliable on real-world data. (2) Image collections in online systems are dynamic in nature - over time, new images are uploaded, old ones are tagged, etc. Annotation engines have traditionally been trained on fixed image collections tagged using fixed vocabularies, which severely constrain adaptability over time. (3) While a solution may be to re-train the annotation engine with newly acquired images, most proposed methods are too computationally intensive to re-train frequently. None of the questions associated with image annotation in an online setting, such as (a) how often to re-train, (b) with what performance gain, and (c) at what cost, have been answered in the annotation literature. A recently proposed system, Alipr [1], incorporates automatic tagging into its photo sharing framework, but it still is limited by the above issues.

MM'07, September 23–28, 2007, Augsburg, Bavaria, Germany.

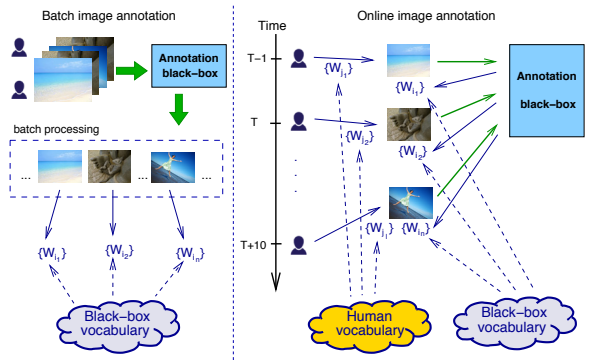

Figure 1: Batch and online image annotation settings.

From a machine-learning point of view, the main difference is in the nature by which ground-truth is made available (refer to Fig. 1). The batch setting (left) is what has traditionally been conceived in the annotation literature, whereby the entire ground-truth is available at once, with no intermittent user-feedback. The online setting (right) is an abstracted representation of how an automated annotation system can be incorporated into a public-domain photo-sharing environment. As discussed, this setting poses challenges which have largely not been previously dealt with.

In this paper, we make two major contributions. First, we propose a principled, lightweight, meta-learning framework for image tagging with very few simplifying assumptions, that can be built atop *any* available annotation engine that we refer to as the 'black-box'. Experimentally, we find that such an approach can dramatically improve annotation performance over the black-box system in a batch setting (and thus make it more viable for real-world implementation), incurring insignificant computational overhead for training and annotation. Second, we explore the online setting, whereby images and user tags enter the system as a temporal sequence, as in the case of Flickr and Alipr. Here, we propose the *tagging over time* (T/T) approach that incrementally trains this meta-learner over time to progressively improve annotation performance and adapt to changing user-system dynamics, without the need to retrain the (computationally intensive) annotation engine. A list of contributions is as follows:

- A meta-learning framework for annotation, based on *inductive transfer*, is proposed, and shown to dramatically boost performance in batch and online settings.
- The meta-learning framework is designed in a way that makes it lightweight for (re-)training and inferencing in an online setting, by making the training process deterministic in time and space consumption.
- Appropriate *smoothing* steps are introduced to deal with sparsity in the meta-learner training data.
- Two different re-training models, *persistent memory* and *transient memory*, are proposed. They are realized through simple incremental/decremental learning steps, and the intuitions behind them are experimentally validated.

Experiments are conducted by building the meta-learner atop two annotation engines, using the popular Corel dataset, and two real-world image traces and user-feedback obtained from the Alipr system. Empirically, various intuitions about the meta-learner and the T/T framework are tested.

1.1 Related Work

Research in automatic image annotation can be roughly categorized into two different 'schools of thought': **(1)** Words and visual features are jointly modeled to yield compound predictors describing an image or its constituent regions. The words and image representations used could be disparate [2, 9, 13] or single vectored representations of text and visual features [20, 18]. **(2)** Automatic annotation is treated as a two step process consisting of supervised image categorization, followed by word selection based on the categorization results [6, 16, 3]. While the former approaches can potentially label individual image regions, ideal region annotation would require precise image segmentation, an open problem in computer vision. Although the the latter techniques cannot label regions, they are typically more scalable to large image collections.

The term meta-learning has historically been used to describe the learning of meta-knowledge about learned knowledge. Research in meta-learning covers a wide spectrum of approaches and applications, as has been reviewed in [22]. Here, we briefly discuss the approaches most pertinent to this work. One of the most popular meta-learning approaches, *boosting* is widely used in supervised classification [11]. Boosting involves iteratively adjusting weights assigned to data points during training, to adaptively reduce misclassification. In *stacked generalization*, weighted combinations of responses from multiple learners are taken to improve overall performance [25]. The goal here is to learn optimal weights using validation data, in the hope of generalization to unseen data. A research area under the meta-learning umbrella that is closest to our work is *inductive transfer/transfer learning*. Research in inductive transfer is grounded on the belief that knowledge assimilated about certain tasks can potentially facilitate the learning of certain other tasks [4]. A recent workshop [21] at NIPS 2005 was devoted to discussing advances and applications of inductive transfer. Incrementally learning support vectors as and when training data is encountered has been explored as a scalable supervised learning procedure [5]. In our work, we draw inspiration from inductive transfer and incremental/decremental learning to develop the meta-learner and the overall T/T framework.

2. META-LEARNING

Given an image annotation system or algorithm, we treat it as a 'black-box' and build a lightweight meta-learner that attempts to understand the performance of the system on each word in its vocabulary, taking into consideration all available information, which includes:

- Annotation performance of the black-box models.
- Ground-truth annotation/tags, whenever available.
- External knowledge bases, e.g., WordNet.
- Visual content of the images.

Here, we discuss the nature of each one, and formulate a probabilistic framework to harness all of them. We consider a black-box system that takes an image as input and guesses one or more words as its annotation, which can be any of [2, 3, 6, 9, 12, 13, 16, 17, 18, 20, 24, 26]. We do not concern ourselves directly with the methodology or philosophy the black-box employs, but care about their output. A ranked ordering of the guesses is not necessary for our framework, but can be useful for empirical comparison.

Assume that either there is ground-truth readily available for a subset of the images, or, in an online setting, images are being uploaded and collaboratively/individually tagged from time to time, which means that ground-truth is made

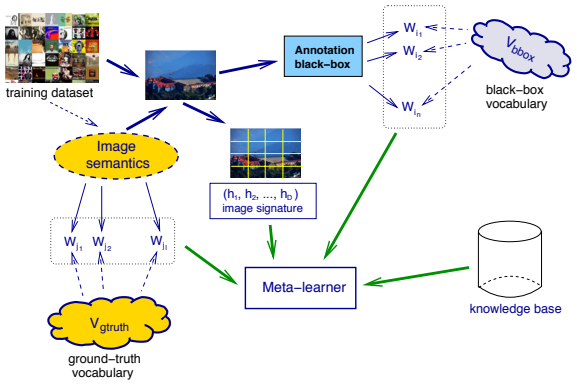

Figure 2: Meta-learner training framework for annotation.

available as and when users tag them. For example, consider that an image is uploaded but not tagged. At this time, the black-box can make guesses at its annotation. At a later time, user provide tags to it, at which point it becomes clear how good the black-box's guesses were. This is where the meta-learner fits in, in an online scenario. The images are also available to the meta-learner for visual content analysis. Furthermore, knowledge bases (e.g., WordNet [19]) can be potentially useful, since semantics recognition is the desiderata of annotation.

