Personalized, Interactive Tag Recommendation for Flickr

Nikhil Garg
Ecole Polytechnique Fédérale de Lausanne
nikhil.garg@epfl.ch

Ingmar Weber
Ecole Polytechnique Fédérale de Lausanne
ingmar.weber@epfl.ch

ABSTRACT

We study the problem of personalized, interactive tag recommendation for Flickr: While a user enters/selects new tags for a particular picture, the system suggests related tags to her, based on the tags that she or other people have used in the past along with (some of) the tags already entered. The suggested tags are dynamically updated with every additional tag entered/selected. We describe a new algorithm, called *Hybrid*, which can be applied to this problem, and show that it outperforms previous algorithms. It has only a single tunable parameter, which we found to be very robust.

Apart from this new algorithm and its detailed analysis, our main contributions are (i) a clean methodology which leads to conservative performance estimates, (ii) showing how classical classification algorithms can be applied to this problem, (iii) introducing a new cost measure, which captures the effort of the whole tagging process, (iv) clearly identifying, when purely local schemes (using only a user's tagging history) can or cannot be improved by global schemes (using everybody's tagging history).

Categories and Subject Descriptors

H.5 [**Information Interfaces and Presentation**]: H.5.2 User Interfaces; H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing

General Terms

Algorithms, Experimentation, Human Factor, Measurement

Keywords

flickr, tagging systems, tag recommendation, tag co-occurrence

1. INTRODUCTION

Although the state-of-the-art in multimedia retrieval is certainly advancing fast, large scale content-based multimedia retrieval systems remain infeasible. For example, there is still no large-scale image retrieval system, where a user can ask for "all pictures showing a horse". In all current systems, the retrieval process relies on the fact that the keyword "horse", or possibly a related term, had been manually added to all relevant pictures. Then, traditional text-based retrieval techniques are employed.[1] These keywords are usually added in the form of tags. Without these added tags, it would be impossible to find even a single relevant image.

The user's motivation to add such tags is two-fold. First, it allows them to later find their own pictures related, e.g., to a particular event or a particular person. Second, it increases the accessibility to the public, as other people interested in a particular topic can now find relevant images.[2] Recent user studies reveal that users annotate their pictures with the motivation of indeed making them more accessible to others [1]. However, most of the time users add very few tags or even none at all. In the case of the Flickr photo sharing system, at least 20% of public photos have no tag at all[3] and cases with $1-3$ tags constitute 64% of the cases with any tags [9]. One of the reasons for this seems to be that users are often reluctant to enter many useful tags or indeed any at all. Tagging an object takes considerably more time than just selecting it for upload. Also note that any particular object (an image) is only tagged by a single user (the owner). This has to be contrasted with the setting for social bookmarking services such as del.icio.us, where a single object (a webpage) can be tagged by multiple users. Only in the latter case can standard collaborative filtering techniques be applied.

In this work, we analyze methods to support the user during the tagging process. Concretely, we consider the following *interactive tag recommendation problem*: Whenever the user wants to add another tag, we recommend a ranked list of relevant tags to the user. The computation of this list depends on the tags already present. The user can then choose whether (i) to select any item from the list of recommendations, or (ii) to ignore all the recommendations and enter a tag herself. In either case, after the new tag has been added, the list of recommendations is updated in real time and the process is repeated.

[1]In the setting of video retrieval, recent progress has been made by applying speech-to-text conversion to the audio data of a videos. The transcription obtained this way can then be searched just as normal text.

[2]As usual, this causes tag spam to occur, where lots of irrelevant tags are added, as certain people want their objects to be found, regardless of the user's interest [6].

[3]This estimate was obtained by our sample of Flickr from crawling it using friendship links. People who maintain their friendship links are, however, usually more "active" than others and also tend to tag more.

RecSys'08, October 23–25, 2008, Lausanne, Switzerland

This problem is independent of any particular application, but we only evaluated our algorithms in the context of Flickr. Note that the methods we will study are *personalized* in the sense that they explicitly use knowledge about the particular user's tagging behavior in the past. This way they can, for the same input tags, recommend different tags to different users. E.g., for the input tag "jaguar", they will propose "car" to one and "cat" to another user. Similarly, for the tags "lausanne" and "epfl" we could recommend "switzerland" to one user, "suisse" to another and "schweiz" to a third depending on one's language.

The quality of algorithms for this problem can be assessed in two different ways. First, for a certain number of already selected tags, the quality of the recommended list can be evaluated within the usual precision-recall kind of setup known from information retrieval. However, this focuses on one particular point in time is not really adequate, as we are interested in the performance over the whole *sequence* of added tags. Therefore, we also propose and use a second evaluation setup.

In absolute numbers, our best method, called Hybrid, achieves the following performance: When half the tags of a picture are given as the input to the algorithm, and only the tags in the other half are considered as relevant, we achieve an average precision at the first position (P@1) of around 55% for users who have tagged $1-31$ pictures before, 65% for users who have tagged $32-255$ pictures before and around 70% for users with even more tagged images. The typical success among the first five positions (S@5) correspondingly lies between 75% and 90%. Note that, due to how we defined relevance, these are *underestimates* of the system's true performance. As shown in the experimental section, our system performs considerably better than two other schemes for this problem, recently presented at the WWW conference [3, 9].

If we suppose the user has a cost of 10 units (in some cost measure) for typing a tag herself, and a cost of 1 for checking if a given tag is relevant (and clicking it, if it is), then we can reduce the average cost for inputting a tag, averaged over *all* the tags of the picture, from 10 units to 8.4 in a conservative setting and to 6.8 in a more realistic setup.

Apart from the presentation and the analysis of our algorithms, our main contribution is the description of a clean methodology for the problem studied and the identification of key factors governing the performance, such as the "coverage", defined in Section 6.1. Furthermore, we will point out several pitfalls concerning the use of a user study for the problem and we will explain how to obtain conservative performance evaluations without such data. This will allow us to evaluate our methods not only for a few hundred photos (for which the suggested tags were manually classified as relevant or not) but even for thousands of photos.

Note that the algorithms studied are not inherently restricted to tags. They only require "sets of items" and, to give personalized results, for each user a history of past sets of items. Other use cases would involve the suggestion of additional items to buy, as a customer fills her shopping basket. Here, we can use both knowledge about what other people have bought together in the past and about what the particular customer has bought before. This can be seen as a personalized and soft version of association rules mining. By "soft" we mean that there are no hard rules found, which depend on a specific level of "confidence" and "support", but rather just a soft ranked list of recommended items.

The rest of this paper is organized as follows. In Section 2 we discuss work related both to the use of tags on Flickr, as well as to automated tagging in general. In particular, the relation to two previous works on tag recommendation for Flickr will be clarified. In Section 3 we introduce our methodology. Section 4 shows how other classification algorithms could be used for the problem of tag recommendation. Before we describe the algorithms in Section 6, we discuss the (partly novel) evaluation measures in Section 5. Finally, our experimental setup and results are presented in Sections 7 and 8.

2. RELATED WORK

The two pieces of work, which are most closely related to our current work, were both presented at the WWW conference this year [9] [3]. Both study the problem of recommending tags to Flickr users. Apart from the actual methods used to compute recommendations, the main differences to the work in [9] are that (i) our tag recommendation are *personalized*, (ii) the evaluation setup avoids the significant problems of a user study (see Section 3.1), (iii) the removal of all crucial tuning parameters, (iv) the performance study in an interactive setting, (v) a detailed breakdown of the performance across various axes (such as the size of a user's profile and the "coverage" of the input tags). Similarly, the main difference to the work in [3] are points (iii) and (v) above. All three schemes are also directly compared to each other in Section 8. The scheme presented in the current work outperformed both by a significant margin, while also being the simplest of them.

In a broader context, various research projects have exploited the tag information in the Flickr community and other tagging systems to automatically extract useful semantics. The GPS and time stamps of photographs have been used along with the tags to extract events ("NY Marathon") and places ("Logan Airport") [8]. With enough pictures automatically annotated with GPS or time data, such ideas could also be applied to our problem. E.g., a picture whose GPS coordinates are in Switzerland is more likely to be tagged with "switzerland". Similarly, the tag "winter" will be more prominently used (on the Northern hemisphere) in January than in July. Although we plan to experiment with these ideas in the future, we did not consider them for this work, as more straightforward personalization already gave very satisfactory results.

Automatically assigning/suggesting tags to *blog posts* [7, 10] is related to our problem. But as blog posts contain full text, finding similar posts and hence related tags is easier than for pictures. Similarly, automatic tagging of web pages can be improved/personalized, if the system has access to the surfer's desktop [2]. If we had additional knowledge from external sources about a particular user, for example from email conversations, then such techniques could also lead to an improvement for our problem. Here, the biggest problem would be how to obtain such data for many users.

In settings where several users collaboratively tag *the same* items, such as for social bookmarking sites, yet other techniques can be applied [11]. Algorithms for mining association rules [5] are also relevant. But they (i) are comparably slow, (ii) would not give any result if the initial set of tags does not occur in any picture, and (iii) would not generate a ranked result list. Still, we might explore adaptations of those techniques in the future.

3. EVALUATION METHODOLOGY

To evaluate the quality of the recommended tags, one needs to know which tags a user would have entered or selected for a given picture. Here two approaches are possible. First, one could evaluate the relevance of the returned tags via a user study and, second, one can work with *only* the tags originally provided by the Flickr user.

3.1 User Study - Why not to Trust it

Given a particular image and a particular tag one would at first think that any intelligent user could easily judge whether this tag is relevant to this picture or not. However, as we will argue here, *only* the actual owner can ultimately decide this question.

Multi-lingual tags: Given the picture of a Swiss flag, the tag "switzerland" is somehow obviously relevant for the picture. However, some users might use the tag "suisse", "schweiz" or "svizzera"[4]. Given only the picture, it is *impossible* to judge the relevance of such terms. One "solution" would be to consider only (American) English tags as relevant, but this is an unsatisfactory approach.

Missing context: The tag "holidays" for a picture taken in, say, Berlin will only make sense to the user, if the image was taken while on holidays. But this contextual information is impossible to obtain for any external human judge. Similarly, one cannot simply assume that the tag "holidays" is clearly relevant for any picture showing a beach, as the owner of the picture might live at this location. The same problem of missing context applies to tags such as "me" or "friends" or any personal name or even most place names.

Missing meta data: Often professional users tag pictures with the brand name of the camera (e.g., "nikon") or with information about the shutter speed or the focal settings. Given only the picture itself, the relevance of such tags is impossible to judge, but a diligent judge could try to obtain such information from meta data, if it is available.