2.1 Generic Framework

Let the black-box annotation system be known to have a word vocabulary denoted by V_{bbox}. Let us denote the ground-truth vocabulary by V_{gtruth}. The meta-learner works on the union of these vocabularies, namely $V = (V_{bbox} \cup V_{gtruth}) = \{w_1, \ldots, w_K\}$, where $K = |V|$, the size of this overall vocabulary. Given an image I, the black-box predicts a set of words to be its correct annotation. To denote these *guesses*, we introduce indicator variables $G_w \in \{0, 1\}$, $w \in V$, where a value of 1 or 0 indicates whether word w_i is predicted by the black-box for I or not. We introduce similar indicator variables $A_w \in \{0, 1\}$, $w \in V$ to denote the ground-truth tagging, where a value of 1 or 0 indicates whether w is a correct annotation for I or not. Strictly speaking, we can conceive the black-box as a multi-valued function f_{bbox} mapping an image I to indicator variables G_{w_i}: $f_{bbox}(I) = (G_{w_1}, \ldots, G_{w_K})$. Similarly, the ground-truth labels can be thought of as a function f_{gtruth} mapping the image to its true labels using the indicator variables: $f_{gtruth}(I) = (A_{w_1}, \ldots, A_{w_K})$.

Regardless of the abstraction of visual content that the black-box uses for annotation, the pixel-level image representation may be still available to the meta-learner. If some visual features extracted from the images represent a different abstraction than what the black-box uses, they can be thought of as a *different viewpoint* and thus be potentially useful for semantics recognition. Such a visual feature representation, that is also simple enough not to add significant computational overhead, can be thought of as a function defined as: $f_{vis}(I) = (h_1, \ldots, h_D)$. Here, we specify a D-dimensional image feature vector representation as an example. Instead, other non-vector representations (e.g., variable-length region-based features) can also be used as long as they are efficient to compute and process, so as to keep the meta-learner lightweight.

Finally, the meta-learner also has at its disposal an external knowledge base, namely the semantic lexicon WordNet, which is essentially a semantic lexicon for the English language that has in the past been found useful for image annotation [14, 8]. In particular, WordNet-based *semantic relatedness measures* (e.g., [15]) have benefited annotation tasks. WordNet, however, does not include most proper nouns and colloquial words that are often prevalent in human tag vocabularies. Such non-WordNet words must therefore be ignored or eliminated from the vocabulary V in order to use WordNet on the entire vocabulary. The meta-learner attempts to assimilate useful knowledge from this lexicon for performance gain. It can be argued that this semantic knowledge base may help discover the connection between the true semantics of images, the guesses made by the black-box model for that image, and the semantic relatedness among the guesses. Once again, the inductive transfer idea comes into play, whereby we conjecture that the black-box, with its ability to recognize semantics of some image classes, may help recognize the semantics of entirely different classes of images. Let us denote the *side-information* extracted (externally) from the knowledge base and the black-box guesses for an image by a numerical abstraction, namely $f_{kbase}(I) = (\rho_1, \ldots, \rho_K)$, where $\rho_i \in \mathbb{R}$, with the knowledge base and the black-box guesses implicitly conditioned.

We are now ready to postulate a probabilistic formulation for the meta-learner. In essence, this meta-learner, trained on available data with feedback (see Fig. 2), acts a function which takes in all available information pertaining to an image I, including the black-box's annotation, and produces a new set of guesses as its annotation. In our meta-learner, this function is realized by taking decisions on each word independently. In order to do so, we compute the following *odds* in favor of each word w_j to be an actual ground-truth tag, given all pertinent information, as follows:

$$\ell_{w_j}(I) = \frac{Pr\Big(A_{w_j} = 1 \mid f_{bbox}(I), f_{kbase}(I), f_{vis}(I)\Big)}{Pr\Big(A_{w_j} = 0 \mid f_{bbox}(I), f_{kbase}(I), f_{vis}(I)\Big)} \quad (1)$$

Note that here $f_{bbox}(I)$ (and similarly, the other terms) denotes a realization of the corresponding random variables given the image I. Using Bayes' Rule, we can re-write:

$$\begin{aligned} \ell_{w_j}(I) &= \frac{Pr(A_{w_j}{=}1, f_{bbox}(I), f_{kbase}(I), f_{vis}(I))}{Pr(f_{bbox}(I), f_{kbase}(I), f_{vis}(I))} \quad (2) \\ &\times \frac{Pr(f_{bbox}(I), f_{kbase}(I), f_{vis}(I))}{Pr(A_{w_j} = 0, f_{bbox}(I), f_{kbase}(I), f_{vis}(I))} \\ &= \frac{Pr(A_{w_j} = 1, f_{bbox}(I), f_{kbase}(I), f_{vis}(I))}{Pr(A_{w_j} = 0, f_{bbox}(I), f_{kbase}(I), f_{vis}(I))} \end{aligned}$$

In $f_{bbox}(I)$, if the realization of variable G_{w_i} for each word w_i is denoted by $g_i \in \{0, 1\}$ given I, then without loss of generality, for each j, we can split $f_{bbox}(I)$ as follows:

$$f_{bbox}(I) = \Big(G_{w_j} = g_j, \bigcup_{i \neq j} (G_{w_i} = g_i)\Big) \quad (3)$$

We now evaluate the joint probability in the numerator and denominator of ℓ_{w_j} separately, using Eq. 3. For a realization $a_j \in \{0, 1\}$ of the random variable A_{w_j}, we can factor the joint probability (using the chain rule of probability) into a

prior and a series of conditional probabilities, as follows:

$$
\begin{aligned}
&Pr\Big(A_{w_j}{=}a_j, f_{bbox}(I), f_{kbase}(I), f_{vis}(I)\Big) \qquad (4)\\
&= Pr\Big(G_{w_j}{=}g_j\Big) \times Pr\Big(A_{w_j} = a_j \mid G_{w_j}{=}g_j\Big)\\
&\times Pr\Big(\bigcup_{i\neq j}(G_{w_i}{=}g_i) \mid A_{w_j} = a_j, G_{w_j}{=}g_j\Big)\\
&\times Pr\Big(f_{kbase}(I) \mid \bigcup_{i\neq j}(G_{w_i}{=}g_i), A_{w_j} = a_j, G_{w_j}{=}g_j\Big)\\
&\times Pr\Big(f_{vis}(I) \mid f_{kbase}(I), \bigcup_{i\neq j}(G_{w_i}{=}g_i), A_{w_j}{=}a_j, G_{w_j}{=}g_j\Big)
\end{aligned}
$$

The odds in Eq. 1 can now be factored using Eq. 2 and 4:

$$
\begin{aligned}
\ell_{w_j}(I) &= \frac{Pr(A_{w_j} = 1 \mid G_{w_j}{=}g_j)}{Pr(A_{w_j} = 0 \mid G_{w_j}{=}g_j)} \qquad (5)\\
&\times \frac{Pr(\bigcup_{i\neq j}(G_{w_i}{=}g_i) \mid A_{w_j} = 1, G_{w_j}{=}g_j)}{Pr(\bigcup_{i\neq j}(G_{w_i}{=}g_i) \mid A_{w_j} = 0, G_{w_j}{=}g_j)}\\
&\times \frac{Pr(f_{kbase}(I) \mid A_{w_j} = 1, \bigcup_{i\neq j}(G_{w_i}{=}g_i), G_{w_j}{=}g_j)}{Pr(f_{kbase}(I) \mid A_{w_j} = 0, \bigcup_{i\neq j}(G_{w_i}{=}g_i), G_{w_j}{=}g_j)}\\
&\times \frac{Pr(f_{vis}(I)|A_{w_j}{=}1, f_{kbase}(I), \bigcup_{i\neq j}(G_{w_i}{=}g_i), G_{w_j}{=}g_j)}{Pr(f_{vis}(I)|A_{w_j}{=}0, f_{kbase}(I), \bigcup_{i\neq j}(G_{w_i}{=}g_i), G_{w_j}{=}g_j)}
\end{aligned}
$$

Note that the ratio of priors $\frac{Pr(G_{w_j}=g_j)}{Pr(G_{w_j}=g_j)} = 1$, and hence is eliminated. The ratio $\frac{Pr(A_{w_j}=1|G_{w_j}=g_j)}{Pr(A_{w_j}=0|G_{w_j}=g_j)}$ is a *sanity check* on the black-box for each word. For $G_{w_j} = 1$, it can be paraphrased as "Given that word w_j is guessed by the black-box for I, what are the odds of it being correct?". Naturally, a higher odds indicates that the black-box has greater *precision* in guesses (i.e., when w_j is guessed, it is usually correct). A similar paraphrasing can be done for $G_{w_j} = 0$, where higher odds implies lower word-specific *recall* in the black-box guesses. A good annotation system should be able to achieve independently (word-specific) and collectively (overall) good precision and recall. These probability ratios therefore give the meta-learner indications about the black-box model's performance for each word in the vocabulary.