Differences of level of detail: Consider a picture of the cathedral in Lausanne, so "lausanne" is probably a relevant tag. What about "vaud" (the Swiss canton Lausanne is in)? What about "switzerland", "europe", "earth" or "universe"? All of this are "correct" in some sense. But for some (Swiss) users even the tag "vaud" might be redundant. A similar problem occurs with standard portraits. Should an ordinary portrait be tagged as "nose", "eyes", "mouth", "nostrils", "eyelashes" etc. or simply as "face"? What would be considered a relevant tag by the actual user is unclear.

Interdependence of tags: What if in the list of suggested tags for a black and white picture there are the tags "blackandwhite", "black", "white", "bw". Should all of them be counted as relevant? What about "usa", "unitedstates", "unitedstatesofamerica", "america" and "us"? What if only the tag "plant" and not "power" is suggested for the image of a power station? What about just "engine" and no "fire"?

Note that these cases are *not* rare exceptions, but they actually constitute the *majority* of cases. We therefore believe that simple user studies, such as used in [9], do not give a reliable estimation of the quality of a system, especially if the category "good" on the scale "very good", "good", "not good" and "don't know" is counted as relevant in all experiments. Our own belief is that they tend to give overestimates of the true quality of a system, as hundreds of tags can be viewed as "good in some sense" for any given picture.

If one still wants to make use of a user study, it is imperative to give clear and unambiguous guidelines to the human judges about how to evaluate the relevance of a tag, in particular addressing all the cases above. Our own approach, discussed in the next section, avoids such problems.

[4]The term "Switzerland" in French, German and Italian respectively.

3.2 Using Only Given Tags

For any picture we *only* consider tags already added by the original user as relevant. A subset of them is then given as input to the system to generate a list of recommended tags. *Only* the remaining tags, which were not given to the system, are considered as relevant. All other tags are considered as not relevant.

Note that this is the standard machine learning approach for classification: Some labels (tags in our case) are given to the system, which then learns a model. Given this model, the system generates possible labels, whose quality is evaluated using hold-out data, not used in the training phase.

In our setting this gives an *underestimation* of the performance of the system as (i) all relevant tags were actually added by the user, who presumably considered them relevant enough to add, and (ii) there might be other tags, which the user would consider relevant.

3.3 Special Tags and Tagging of Sets

To be absolutely sure to not overestimate the quality of a system, the following two issues need to be addressed.

Tags, which the user cannot decide to add: Flickr has a lot of special tags, such as "supershot" or "abigfave", which are awarded to high quality pictures. These tags are *not* added by the user herself, when she uploads the picture, but only later after a sufficient number of people has acknowledged the quality of the picture and the picture is then invited to special groups.
To suggest such tags to the user, simply does not make any sense, as the user cannot simply choose to add such a tag (or is at least not supposed to). Therefore, we removed a list of the 15 most prominently used tags ("15fave", "abigfave", "anawesomeshot", "aplusphoto" and 11 others) from the consideration of the system. These tags, even if present in the original picture, are counted as non-existent.

Tagging many pictures at once: Flickr allows the batch upload of pictures, where the same set of tags can be added to all pictures at once. Especially, for schemes using the user's profile to suggest tags, this can lead to an overestimate of the quality of a scheme. If we happen to use one of these pictures for the evaluation, and if we give, say, half of its tags to the system, then the system can note that there are already a number of pictures with all of these tags present. So it can easily suggest the remaining tags. However, this would be an artifact of the evaluation setup, as all these pictures were actually tagged *at the same time*, so that the case we are evaluating would never occur in practice. Therefore, we again take a conservative approach, and only consider the *distinct* tag sets for each user. So if a user used the tag combination "lausanne" and "lake" several times, we only count it once, as we cannot know, whether it was created by batch

tagging. This will further ensure that our reported numbers are a lower bound for the real setting.

4. USING CLASSIFICATION ALGORITHMS

As it might not be obvious, we will demonstrate how any classification algorithm can be applied to the problem of tag suggestion, as long as it can cope with extremely high dimensional but sparse feature vectors and a huge number of possible classes. Recall, that any classification algorithm first trains on labeled data, that is, feature vectors with ground truth label. Then, once whichever algorithm has learned a model, the algorithm can predict the label for a new feature vector.

The main "trick" in mapping our problem to this setting is to use tags both as features and as labels. A feature vector will then have dimensions corresponding to the tag vocabulary, where all entries are binary. Concretely, a picture tagged with "a", "b" and "c" would lead to a training set of three feature vectors (corresponding to "a","b", to "a","c", and to "b","c") with the three labels "c", "b" and "a" respectively.

Given a set of tags already input by the user for a new picture, the algorithm can then predict a new tag. As most classification algorithms internally compute some soft, i.e. non-binary, relevance score for each possible class, we can obtain a ranked list of tags to suggest.

5. EVALUATION MEASURES

We used two different kinds of evaluation setups. The first is a standard information retrieval setup for evaluating the quality of a ranked list, which is well-established in the literature. The second setup is especially tailored for our setting and takes the *whole* input process into account.

5.1 Standard IR Setting

In this setting, we evaluate the quality of a single ranked list of tags when a certain number of tags have already been provided by the user. The quality of this list is evaluated using the following classical metrics.

P@1 - "precision at one": The percentage of runs, in which the first recommended tag was relevant. It is, by definition, equal to S@1.