When $g_j = 1$, the ratio $\frac{Pr(\bigcup_{i\neq j}(G_{w_i}=g_i)|A_{w_j}=1,G_{w_j}=g_j)}{Pr(\bigcup_{i\neq j}(G_{w_i}=g_i)|A_{w_j}=0,G_{w_j}=g_j)}$ in Eq. 5 relates each correctly/wrongly guessed word w_j to how every other word $w_i, i \neq j$ is guessed by the black-box. This component has strong ties with the concept of co-occurrence popular in the language modeling community, the difference being that here it models the word co-occurrence of the black-box's outputs with respect to ground-truth. Similarly, for $g_j = 0$, it models how certain words do not co-occur in the black-box's guesses, given the ground-truth. Since the meta-learner makes decisions about each word independently, it is intuitive to separate them out in this ratio as well. That is, the question of whether word w_i is guessed or not, given that another word w_j is correctly/wrongly guessed, is treated independently. Furthermore, efficiency and robustness become major issues in modeling joint probability over a large number of random variables, given limited data. Considering these factors, we assume the guessing of each word w_i conditionally independent of each other, given a correctly/wrongly guessed word w_j, leading to the following approximation:

$$
\begin{aligned}
&Pr\Big(\bigcup_{i\neq j}(G_{w_i}{=}g_i) \mid A_{w_j} = a_j, G_{w_j}{=}g_j\Big) \qquad (6)\\
&\approx \prod_{i\neq j} Pr(G_{w_i}{=}g_i \mid A_{w_j} = a_j, G_{w_j} = g_j)
\end{aligned}
$$

The ratio can then be written as

$$
\begin{aligned}
&\frac{Pr(\bigcup_{i\neq j}(G_{w_i}{=}g_i) \mid A_{w_j} = 1, G_{w_j}{=}g_j)}{Pr(\bigcup_{i\neq j}(G_{w_i}{=}g_i) \mid A_{w_j} = 0, G_{w_j}{=}g_j)} \qquad (7)\\
&= \prod_{i\neq j} \frac{Pr(G_{w_i}{=}g_i \mid A_{w_j} = 1, G_{w_j} = g_j)}{Pr(G_{w_i}{=}g_i \mid A_{w_j} = 0, G_{w_j} = g_j)}
\end{aligned}
$$

The problem of conditional multi-word co-occurrence modeling has been effectively transformed into that of pairwise co-occurrences, which is attractive in terms of modeling, representation, and efficiency. While co-occurrence really happens when $g_i = g_j = 1$, the other combinations of values can also be useful, e.g., how the frequency of certain word pairs not being both guessed differs according to the correctness of these guesses. The usefulness of component ratios of this product to meta-learning, namely $\frac{Pr(G_{w_i}=g_i|A_{w_j}=1,G_{w_j}=g_j)}{Pr(G_{w_i}=g_i|A_{w_j}=0,G_{w_j}=g_j)}$, can again be justified based on ideas of inductive transfer. The following examples illustrate this:

- Some visually coherent objects do not often co-occur in the same natural scene. If the black-box strongly associates orange color with the setting sun, it may often be making the mistake of labeling orange (fruit) as the sun, or vice-versa, but both occurring in the same scene may be unlikely. In this case, with $w_i =$ 'oranges' and $w_j =$ 'sun' (or vice-versa), w_i and w_j will frequently co-occur in the black-box's guesses, but in most such instances, one guess will be wrong. This will imply low values of the above ratio for this word pair, which in turn models the fact that the black-box *mistakenly confuses* one word for another, for visual coherence or otherwise.
- Some objects that are visually coherent also frequently co-occur in natural scenes. For example, in images depicting beaches, 'water', and 'sky' often co-occur as correct tags. Since both are blue, the black-box may mistake one for the other. However, such mistakes are acceptable if both are actually correct tags for the image. In such cases, the above ratio is likely to have high values for this word pair, modeling the fact that evidence about one word *reinforces belief* in another, for visual coherence coupled with co-occurrence (See Fig. 3 (A)).
- For some word w_j, the black-box may not have effectively learned anything. This may happen due to lack of good training images, inability to capture apt visual properties, or simply the absence of the word in V_{bbox}. For example, users may be providing the word $w_j =$'feline' as ground-truth for images containing $w_i =$'cat', while only the latter may be in the black-box's vocabulary. In this case, $G_{w_j} = 0$, and the ratio $\frac{Pr(G_{w_i}=g_i|A_{w_j}=1,G_{w_j}=0)}{Pr(G_{w_i}=g_i|A_{w_j}=0,G_{w_j}=0)}$ will be high. This is a direct case of inductive transfer, where the training on one word *induces guesses* at another word in the vocabulary (See Fig. 3 (C)).

Other such scenarios where this ratio provides useful information can be conceived (See Fig. 3 (B), (D)). For the term $\frac{Pr(f_{kbase}(I)|A_{w_j}=1,\bigcup_{i\neq j}(G_{w_i}=g_i),G_{w_j}=g_j)}{Pr(f_{kbase}(I)|A_{w_j}=0,\bigcup_{i\neq j}(G_{w_i}=g_i),G_{w_j}=g_j}$ in Eq. 5, since we

deal with each word separately, the numerical abstractions $f_{kbase}(I)$ relating WordNet to the model's guesses/ground-truth can be separated out for each word (conditionally independent of other words). Therefore, we can write

$$\frac{Pr(f_{kbase}(I) \mid -)}{Pr(f_{kbase}(I) \mid -)} \approx \frac{Pr(\rho_j \mid A_{w_j}=1, -)}{Pr(\rho_j \mid A_{w_j}=0, -)} \quad (8)$$

Finally, $\frac{Pr(f_{vis}(I)|A_{w_j}=1, f_{kbase}(I), \bigcup_{i\neq j}(G_{w_i}=g_i), G_{w_j}=g_j)}{Pr(f_{vis}(I)|A_{w_j}=0, f_{kbase}(I), \bigcup_{i\neq j}(G_{w_i}=g_i), G_{w_j}=g_j)}$ in Eq. 5 can be simplified, since $f_{vis}(I)$ is the meta-learner's own visual representation $f_{vis}(I)$, unrelated to the black-box's visual abstraction used for making guesses, and hence also the semantic relationship $f_{kbase}(I)$. Therefore, we re-write

$$\frac{Pr(f_{vis}(I)|A_{w_j}=1, -)}{Pr(f_{vis}(I)|A_{w_j}=0, -)} \approx \frac{Pr((h_1, \ldots, h_D)|A_{w_j}=1)}{Pr((h_1, \ldots, h_D)|A_{w_j}=0)} \quad (9)$$

which is essentially the ratio of conditional probabilities of the visual features extracted by the meta-learner, given w_j is correct/incorrect. A strong support for the independence assumptions made in this formulation comes from the superior experimental results. Putting everything together, and taking logarithm (monotonically increasing) to get around issues of machine precision, we can re-write Eq. 5 as a *logit*:

$$\begin{aligned}\log \ell_{w_j}(I) = &\log\left(\frac{Pr(A_{w_j}=1 \mid G_{w_j}=g_j)}{1-Pr(A_{w_j}=1 \mid G_{w_j}=g_j)}\right) \quad (10)\\ &+ \sum_{i\neq j} \log\left(\frac{Pr(G_{w_i}=g_i \mid A_{w_j}=1, G_{w_j}=g_j)}{Pr(G_{w_i}=g_i \mid A_{w_j}=0, G_{w_j}=g_j)}\right)\\ &+ \log\left(\frac{Pr(\rho_j \mid A_{w_j}=1, \bigcup_{i\neq j}(G_{w_i}=g_i), G_{w_j}=g_j)}{Pr(\rho_j \mid A_{w_j}=0, \bigcup_{i\neq j}(G_{w_i}=g_i), G_{w_j}=g_j)}\right)\\ &+ \log\left(\frac{Pr(h_1, \ldots, h_D \mid A_{w_j}=1)}{Pr(h_1, \ldots, h_D \mid A_{w_j}=0)}\right)\end{aligned}$$

This logit is used by our meta-learner for annotation, where a higher value for a word indicates a higher odds in its support, given all pertinent information. What words to eventually use as annotation for an image I can then be decided in at least two different ways, as found in the literature:

- **Top r:** After ordering all words $w_j \in V$ in the increasing magnitude of $\log \ell_{w_j}(I)$ to obtain a rank ordering, we annotate I using the top r ranked words.
- **Threshold r%**: We can annotate I by thresholding at the top r percentile of the range of $\log \ell_{w_i}(I)$ values for the given image over all the words.

The formulation at this point is fairly generic, particularly with respect to harnessing of WordNet ($f_{kbase}(I)$) and the visual representation ($f_{vis}(I)$). We now go into specifics of a particular form of these functions that we use in experiments. Furthermore we consider robustness issues that the meta-learner runs into, which we discuss in the next section.

2.2 Estimation and Smoothing

The crux of the meta-learner is Eq. 10, which takes in an image I and the black-box guesses for it, and subsequently computes odds for each word. The probabilities involving A_{w_j} must all be estimated from any training data that may be available to the meta-learner. In a temporal setting, there will be *seed* training data to start with, and the estimates will be refined as and when more data/feedback becomes available. Let us consider the estimation of each term separately, given a *training set* of size L, consisting of images

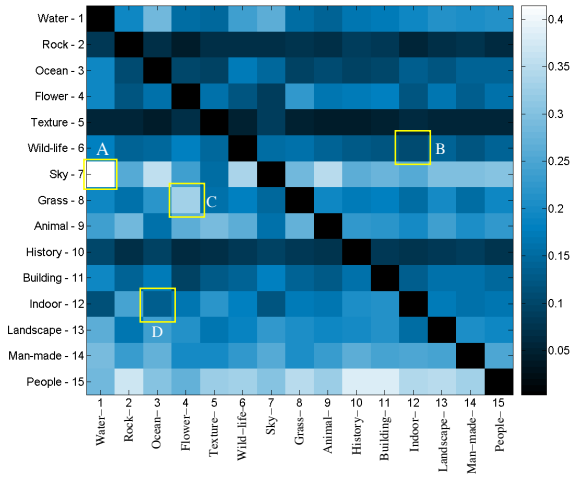

Figure 3: Visualization of $Pr(G_{w_i}=g_i \mid A_{w_j}=1, G_{w_j}=0)$ as obtained empirically with the real-world Alipr data (Sec. 4) for 15 most-frequent words. Highlighted are cases interesting from the meta-learner's viewpoint. For example, (A) is read as "When 'water' is a correct guess, 'sky' is also guessed".

$\{I^{(1)}, \ldots, I^{(L)}\}$, the corresponding word guesses made by the black-box, $\{f_{bbox}(I^{(1)}), \ldots, f_{bbox}(I^{(L)})\}$, and the actual ground-truth/feedback, $\{f_{gtruth}(I^{(1)}), \ldots, f_{gtruth}(I^{(L)})\}$. To make estimation lightweight, and thus scalable, each component of the estimation is based on empirical frequencies, and is a fully deterministic process. Moreover, this property of our model estimation makes it adaptable to incremental or decremental learning, as we will see in Sec. 3.

The probability $Pr(A_{w_j}=1 \mid G_{w_j}=g_j)$ in Eq. 10 can be estimated from the size L training data as follows:

$$\widehat{Pr}(A_{w_j}=1 \mid G_{w_j}=g_j) = \frac{\sum_{n=1}^{L} \mathcal{I}\left\{G_{w_j}^{(n)}=g_j \;\&\; A_{w_j}^{(n)}=1\right\}}{\sum_{n=1}^{L} \mathcal{I}\left\{G_{w_j}^{(n)}=g_j\right\}}$$

Here, $\mathcal{I}(\cdot)$ is the indicator function. A natural issue of robustness arises when the training set contains too few or no samples for $G_{w_j}^{(n)}=1$, where estimation will be poor or impossible. Therefore, we perform a standard *interpolation-based smoothing* of probabilities. For this we require a *prior* estimate, which we compute as

$$\widehat{Pr}_{prior}(g) = \frac{\sum_{i=1}^{K}\sum_{n=1}^{L} \mathcal{I}\left\{G_{w_i}^{(n)}=g \;\&\; A_{w_i}^{(n)}=1\right\}}{\sum_{i=1}^{K}\sum_{n=1}^{L} \mathcal{I}\left\{G_{w_i}^{(n)}=g\right\}} \quad (11)$$

where $g \in \{0,1\}$. For $g=1$ (or 0), it is the estimated probability that a word that is guessed (or not guessed) is correct. The word-specific estimates are interpolated with the prior to get the final estimates as follows:

$$\begin{aligned}&\widetilde{Pr}(A_{w_j}=1|G_{w_j}=g_j) \quad (12)\\ =\; &\begin{cases} \widehat{Pr}_{prior}(g_j) & m \leq 1 \\ \frac{1}{m}\widehat{Pr}_{prior}(g_j) + \frac{m}{m+1}\widehat{Pr}(A_{w_j}=1|G_{w_j}=g_j) & m > 1 \end{cases}\end{aligned}$$

where $m = \sum_{n=1}^{L} \mathcal{I}\{G_{w_j}^{(n)}=g_j\}$, the number of instances out of L where W_j is guessed (or not guessed, depending upon g_j).

The probability $Pr(G_{w_i}=g_i|A_{w_j}=1, G_{w_j}=g_j)$ in Eq. 10

can be estimated from the training data as follows:

$$\widehat{Pr}(G_{w_i}{=}g_i|A_{w_j}{=}1, G_{w_j}{=}g_j) \quad (13)$$
$$= \frac{\sum_{n=1}^{L} \mathcal{I}\Big\{G_{w_i}^{(n)}{=}g_i \;\&\; G_{w_j}^{(n)}{=}g_j \;\&\; A_{w_j}^{(n)}{=}1\Big\}}{\sum_{n=1}^{L} \mathcal{I}\Big\{G_{w_j}^{(n)}{=}g_j \;\&\; A_{w_j}^{(n)}{=}1\Big\}}$$

Here, we have a more serious robustness issue, since many word pairs may not appear in the black-box's guesses. A popular smoothing technique for word pair co-occurrence modeling is *similarity-based smoothing* [7], which is appropriate in this case, since semantic similarity based propagation of information is meaningful here. Given a WordNet-based semantic similarity measure $W(w_i, w_j)$ between word pairs w_i and w_j, the smoothed estimates are given by:

$$\widetilde{Pr}(G_{w_i}{=}g_i|A_{w_j}{=}1, G_{w_j}{=}g_j) \quad (14)$$
$$= \sum_{k=1}^{K} \frac{W(w_j, w_k)}{Z} \widehat{Pr}(G_{w_i}{=}g_i|A_{w_k}{=}1, G_{w_k}{=}g_k)$$

where Z is a normalization factor. When $\widehat{Pr}(\cdot \mid \cdot, \cdot)$ cannot be estimated due to lack of samples, a *prior* probability estimate, computed as in the previous case, is used in its place. The Leacock and Chodorow (LCH) word similarity measure [15], used as $W(\cdot, \cdot)$ here, generates scores between 0.37 and 3.58, higher meaning more semantically related. Thus, this procedure weighs the probability estimates for words semantically closer to w_j more than others.