P@5 - "precision at five": The percentage of relevant tags among the first five, averaged over all runs.

S@5 - "success at five": The percentage of runs, in which there was at least one relevant tag among the first five returned tags.

MRR - "mean reciprocal rank": If the first relevant returned tag is at rank r, then the MRR is $1/r$, where this measure is also averaged over all the runs.

5.2 Input Cost

While the previous measures all looked at the static case of evaluating a single ranked list of tags, the measure we will introduce here takes the whole input sequence into account.

At each point of the tagging process, even when no tags have been added so far, a ranked list of recommended tags is presented to the user. The user then goes through this list from the top and checks each tag, one after the other, for its relevance. The "cost" for this checking is essentially the time it takes to read the tag and we charge a unit cost for this action. Once the user has checked a tag, she can select it by clicking, for which we do not charge any further cost. If the user does not find a single relevant tag in the list of k tags presented to her, she first pays a cost of k for checking all tags and then an additional cost of t, where $t > 1$, for typing a relevant tag herself. Regardless of whether she could simply click a recommended tag or whether she had to type a tag herself, at the end there is one additional given input tag and the list of recommended tags is updated.

For fixed values of k and t we can thus look at the average cost for adding a single tag. In the best case, this cost is close to 1, as the user can often simply select a tag from close to the top of the list by clicking. A cost of t can be obtained trivially when no recommendations are shown and the user has to type all the tags herself. If the list of recommendations only contains irrelevant tags, then a cost of $k + t$ will be incurred. Note that the cost of entering a tag will, in practice, depend on the number of tags already entered.

Several refinements for this model are possible. Firstly, we can take the length of the tag manually added by the user into consideration, and have a tag-dependent t. Secondly, we could also charge more or less than 1 for the cost of reading a tag. For example, there might be a certain "annoyance factor" associated with some tags (e.g., obscene tags), or there might be an "educational factor" associated with other tags (when the system suggests a tag the user likes, but would not have thought of herself).

For our evaluation, we used the simple model with $t = 10$ (the typing cost) and $k = 5$ (the length of the list of suggestions). The particular value of k was chosen as it seemed to be the case that, if a relevant tag is found by our system, it is found among the top 5 tags. The value of t was chosen to agree, at least approximately, with the typing overhead and also with the missed opportunity of exploring other tags. Also note that the value of k could be set by the user. There could be a short list of high-quality suggestions, and then the option to click "more" to see further recommendations.

6. EVALUATED ALGORITHMS

6.1 Notation and Terminology

Let the number of different tags in the system, i.e. the tag vocabulary, be n. Let m be the number of pictures used by the system[5]. $m^{(x)}$ denotes the number of pictures tagged with x and $m_u^{(x)}$ denotes the number of pictures in the collection of user u which are tagged with x. We will call the set of tags already input by the user u, for which we want to suggest further tags, a *query* and we denote it by q_u. q_u is simply an n-dimensional binary vector which has 1's in only the dimension for the given tags. $|q_u|$ is the number of input tags. The algorithms will make crucial use of (at least one of) the following entities:

H_u - the history of user u. Concretely, H_u is a matrix with dimensions $n \times m_u$, where m_u is the number of pictures tagged by user u in the past. H_u is a binary matrix for which an entry (i, j) is 1, if and only if user u tagged picture j with tag i. Most of the time, u will be the user for whom we are generating a list of tags to suggest. In this case,

[5]More correctly, we use the number of tag sets (see Section 7.2), to avoid overestimating the quality of our methods.

any information derived from H_u is referred to as "local". Similarly, if the user considered is a different user, we call such information "non-local" or "global".

G - the global history. Concretely, G is a matrix with dimensions $n \times m$, defined in a similar way to H_u. Information derived from G is also referred to as "global".

C - the global co-occurrence matrix. C is defined as GG^t and an entry (i, j) counts on how many images the tags i and j co-occur in the global history. Similarly, we can define $C_u = H_u H_u^t$ for a specific user u. Note that both of these matrices will be dominated by their diagonal entries, which simply count how often a certain tag x was used ($m^{(x)}$ and $m_u^{(x)}$).

We also need to define the important notion of *coverage* of a query q_u. A query q_u is said to have a coverage of v_q[6], if the user u has a picture which contains v of the $|q_u|$ input tags, but no picture which contains $v+1$ of these tags. The intuition is that if for a set of 5 input tags there is already a picture containing all 5 of these tags (and probably others), then we can simply "copy-and-paste" the remaining tags of that picture to get good suggestions. So a low coverage v_q signifies a potentially hard instance.

6.2 Algorithm 1: Naive Bayes as a Local Scheme

One ingredient of our best performing scheme is the use of a Naive Bayes classifier to obtain tag recommendations based on H_u. See [4] for details about Naive Bayes. Concretely, we attribute the following score $s_q(x)$ to tag x for input query q:

$$s_q(x) = P(x) \prod_{i \in q} P'(i|x) \tag{1}$$

Here, $P(x) = H_u(x,x)/m_u = m_u^{(x)}/m_u$ is the (estimated) probability of tag x being used by user u and $P'(i|x)$ is the smoothed probability of seeing tag i (which is present in the query) given tag x. Concretely, $P'(i|x) = \lambda P(i) + (1-\lambda)P(i|x)$ where $P(i|x) = H_u(x,i)/H_u(x,x)$. In all of our experiments, we used a value of $\lambda = 0.1$ to avoid zero probabilities. When there is no input tag, i.e. $|q| = 0$, or none of the input tags has been used by the user in the past, then we simply set $s_q(x) = P(x)$ and rank according to the frequency with which a tag was used in the past.

In Table 1 this scheme is referred to as *Local*.

6.3 Algorithm 2: tf-idf Based Global Scheme

This scheme uses *only* global information and does not personalize the suggestions in any way. It starts by deriving a new matrix $\hat{C}$ from C in two steps. First, all diagonal values of C are set to zero. Note that the diagonal values (i) dominate the matrix and (ii) are not directly useful for generating recommendations, as they do not contain any information about the co-occurrence of *different* tags. Second, each column of this new matrix is normalized, i.e. scaled, so that the maximum in each column is 1.[7] The vector of scores s_q is then computed as

$$s_q = (\hat{C}q) \cdot \text{idf} \tag{2}$$

[6]The coverage depends on the user u, but this additional subscript is dropped for simplicity.

[7]Tags which never co-occur with any other tag are useless and are removed at the beginning.

Here, the "$\cdot$" stands for the component-wise product of two vectors and $\text{idf}(x) = \log(m/m^{(x)})$ is a vector of "inverse document frequencies" (or rather "inverse tag frequencies"). Note that $\hat{C}$ can be seen as a normalized version of a standard document-term matrix, so that this scheme is just a simple tf-idf retrieval. Also observe that whereas in the Naive Bayes scheme the contributions from the various input tags (the $P'(i|x)$) are *multiplied*, here the corresponding contributions are *added*. We also experimented with using the Naive Bayes Formula 1 for the global setting, but this gave considerably worse results.

In Table 1 this scheme (not using Naive Bayes) is referred to as *Global*.

6.4 Algorithm 3: The Best of Two Worlds

In settings with a high coverage v_q (see the end of Section 6.1), local schemes such as the one in Section 6.2 work well, and global information does not lead to a (significant) performance improvement. However, in settings with a low coverage or even $v_q = 0$, the local information can only serve to give information about the overall frequency of individual tags, but does not provide any useful co-occurrence information. In these cases, the use of global information gives a significant improvement in performance.

Using this simple observation we define a new hybrid scheme as follows.

$$s_q = \mu_q s_q^\ell + (1 - \mu_q) s_q^g \tag{3}$$

Here, the s_q^ℓ is the s_q defined in the (local) Equation 1 and, correspondingly, the s_q^g is the s_q defined in the (global) Equation 2. The query (and user) dependent weight μ_q is *not* a tuning parameter, but it is defined as $v_q/|q|$, i.e. as the highest percentage of input tags covered by any of the users past pictures.

Note that it would also have been sensible to choose the aggregation parameter μ in dependence of the size of the user's history, i.e. depending on m_u. However, as Table 3 shows, the performance difference between the local and the global scheme depends much more distinctly on the coverage v_q than on m_u.

Note that the *only* tuning parameter involved in the whole scheme is the λ for the smoothing in Naive Bayes (see Section 6.2). Here, we found the performance to be fairly robust within a range of roughly $[0.05, 0.2]$ and we expect a value of $\lambda = 0.1$ to also work well for other collections.

In Table 1 this scheme is referred to as *Hybrid*.

6.5 Algorithm 4: Our Old Scheme

In our previous work [3], we presented a different approach to using both global and local information. This scheme works in three phases.

First, it uses a ranking of local tags according to the following formula.

$$s_q^\ell = (q_u + 0.8 \cdot (C_u q_u)) \cdot \text{idf} \tag{4}$$

Here, $\text{idf}(x) = \log(m/m^{(x)})$. Of course, the addition of q_u is irrelevant, as tags originally present will never be counted as valid suggestions, but we also use this vector s_q^ℓ as an *expanded query* in the following steps.

Second, it finds a set of top $k = 10$ relevant groups using the cosine product $H_v s_q^\ell$. [8] Note that as groups on Flickr also

[8]Query-independent components, only involving H_v, H_u

come with pictures, they can be treated just as regular users. However, using related groups (and not users) gives slightly better results, as groups are more semantically focused than individual users. For each of the 10 groups we obtain suggestions from their history H_v according to the same Equation 4, where q_u is now replaced by the (expanded) s_q^ℓ and the C_u of the original user is replaced by C_v. All of these 10 different sets of scores where then added into a single "remote" set of scores s_q^r.

Third, it combines the initial ranking s_q^ℓ and the rankings obtained from the groups s_q^r in a query-independent manner. The final scores are then given by

$$s_q = 0.85 \cdot s_q^\ell / \max(s_q^\ell) + 0.15 \cdot s_q^r / \max(s_q^r) \quad (5)$$

In Table 1 this scheme is referred to as *Old*.

6.6 Algorithm 5: Recent WWW Scheme

Recently, a research group from Yahoo also presented a method for tag suggestion at the WWW conference. They use the global co-occurrence matrix C in a similar way to Equation 2. Additionally, they also use the values $m^{(x)}$ to punish both very frequent and very infrequent tags. The details can be found in [9].