The estimation of $Pr(\rho_j \mid A_{w_j} = a, \bigcup_{i \neq j}(G_{w_i}{=}g_i), G_{w_j}{=}g_j)$, $a \in \{0, 1\}$ in Eq. 10 will first require a suitable definition for ρ_j. As mentioned, it can be thought of as a numerical abstraction relating the knowledge base to the black-box's guesses. The hope here is that the distribution over this numerical abstraction will be different when certain word guesses are correct, and when they are not. One such formulation is as follows. Suppose the black-box makes Q word guesses for an image I that has word w_j as a correct (or wrong) tag, for $a = 1$ (or $a = 0$). We model the number of these guesses, out of Q, that are semantically related to w_j, using the *binomial distribution*, which is apt for modeling counts within a bounded domain. Semantic relatedness here is determined by thresholding the LCH relatedness score $W(\cdot, \cdot)$ between pairs of words (a score of 1.7, ~ 50 percentile of the range, was arbitrarily chosen as threshold). Of the two binomial parameters (N, p), N is set to the number of word guesses Q made by the black-box, if it always makes a fixed number of guesses, or the maximum possible number of guesses, whichever appropriate. The parameter p is calculated from the training data as the *expected value* of ρ_j for word w_j, normalized by N, to obtain estimates $\hat{p}_{j,1}$ (and $\hat{p}_{j,0}$) for A_{w_j} being 1 (and 0). This follows from the fact that the expected value over a binomial PMF is $N \cdot p$. Since robustness may be an issue here again, interpolation-based smoothing, using a prior estimate on p, is performed. Thus, the ratio of the binomial PMFs can be written as follows:

$$\frac{\widetilde{Pr}(\rho_j \mid A_{w_j} = 1, -)}{\widetilde{Pr}(\rho_j \mid A_{w_j} = 0, -)} = \Big(\frac{\hat{p}_{j,1}}{\hat{p}_{j,0}}\Big)^{\rho_j} \Big(\frac{1 - \hat{p}_{j,1}}{1 - \hat{p}_{j,0}}\Big)^{Q - \rho_j} \quad (15)$$

Finally, we discuss $Pr(h_1, \ldots, h_D \mid A_{w_j}{=}a)$, $a \in \{0, 1\}$, the visual representation component in Eq. 10. The idea is that the probabilistic model for a simple visual representation may differ when a certain word is correct, versus when it is not. While various feature representations are possible, we employ one that can be efficiently computed and is also suited to efficient incremental/decremental learning. Each image is divided into 16 equal partitions, by cutting along each axis into four equal parts. For each block, the RGB color values are transformed into the LUV space, and the triplet of average L, U, and V values represent that block. Thus, each image is represented by a 48-dimensional vector consisting of these triplets, concatenated in raster order of the blocks from top-left, to obtain $(h_1, .., h_{48})$. For estimation from training, each of the 48 components is fitted with a univariate Gaussian, which involves calculating the estimated mean $\hat{\mu}_{j,d,a}$ and std. dev. $\hat{\sigma}_{j,d,a}$. Smoothing is performed by interpolation with estimated priors $\hat{\mu}$ and $\hat{\sigma}$. The joint probability is computed by treating each component as conditionally independent given a word w_j:

$$\widetilde{Pr}(h_1, \ldots, h_D \mid A_{w_j}{=}a) = \prod_{d=1}^{48} \mathcal{N}(h_d \mid \hat{\mu}_{j,d,a}, \hat{\sigma}_{j,d,a}) \quad (16)$$

Here, $\mathcal{N}(.)$ is the Gaussian PDF. So far, we have discussed the static case, where a batch of images are trained on. If ground-truth for some images is available, it can be used to train the meta-learner, to annotate the remaining ones. We experiment with this setting in Sec. 4, to see if a meta-learner built atop the black-box is advantageous or not.

3. META-LEARNING OVER TIME

We now look at image annotation in online settings. The meta-learning framework proposed in the previous section has the property that the learning components involve summation of instances, followed by simple $O(1)$ parameter estimation. Inference steps are also lightweight in nature. This makes online re-training of the meta-learner convenient via incremental/decremental learning. Imagine the online settings presented in Sec. 1 (Fig. 1). Here, images are annotated as they are uploaded, and whenever the users choose to provide feedback by pointing out wrong guesses, adding tags, etc. For example, in Flickr [10], images are publicly uploaded, and independently or collaboratively tagged, not necessarily at the time of uploading. In Alipr [1], feedback is solicited immediately upon uploading. In both these cases, ground-truth arrives into the system sequentially, giving an opportunity to learn from it to annotate future pictures better. Note that when we say of tagging 'over time', we mean tagging in sequence, temporally ordered.

At its inception, an annotation system may not have collected any ground-truth for training the meta-learner. Hence, over a certain initial period, the meta-learner stays inactive, collecting an L_{seed} number of *seed* user feedback. At this point, the meta-learner is trained quickly (being lightweight), and starts annotation on incoming images. After an L_{inter} number of new images has been received, the meta-learner is re-trained (Fig. 4 provides an overview). The primary challenge here is to make use of the models already learned, so as not to redundantly train on the same data. Re-training can be of two types depending on the underlying 'memory model':

- **Persistent Memory**: Here, the meta-learner accumulates new data into the current model, so that at steps of L_{inter}, it learns from all data since the very beginning, inclusive of the seed data. Technically, this only involves incremental learning.

- **Transient Memory:** Here, while the model learns from new data, it also 'forgets' an equivalent amount of the earliest memory it has. Technically, this involves incremental and decremental learning, whereby at every L_{inter} jump, the meta-learner is updated by (a) assimilating new data, and (b) 'forgetting' old data.

3.1 Incremental/Decremental Meta-Learning

Our meta-learner formulation makes incremental and decremental learning efficient. Let us denote ranges of image sequence indices, ordered by time, using the superscript $[start : end]$, and let the index of the current image be L_{cu}. We first discuss incremental learning, required for the case of *persistent memory*. Here, probabilities are re-estimated over all available data upto the current time, i.e., over $[1 : L_{cu}]$. This is done by maintaining *summations* computed in the most recent re-training at l_{pr} (say), over a range $[1 : L_{pr}]$, where $L_{pr} < L_{cu}$. For the first term in Eq. 10, suppressing the irrelevant variables, we can write

$$\widehat{Pr}(A_{w_j} \mid G_{w_j})^{[1:L_{cu}]} = \frac{\sum_{n=1}^{L_{cu}} \mathcal{I}\{G_{w_j}^{(n)} \ \& \ A_{w_j}^{(n)}\}}{\sum_{n=1}^{L_{cu}} \mathcal{I}\{G_{w_j}^{(n)}\}} \quad (17)$$

$$= \frac{\mathcal{S}(G_{w_j} \ \& \ A_{w_j})^{[1:L_{pr}]} + \sum_{n=L_{pr}+1}^{L_{cu}} \mathcal{I}\{G_{w_j}^{(n)} \ \& \ A_{w_j}^{(n)}\}}{\mathcal{S}(G_{w_j})^{[1:L_{pr}]} + \sum_{n=L_{pr}+1}^{L_{cu}} \mathcal{I}\{G_{w_j}^{(n)}\}}$$

Therefore, updating and maintaining the summation values $\mathcal{S}(G_{w_j})$ and $\mathcal{S}(G_{w_j} \ \& \ A_{w_j})$ suffices to re-train the meta-learner without using time/space on past data. The *priors* are also computed using these summation values in a similar manner, for smoothing. Since the meta-learner is re-trained at fixed intervals of L_{inter}, i.e., $L_{inter} = L_{cu} - L_{pr}$, only a fixed amount of time/space is required every time for getting the probability estimates, regardless of the value of L_{cu}. The second term in Eq. 10 can also be estimated in a similar manner, by maintaining the summations, taking their quotient, and smoothing with re-estimated priors. For the third term related to WordNet, the estimation is similar, except that the summations of ρ_j for $A_{w_j} = 0$ and 1, are maintained instead of counts, to obtain estimates $\hat{p}_{j,0}$ and $\hat{p}_{j,1}$ respectively. For the fourth term related to visual representation, the estimated mean $\hat{\mu}_{j,d,a}$ and std.dev. $\hat{\sigma}_{j,d,a}$ can also be updated with values of $(h_1, \ldots, h_{48})$ for the new images by only storing summation values, as follows:

$$\hat{\mu}_{j,d,a}^{[1:L_{cu}]} = \frac{1}{L_{cu}} \Big(\mathcal{S}(h_d)^{[1:L_{pr}]} + \sum_{n=L_{pr}+1}^{L_{cu}} h_d^{(n)} \Big)$$

$$\hat{\sigma}_{j,d,a}^{[1:L_{cu}]} = \sqrt{\frac{1}{L_{cu}} \Big(\mathcal{S}(h_d^2)^{[1:L_{pr}]} + \sum_{n=L_{pr}+1}^{L_{cu}} \big(h_d^{(n)}\big)^2 \Big) - \Big(\hat{\mu}_{j,d,a}^{[1:L_{cu}]}\Big)^2}$$

owing to the fact that $\sigma^2(X) = \mathbf{E}(X^2) - (\mathbf{E}(X))^2$. Here, $\mathcal{S}(h_d^2)^{[1:L_{pr}]}$ is the sum-of-squares of the past values of feature h_d, to be maintained, and $\mathbf{E}(.)$ denotes expectation. This justifies our simple visual representation, since it conveniently allows incremental learning by only maintaining aggregates. Overall, this process continues to re-train the meta-learner, using the past summation values, and updating them at the end, as depicted in Fig. 4.

In the *transient memory* model, estimates need to be made over a fixed number of recent data instances, not necessarily from the beginning. We show how this can be performed efficiently, avoiding redundancy, by a combination of incremental/decremental learning. Since every estimation process involves summation, we can again maintain summation values, but here we need to *subtract* the portion that is to be removed from consideration. Suppose the *memory span* is decided to be L_{ms}, meaning that at the current time L_{cu}, the re-estimation must only involve data over the range $[L_{cu} - L_{ms} : L_{cu}]$. Let $L_{old} = L_{cu} - L_{ms}$. Here, we show the re-estimation of $\hat{\mu}_{j,d,a}$. Here, along with summation $\mathcal{S}(h_d)^{[1:L_{pr}]}$, we also require $\mathcal{S}(h_d)^{[1:L_{old}-1]}$. Therefore,

$$\hat{\mu}_{j,d,a}^{[L_{old}:L_{cu}]} = \frac{1}{L_{ms}+1} \sum_{n=L_{old}}^{L_{cu}} h_d^{(n)}$$

$$= \frac{1}{L_{ms}+1} \Big(\mathcal{S}(h_d)^{[1:L_{pr}]} + \sum_{n=L_{pr}+1}^{L_{cu}} h_d^{(n)} - \mathcal{S}(h_d)^{[1:L_{old}-1]} \Big)$$

Since L_{ms} and L_{inter} are decided *a priori*, it is straightforward to know the values of L_{old} for which $\mathcal{S}(h_d)^{[1:L_{old}-1]}$ will be required, and we store them along the way. Other terms in Eq. 10 can be estimated similarly.

Putting things together, a high-level version of our T/T approach is presented in Algo. 3.1. It starts with an initial training of the meta-learner using seed data of size L_{seed}. This could be accumulated online using the annotation system itself, or from an external source of images with ground-truth (e.g., Corel images). The process then takes one image

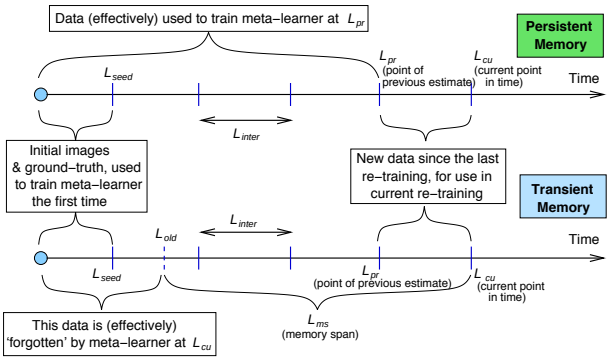

Figure 4: A schematic overview of 'tagging over time'.

Algorithm 1 Tagging over Time

Require: Image stream, Black-box, Feedback
Ensure: Annotation guesses for each incoming image
for $L_{cu} = 1$ to L_{seed} **do**
  Dat(L_{cu}) ← Black-box guesses, feedback, etc. for $I_{L_{cu}}$
end for
Train meta-learner on Dat$(1 : L_{seed})$
repeat {I ← incoming image}
  Annotate I using meta-learner
  if Feedback received on annotation for I **then**
    $L_{cu} \leftarrow L_{cu} + 1$, $I_{L_{cu}} \leftarrow I$
    Dat(L_{cu}) ← Black-box guesses, feedback, etc.
  end if
  if $((L_{cu} - L_{seed})$ modulo $L_{inter}) = 0$ **then**
    if Strategy = 'Persistent Memory' **then**
      Re-train meta-learner on Dat$(1 : L_{cu})$
      /* Use incremental learning for efficiency */
    else
      Re-train meta-learner on Dat$(L_{cu} - L_{ms} : L_{cu})$
      /* Use incremental/decremental learning for efficiency */
    end if
  end if
until End of time

at a time, annotates it, and solicits feedback. Any feedback received is stored for future meta-learning. After gaps of l_{inter}, the model is re-trained based on the chosen strategy.

4. EXPERIMENTAL RESULTS

We perform experiments for the two scenarios shown in Fig. 1; (1) Static tagging, where a batch of images are tagged at once, and (2) Tagging over time (online setting) where images arrive in temporal order, for tagging. In the former, our meta-learning framework simple acts as a *new annotation system* based on an underlying black-box system. We explore whether the meta-learning layer improves performance over the black-box or not. In the latter, we have a realistic scenario that is particularly suited to online systems (Flickr [10], Alipr [1]). Here, we see how the *seed* meta-learner fares against the black-box, and whether its performance improves with newly accumulated feedback or not. We also explore how the two meta-learning memory models, *persistent* and *transient*, fare against each other.

Experiments are performed on standard datasets and real-world data. First, we use the Corel Stock photos, to compare our meta-learning approach with the state-of-the-art. This collection of images is tagged with a 417 word vocabulary. Second, we obtain two real-world, temporally ordered traces from the Alipr system [1], each 10,000 in length, taken over *different periods of time*. Each trace consists of publicly uploaded images, the annotations provided by Alipr, and the user-feedback received on these annotations. The Alipr system provides the user with 15 guessed words (ordered by likelihoods), and the user can opt to select the correct guesses and/or add new ones. This is the feedback for our meta-learner. Here, ignoring the non-WordNet words in either vocabulary (to be able to use the WordNet similarity measure uniformly, and to reduce noise in the feedback), we have a consolidated vocabulary of 329 unique words.

Two different *black-box* annotation systems, which use different approaches to image tagging, are used in our experiments. A good meta-learner should fare well for different underlying black-box systems, which is what we set out to explore here. The first is Alipr, which is a real-time, online system, and the second is a recently proposed approach [8] that was shown to outperform earlier systems. For both models, we are provided guessed tags given an image, ordered by decreasing likelihoods. Annotation performance is gauged using three standard measures, namely *precision*, *recall* and F_1-*score* that have been used in the past. Specifically, for *each* image, *precision*$=\frac{\#(\text{tags guessed correctly})}{\#(\text{tags guessed})}$, *recall*$=\frac{\#(\text{tags guessed correctly})}{\#(\text{correct tags})}$, and F_1-*score*$=\frac{2\times Precision\times Recall}{Precision+Recall}$ (harmonic mean of precision and recall). Results reported in each case are averages over all images tested on.