The main differences are the following. First, they use 4 parameters of which at least 3 have an crucial impact and need to be fine-tuned. To report fair performance numbers for their system, we had to invest quite a bit of time into finding appropriate values. The parameters used in [9], namely $m = 25$, $k_s = 9$,$k_d = 11$ and $k_r = 4$, gave a P@1 of a mere .05. The parameters we finally used were $m = 25$, no k_s, $k_d = 7$, $k_r = 4$, which led to a P@1 of around .15 (see Table 1). Second, they cannot easily sum the contributions pertaining to different input tags, as is done in Equation 2, as these contributions are not normalized. However, without removing the diagonal values of the matrix C, these values cannot be normalized in a meaningful way, as we also observed in our own experiments. Thus, they have to combine different contributions by looking at the rank of a tag in each contributing list. Besides the computational overhead of an additional sort operation for each such list, this then creates the need for the parameter k_r, which governs the combination of such ranked lists.

In Table 1 this scheme is referred to as *WWW*.

7. EXPERIMENTAL SETUP

We obtained all of our data for the experiments using the publicly available Flickr API[9]. In total, we downloaded tag information for 50 million public photos with at least one tag. Out of these, we extracted 24 million tag sets (see Section 7.2). These pictures/tag sets came from 77,166 different users. In our experiments we looked at breakdowns according to two dimensions, discussed in the following two sections.

7.1 Sets of Users

Users on Flickr differ widely with respect to (i) how many pictures they uploaded to Flickr and (ii) how likely they are to add tags to these pictures. For our study, only users who have tagged at least one picture were considered. Figure 1 shows the distribution of big and small users.

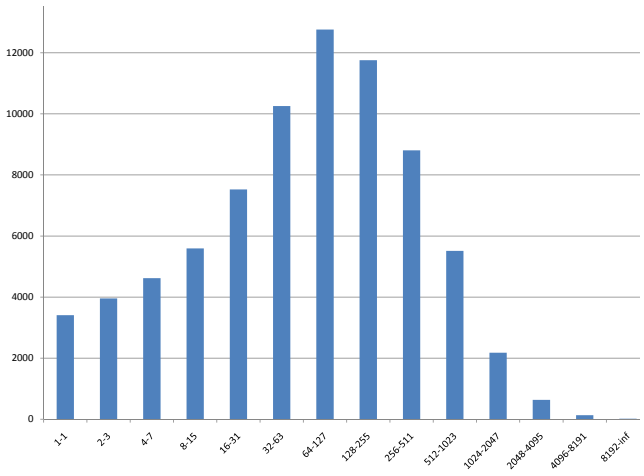

Figure 1: Number of users (on the y-axis) with a certain number of distinct tag sets (on the x-axis). We only used the *distinct* tag sets for each user to avoid inflating the performance numbers. See Section 7.2 for details.

A picture taken randomly from the set of tagged pictures is far more likely to come from a user, who has tagged a lot of pictures in the past than from a user who has only tagged relatively few pictures. So sampling pictures uniformly at random does not lead to a uniform sampling distribution over the users. As methods using local information derived from the user's past history are more likely to give good results for such "big" users, we tried to avoid this inflation of the reported numbers by sampling users from the following three categories.

Small users: These are users with only $1 - 31$ previously tagged pictures[10]. Due to the sparseness of local information for them, methods using global information will work better for them.

Medium users: These are users with between 32 and 255 previously tagged pictures. Figure 1 shows that typical users (who tag at all) fall into this category.

Big users: These are users with at least 256 previously tagged pictures. For these users, methods using local information will work best.

7.2 Sets of Pictures

The number of tags used per picture differs widely between 1 and several hundreds. The typical range of the number of tags is between 1 and 6 [9]. However, for our (conservative) evaluation setup, pictures with less than 4 tags cannot be used in a meaningful way, as there is not enough data to evaluate the list of recommended tags.

This does *not* mean that we assume in our experiments that the user has already entered 4 or more tags. In the basic setting, we give half the tags (rounded up) to the system and then evaluate the quality of the list. This is the setting used for Tables 1 and 3. However, in Table 2 we also give a

and not q_u, can also be used, but this did not lead to an performance improvement, at least not when the query q_u is already expanded to s_q^ℓ.

[9]http://www.flickr.com/services/api/

[10]Wherever it says "picture" or "image" it should say "distinct tag set". See Section 7.2 for a detailed explanation.

breakdown of the quality of the returned lists for any given number of input tags, thereby analyzing the whole tagging process, and not just a snapshot. This should be contrasted with the approach used in [9], which used *all* the given tags as input and then used a user study to evaluate the quality of the suggestions.

For our experiments, we bucketed the pictures into three categories. One for pictures with $4-7$ tags, one with $8-15$ tags and one with $16-\infty$ tags. As we only use given tags as relevant tags, there are more possible tags to be found for pictures with more tags, and so the performance for these pictures is higher (see Table 3).

7.3 Choice of Input Tags

For our basic setup we chose half the given tags (rounded up) as input to whichever system we evaluated. These tags were obtained in an alternating manner, meaning, that the first, third, fifth and so on tag would be used as input, whereas the second, fourth etc. tag were used as hold-out data to evaluate the performance.

For the Input Cost setting (Section 5.2), we started with *no* tag given as input. Here, the system simply ranked tags according to their past usage frequencies. Then, we followed the "flow" of either typed or clicked tags. This way, input tags were added one after the other, and the exact sequence depended on the previous output of the system.