The 'lightweight' nature of our meta-learner is validated by the fact that the (re-)training of each visual category in [16] and [8] are reported as 109 sec. and 106 sec. respectively. Therefore, at best, re-training will take these times when the models are built *fully in parallel*. In contrast, our meta-learner re-trains on 10,000 images in $\sim$ 6.5 sec. on a *single* machine. Furthermore, the additional computation due to the meta-learner during annotation is negligible.

4.1 Tagging in a Static Scenario

In [8], it was reported that 24,000 Corel images, drawn from 600 image categories were used for training, and 10,000 test images were used to assess performance. We use this system as black-box by obtaining the word guesses made by it, along with the corresponding ground-truth, for each image. Our meta-learner uses an additional $L_{seed} = 2,000$ images (randomly chosen) from the Corel dataset as the *seed* data. Therefore, effectively, (black-box + meta-learner) uses 26,000 instead of 24,000 images for training. We present results on this static case in Table 1. Results for our meta-learner approach are shown for both **Top r** ($r = 5$) and **Threshold r**% (r=60), as described in Sec. 2.1. The baseline results are those reported in [8]. Note the significant jump in performance with our meta-learner in both cases. Effectively, this improvement comes at the cost of only 2,000 extra images and marginal addition to computation time.

Table 1: Results on 10,000 Corel images (static)

Approach	Precision	Recall	F_1-*score*
Baseline [8]	25.38%	40.69%	31.26
Meta-learner (Top r)	32.47%	**74.24%**	45.18
Meta-learner (Thresh.)	**40.25%**	61.18%	**48.56**

Table 2: Results on 16,000 real-world images (static)

Approach	Precision	Recall	F_1-*score*
Baseline [16] (All 15)	13.07%	**81.50%**	22.53
Baseline [16] (Top r)	17.22%	40.89%	24.23
Meta-learner (Top r)	22.12%	47.94%	30.27
Meta-learner (Thresh.)	**33.64%**	58.09%	**42.61**

Next, we experiment with real-world data obtained from Alipr [1], which we use as the black-box, and the data is treated as a batch here, to emulate a static scenario. We use both data traces consisting of 10,000 images each, the tags guessed by Alipr for them, and the user feedback on them, as described before. It turns out that most people provided feedback by selection, and a much smaller fraction typed in new tags. As a result, the *recall* is by default very high for the black-box system, but it also yields poor precision. For each trace, our meta-learner is trained on $L_{seed} = 2,000$ *seed* images, and tested on the remaining 8,000 images. In Table 2, averaged-out results for our meta-learner approach for both **Top r** ($r = 5$) and **Threshold r**% (r=75), as described in Sec. 2.1, are presented alongside the baseline [16] performance on the same data (all 15 and top 5 guesses). Again we observe significant performance improvements over the baseline in both cases. As is intuitive, a lower percentile cut-off for threshold, or a higher number r of top words both lead to higher recall, at the cost of lower precision. Therefore, either number can be adjusted according to the specific needs of the application.

4.2 Tagging over Time

We now look at the T/T case. Because the Alipr data was generated online in a real-world setting, it makes an apt test data for our T/T approach. Again, the black-box here is the Alipr system, from which we obtain the guessed tags and user feedback. The annotation performance of this system acts as a baseline for all experiments that follow.

First, we experiment with the two data traces separately. For each trace, a seed data consisting of the first $L_{seed} = 1,000$ images (in temporal order) is used to initially train the meta-learner. Re-training is performed in intervals of

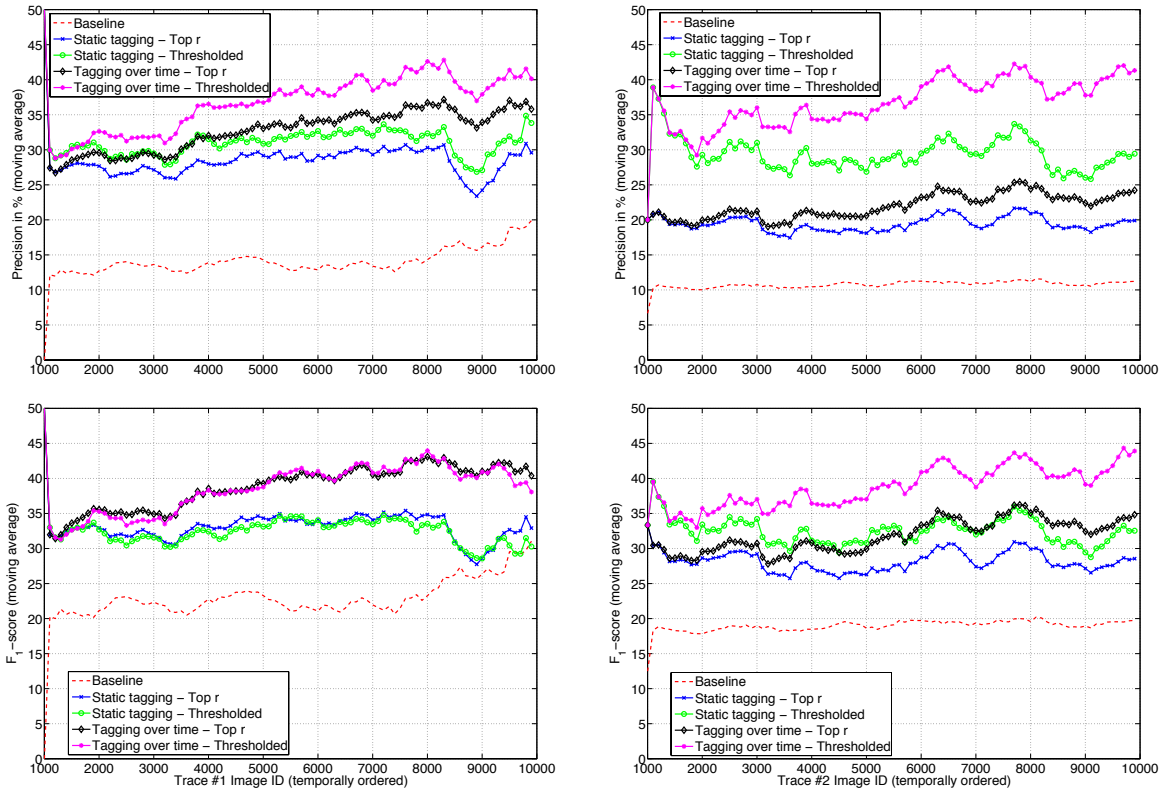

Figure 5: The *precision* and F_1-*score* comparisons for traces #1 (left) and #2 (right).

$L_{inter} = 200$. We test on the remaining 9,000 images of the trace for (a) static case, where the meta-learner is not further re-trained, and (b) T/T case, where meta-learner is re-trained over time, using (a) **Top r** ($r = 5$), and (b) **Threshold r**% (r=75) for each case. For these experiments, the *persistent memory* model is used. Comparison is made using *precision* and $\mathbf{F_1}$-score, with the baseline performance being that of Alipr, the black-box. Here a comparison of *recall* is not interesting because it is generally high for the baseline (as explained before), and it is anyway dependent on the other two measures. These results are shown in Fig. 5. The scores shown are *moving averages* over 500 images (or less, for the initial 500 images).

Next, we explore how the *persistent* and *transient* memory models fare against each other. The main motivation for transient learning is to 'forget' earlier training data that is irrelevant, due to a shift in trend in the nature of images and/or feedback. Because we observed such a shift between Alipr traces #1 and #2 (being taken over distinct time-periods), we merged them together to obtain a new 20,000 image trace to emulate a scenario of shifting trend. Performing a seed learning over images 4,001 to 5,000 (part of trace #1), we test on the trace from 5,001 to 15,000. The results for the two memory models for T/T, along with the static and baseline cases, are presented in Fig. 6. Note the performance dynamics around the 10,000 mark where the two traces merge. While the persistent and transient models follow each other closely till around this mark, the latter performs better after it (by upto 10%, in precision), verifying our hypothesis that under significant changes, 'forgetting' helps to produce a better-adapted meta-learner.