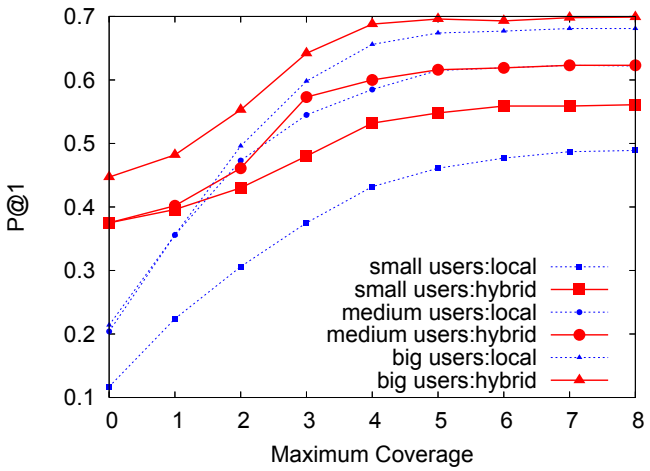

Figure 2: A detailed analysis of when our *Hybrid* scheme (solid red line) has the biggest performance advantage compared to the simple local scheme (dotted blue line). The P@1 (y-axis) results shown are for pictures with $8-15$ tags coming from different kinds of users (square = "small" users, circle = "medium" users, triangle = "big" users). To restrict the evaluation to a certain maximum coverage (x-axis), all other pictures in the user's profile with a higher coverage than the threshold were artificially ignored by the system. For low coverage, which corresponds to the situation of both new users and a new tag topic for old users, the improvement in P@1 between the local and the hybrid scheme is between 100% and 200%. This can be seen by comparing, for a fixed symbol, the two different colors (or line types).

	Number of tags	P@1 , MRR				
		local	global	hybrid	old	www
Small	$4-7$	.33, .39	.20, .27	*.37, .44*	.26, .33	.10, .13
	$8-15$	.49, .55	.38, .48	*.56, .64*	.41, .49	.21, .26
	$16-\infty$	.59, .65	.58, .69	*.71, .79*	.53, .61	.32, .39
Medium	$4-7$	*.51*, .58	.25, .34	.50, *.59*	.37, .44	.09, .13
	$8-15$	.62, .70	.38, .49	*.62, .71*	.55, .64	.16, .20
	$16-\infty$	.73, .79	.58, .69	*.78, .84*	.70, .77	.28, .34
Big	$4-7$	.50, *.60*	.28, .37	*.50, .60*	.37, .46	.09, .12
	$8-15$	.68, .76	.45, .56	*.70, .77*	.59, .67	.20, .25
	$16-\infty$	.75, .82	.62, .72	*.76, .84*	.69, .77	.31, .38

Table 1: Comparison of several methods for $1,000$ pictures in each set. Since no limit was set for the coverage of tags, the hybrid method only slightly improves over the purely local method for medium and big users. See Table 3 for a breakdown of the first three methods for different levels of coverage.

8. SUMMARY OF RESULTS

Our "Hybrid" scheme outperforms both our own previous system ("Old") and another recently presented scheme ("WWW"). See Table 1. For cases of low coverage, "Hybrid" improves dramatically over the simple "Local" scheme. See Table 3 and Figure 2. For a simple model of measuring the cost of inputting tags, our scheme improves the average cost of tagging by at least 16% in a conservative setting and by at least 32% in more realistic setting. See Table 2.

9. FUTURE WORK

In this work, we implictly defined a tag as "good", if it would be selected by a user, when recommended to her. In the future, we plan to investigate other definitions of "goodness" related to either (i) the usefulness to *other* users (e.g., using query logs for image searches) or (ii) the navigability of the user's collection (e.g., using entropy measures to avoid that the user tags *all* her pictures in a very similar fashion).

10. REFERENCES

[1] M. Ames and M. Naaman. Why we tag: motivations for annotation in mobile and online media. In *The conference on Human factors in computing systems (CHI'07)*, pages 971–980, 2007.

[2] P. A. Chirita, S. Costache, W. Nejdl, and S. Handschuh. P-tag: large scale automatic generation of personalized annotation tags for the web. In *The 16th international conference on World Wide Web (WWW'07)*, pages 845–854, 2007.

[3] N. Garg and I. Weber. Personalized tag suggestion for flickr. In *The 17th international conference on World Wide Web (WWW'08)*, pages 1063–1064, 2008.

[4] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[5] J. Hipp, U. Güntzer, and G. Nakhaeizadeh. Algorithms for association rule mining. *SIGKDD Explorations Newsletter*, 2(1):58–64, 2000.

[6] G. Koutrika, F. A. Effendi, Z. Gyöngyi, P. Heymann, and H. Garcia-Molina. Combating spam in tagging systems. In *The 3rd international workshop on*

	Number of tags	Method	Max. coverage 0				Max. coverage 1				Max. coverage 2				Max. coverage 4				Max. coverage ∞			
			P@1	P@5	S@5	a.c.	P@1	P@5	S@5	a.c.	P@1	P@5	S@5	a.c.	P@1	P@5	S@5	a.c.	P@1	P@5	S@5	a.c.
Small users	$4-7$	local	.07	.03	.15	.0	.19	.08	.30	.5	.28	.11	.40	.9	.33	.17	.45	1.3	.33	.17	.45	1.3
		hybrid	.20	.03	.37		.25	.12	.45		.33	.15	.52		.37	.20	.55		.37	.20	.55	
	$8-15$	local	.12	.05	.20	.0	.22	.11	.37	.6	.31	.15	.44	1.0	.43	.25	.56	1.8	.49	.32	.75	2.3
		hybrid	.38	.26	.63		.40	.27	.64		.43	.28	.68		.53	.35	.74		.56	.40	.75	
	$16-\infty$	local	.11	.05	.20	.0	.18	.10	.32	.5	.27	.15	.42	1.0	.36	.24	.53	1.8	.59	.51	.72	6.1
		hybrid	.58	.48	.86		.58	.48	.86		.59	.49	.86		.62	.50	.88		.71	.63	.91	
Medium users	$4-7$	local	.16	.07	.28	.0	.31	.13	.49	.9	.46	.18	.62	1.5	.51	.23	.68	1.9	.51	.23	.68	1.9
		hybrid	.25	.12	.45		.30	.14	.54		.45	.19	.66		.50	.23	.71		.50	.23	.71	
	$8-15$	local	.20	.09	.35	.0	.36	.18	.56	.9	.47	.25	.68	1.7	.59	.36	.78	2.7	.62	.42	.82	3.3
		hybrid	.38	.26	.66		.40	.27	.68		.46	.29	.72		.60	.37	.81		.62	.42	.84	
	$16-\infty$	local	.23	.12	.40	.0	.38	.20	.60	.9	.49	.28	.71	1.8	.60	.39	.81	3.1	.73	.61	.89	6.4
		hybrid	.58	.45	.83		.59	.46	.85		.60	.46	.86		.65	.49	.88		.78	.64	.93	
Big users	$4-7$	local	.15	.06	.27	.0	.30	.13	.49	1.0	.43	.19	.65	1.7	.50	.24	.73	2.3	.50	.24	.73	2.3
		hybrid	.28	.14	.47		.33	.15	.55		.44	.20	.68		.50	.24	.74		.50	.24	.74	
	$8-15$	local	.21	.10	.38	.0	.36	.17	.59	1.0	.50	.26	.73	1.9	.66	.41	.85	3.1	.68	.46	.87	3.7
		hybrid	.45	.29	.72		.48	.30	.74		.55	.32	.78		.69	.40	.85		.70	.45	.87	
	$16-\infty$	local	.25	.12	.43	.0	.41	.20	.62	1.0	.51	.27	.71	1.9	.61	.39	.82	3.4	.75	.62	.90	6.5
		hybrid	.62	.48	.86		.62	.49	.87		.64	.49	.88		.69	.51	.89		.76	.63	.93	

Table 3: Detailed performance results for our new hybrid scheme compared to the purely local scheme it comprises. The breakdown is across various user profiles and number of tags per pictures. In all cases, half the tags of a picture (rounded up) were given to the system as input and the other half was withheld as labeled test data. For each (user group)-(tag range) combination, we ran our experiments for $1,000$ pictures, giving a total of $9,000$ pictures. A maximum coverage of ∞ means that all of the pictures previously tagged by the user were used. This is the *normal* case when the system is used in practice. For the other values of maximum coverage we *artificially* reduced the user's history and only used those pictures which covered at most the corresponding number of input tags. This was done to get an insight into when exactly only local information is (in-)sufficient. A maximum coverage of 0 means that only pictures without *any* of the given tags were used. In this case, the local method can merely compute the frequency of the tags used in the past, derived from the remaining images. For a maximum coverage of 0, the hybrid scheme reduces to the *purely* global scheme. The "a.c." stands for the average coverage for the $1,000$ pictures in the corresponding set.

Adversarial information retrieval on the web (AIRWeb'07), pages 57–64, 2007.

[7] G. Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *The 15th international conference on World Wide Web (WWW'06)*, pages 953–954, 2006.

[8] T. Rattenbury, N. Good, and M. Naaman. Towards automatic extraction of event and place semantics from flickr tags. In *The 30th international conference on Research and development in information retrieval (SIGIR'07)*, pages 103–110, 2007.

[9] B. Sigurbjörnsson and R. van Zwol. Flickr tag recommendation based on collective knowledge. In *The 17th international conference on World Wide Web (WWW'08)*, pages 327–336, 2008.

[10] S. C. Sood, K. J. Hammond, S. H. Owsley, and L. Birnbaum. TagAssist: Automatic Tag Suggestion for Blog Posts. In *The 1st International Conference on Weblogs and Social Media (ICWSM 2007)*, 2007.

[11] Z. Xu, Y. Fu, J. Mao, and D. Su. Towards the semantic web: Collaborative tag suggestions. *Collaborative Web Tagging Workshop at WWW2006*, 2006.

Tag position	Cost for tag
Pos 1	4.6 (4.6)
Pos 2	4.7 (4.7)
Pos 3	6.0 (6.0)
Pos 4	6.9 (6.9)
Pos 5	8.1 (8.1)
Pos 6	9.1 (8.5)
Pos 7	10.3 (9.1)
Pos 8	11.4 (9.2)
Pos 12	11.7 (11.4)
Pos 14	12.1 (–)

Table 2: Details of the Input Cost measure (see Section 5.2) for our *Hybrid* scheme and for $1,000$ pictures with $8-15$ tags from "medium" users. The cost continues to rise as, in our conservative setting, there are fewer and fewer relevant tags left to find. The numbers in parentheses refer to the setting, where we stop suggesting, when there are only three tags left to find. The average cost for entering a tag, averaged over all $1,000$ pictures, is 8.4 in the original setting and 6.8, when the suggestion process stops when there are only three tags left.