A strategic question to ask, on implementation, is 'How often should we re-train the meta-learner, and at what cost?'. To analyze this, we experimented with the 10,000 images in Alipr trace #1, varying the interval L_{inter} between re-training while keeping everything else identical, and measuring the F_1-score. In each case, the computation time is noted (ignoring the latency incurred due to user waits, treated as constant here). These durations are normalized by the maximum time incurred, i.e., at $L_{inter} = 100$. These results are presented in Fig. 7. Note that with increasing gaps in re-training, F_1-score decreases to a certain extent, while computation time saturates quickly, to the amount needed exclusively for tagging. There is a clear trade-off between computational overhead and the F_1-score achieved. A graph of this nature can therefore help decide on this trade-off for a given application.

Finally, in Fig. 8, we show an image sampling from a large number of cases where we found the annotation performance to improve meaningfully with re-training over time. Specifically, against time 0 is shown the top 5 tags given to the image by Alipr, along with the meta-learner guesses after training over $L_1 = 1000$ and $L_2 = 3000$ images over time. Clearly, more correct tags are pushed up by the meta-learning process, which improves with more re-training data.

5. CONCLUSIONS

We have two main contributions in this paper. First, we have formulated a principled lightweight meta-learning framework for image annotation, and through extensive experiments on two different state-of-the-art black-box annotation systems have shown that a meta-learning layer can vastly improve their performance. Second, we have considered a new annotation scenario which has considerable

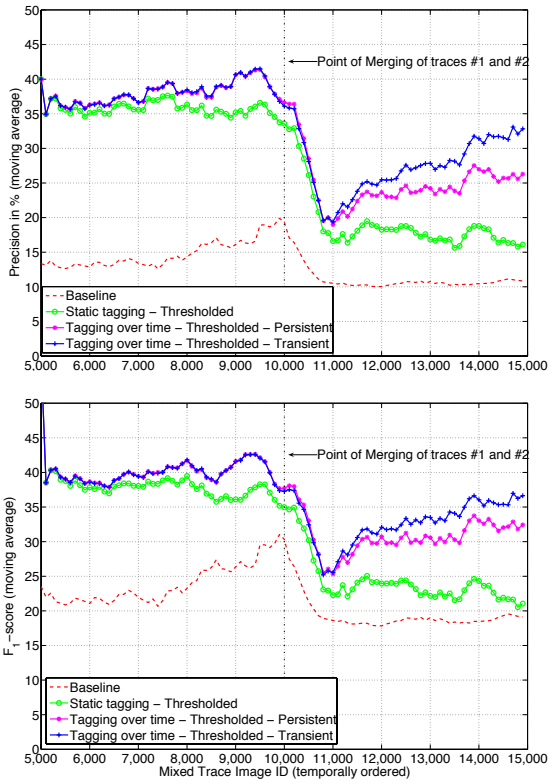

Figure 6: ***Precision* & F_1*-score* for mem. model comparison.**

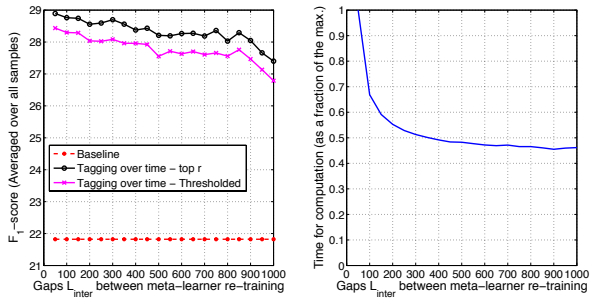

Figure 7: Comparing F_1*-score* & time with varying L_{inter}.

potential for real-world implementation. Taking advantage of the lightweight design of our meta-learner, we have proposed a 'tagging over time' algorithm for efficient re-training of the meta-learner over time, as new user-feedback becomes available. Experimental results on standard and real-world datasets show dramatic improvements in performance. We have proposed and experimentally contrasted two memory models for meta-learner re-training. The meta-learner approach to annotation appears to have a number of attractive properties, and it seems worthwhile to implement it atop other existing systems to strengthen this conviction.

Acknowledgments: The research is supported in part by the NSF under Grant Nos. 0202007 and 0347148.

6. REFERENCES

[1] Alipr. `http://www.alipr.com`, 2006.
[2] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *JMLR*, 3(1107–1135), 2003.
[3] G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Trans. PAMI*, 29(3):394–410, 2007.

Figure 8: Sample annotation results, improving over time.

[4] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.
[5] G. Cauwenberghs and T. Poggio. Incremental and decremental support vector machine learning. In *Proc. NIPS*, 2001.
[6] E. Chang, G. Kingshy, G. Sychay, and G. Wu. CBSA: Content-based soft annotation for multimodal image retrieval using bayes point machines. *IEEE Trans. on Circuits and Systems for Video Tech*, 13:26–38, 2003.
[7] I. Dagan, L. Lee, and F. Pereira. Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34(1-3):43–69, 1999.
[8] R. Datta, W. Ge, J. Li, and J. Z. Wang. Toward bridging the annotation-retrieval gap in image search by a generative modeling approach. In *Proc. ACM Multimedia*, 2006.
[9] S. L. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *Proc. IEEE CVPR*, 2004.
[10] Flickr. `http://www.flickr.com`, Yahoo!, 2005.
[11] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Proc. ICML*, 1996.
[12] Y. Gao, J. Fan, H. Luo, X. Xue, and R. Jain. Automatic image annotation by incorporating feature hierarchy and boosting to scale up svm classifiers. In *Proc. ACM Multimedia*, 2006.
[13] R. Jin, J. Y. Chai, and L. Si. Effective automatic image annotation via a coherent language model and active learning. In *Proc. ACM Multimedia*, 2004.
[14] Y. Jin, L. Khan, L. Wang, and M. Awad. Image annotations by combining multiple evidence & wordnet. In *Proc. ACM Multimedia*, 2005.
[15] C. Leacock and M. Chodorow. Combining local context and wordnet similarity for word sense identification. *C. Fellbaum, Ed., WordNet: An Electronic Lexical Database*, pages 265–283, 1998.
[16] J. Li and J. Z. Wang. Real-time computerized annotation of pictures. In *Proc. ACM Multimedia*, 2006.
[17] X. Li, L. Chen, L. Zhang, F. Lin, and W. Y. Ma. Image annotation by large-scale content-based image retrieval. In *Proc. ACM Multimedia*, 2006.
[18] W. Liu and X. Tang. Learning an image-word embedding for image auto-annotation on the nonlinear latent space. In *Proc. ACM Multimedia*, 2005.
[19] G. Miller. Wordnet: A lexical database for english. *Comm. of the ACM*, 38(11):39–41, 1995.
[20] F. Monay and D. Gatica-Perez. On image auto-annotation with latent space models. In *Proc. ACM Multimedia*, 2003.
[21] D. Silver, G. Bakir, K. Bennett, R. Caruana, M. Pontil, S. Russell, and P. Tadepalli. Inductive transfer: 10 years later. In *Int. Workshop at NIPS*, 2005.
[22] R. Vilalta and Y. Drissi. A perspective view and survey of meta-learning. *AI Review*, 18(2):77–95, 2002.
[23] T. Volkmer, J. R. Smith, and A. Natsev. A web-based system for collaborative annotation of large image and video collections: An evaluation and user study. In *Proc. ACM Multimedia*, 2005.
[24] C. Wang, F. Jing, L. Zhang, and H.-J. Zhang. Image annotation refinement using random walk with restarts. In *Proc. ACM Multimedia*, 2006.
[25] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.
[26] W. Wu and J. Yang. Smartlabel: An object labeling tool using iterated harmonic energy minimization. In *Proc. ACM Multimedia*, 2006.