

# TagEz: Flickr Tag Recommendation

Ashton Anderson and Karthik Raghunathan and Adam Vogel

## Abstract

User tagging of multimedia has emerged as the premier organizational tool for large sets of rapidly growing information. We present a tag prediction system for images on Flickr which combines both linguistic and vision features. We describe methods for building language models of tags on Flickr, similar in spirit to traditional language modeling in the NLP community. We evaluate our system against held-out Flickr data, and achieve competitive performance.

## Introduction

Freeform keyword annotations have become a hallmark of Web 2.0-style Internet applications, allowing users to organize large, rapidly changing datasets. Multimedia tagging in websites such as Flickr and YouTube is a key component in image retrieval and search. These new platforms present several opportunities and challenges: we now have unparalleled access to images with keyword annotations, but there are no rules for how a tag applies to a photo, resulting in a very organic dataset. Although the large size of these datasets is attractive for machine learning, their scale presents challenges to our algorithms. The ubiquity of this problem has attracted interest in the vision (Barnard et al. 2003), web (Sigurbjörnsson and van Zwol 2008), and database (Heymann, Ramage, and Garcia-Molina 2008) communities.

In this paper we present TagEz, a tag prediction system for Flickr. Given a Flickr image, which has possibly already been tagged, TagEz outputs a ranked list of five candidate tags which might also apply. Previous work typically focuses either on the vision (Barnard et al. 2003; Li and Wang 2006) or language (Sigurbjörnsson and van Zwol 2008) portions of the tagging problem. We combine both vision and language features into one global model.

Using the language model described in (Sigurbjörnsson and van Zwol 2008) and the vision system from (Li and Wang 2006), we utilize rank aggregation methods from social choice theory (Borda 1781) to combine them into a final ranking.

Secondly, we describe a Greasemonkey script which a user can download that seamlessly adds our tag prediction system to the normal Flickr tag interface. Using AJAX technology, this allows us to asynchronously run our image analysis (which takes on average less than 2 seconds per novel image), returning our list in near real-time to the user.

Lastly, we present an empirical evaluation of the TagEz system using standard Information Retrieval metrics. These results show that the language component outperforms the vision component, and that their combination actually underperforms just the language component. We discuss a method for using held out data in a lower bound evaluation to avoid the labors of manual annotation.

## System Architecture

TagEz consists of a language, vision, and aggregation component. Figure 1 displays our architecture, with an example taken from Flickr. Note that in this example the user has already applied some tags to the image, which the language component uses to find commonly co-occurring tags throughout the rest of Flickr. Alternatively, the vision component learns correlations between the contents of the image and tags, but ignores any previously applied tags. This allows for a purely vision based approach when the user has not supplied any tags, and a surprisingly accurate language component when the user has input a few tags.

Outside of the box in Figure 1, we also have a Flickr crawler and our Greasemonkey front-end, which allows users to actually apply our system to Flickr.

## Front End

For actual usage of the TagEz system, we wrote a Greasemonkey (Boodman 2005) script which adds our functionality directly to the Flickr website<sup>1</sup>. Greasemonkey is a Mozilla Firefox extension that allows on-the-fly changes to web pages after a user downloads a small javascript-like script. Then when a user clicks the

<sup>1</sup>Available from <http://cs.stanford.edu/people/acvogel/tagez/>

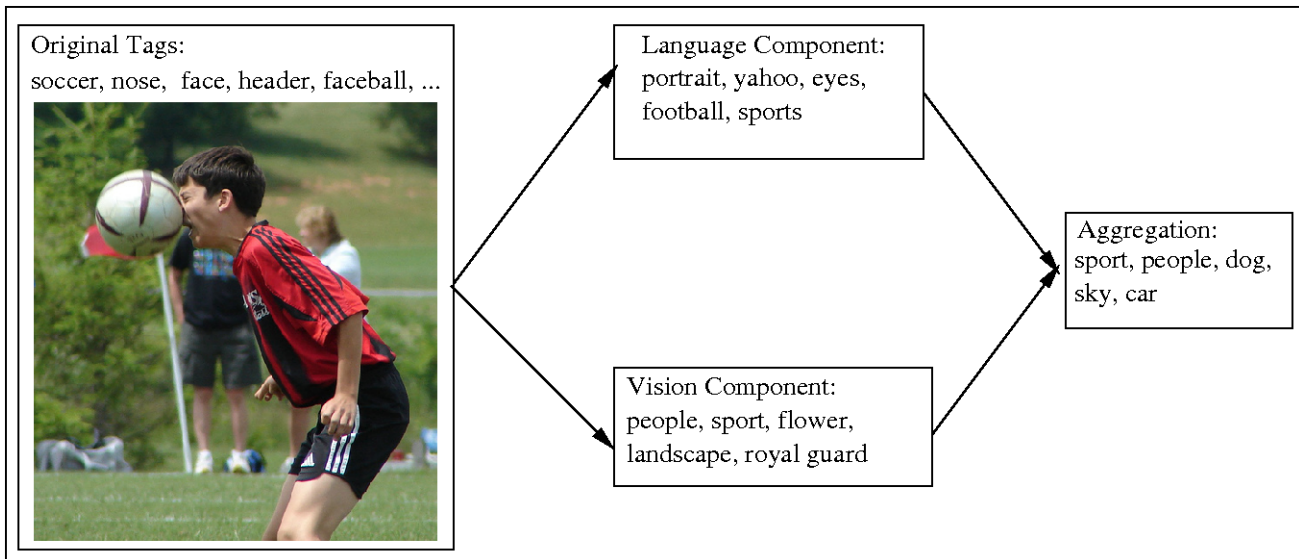


Figure 1: Recommendation System Architecture

“Add Tag” link on a photo in Flickr, this Greasemonkey script adds a line which calls and displays the TagEz recommendations. Critically, we use an asynchronous javascript call (AJAX), which does not require a new pageload to display our results.

More specifically, this Greasemonkey AJAX call is handled by a Jython webserver, which in turn calls our TagEz Java implementation. Jython is an implementation of Python on the JVM, which allows native calls of Java libraries directly from Python. When the AJAX call returns, the javascript in our Greasemonkey script displays the relevant links, which a user can then click to apply the tags, in the same manner that Flickr “common tags” are usually applied.

## Flickr Crawler

Our Flickr language model requires many successive API calls to gather the relevant unigram and bigram statistics for tags of interest. This is not a problem in offline evaluation and usage, but real world users won’t wait four minutes. Thus we cache all Flickr information we gather, and wrote a simple crawler to gather the relevant statistics for common tags.

We use the Flickr Hot Tag list as our crawler seed set, which is a list of 200 commonly used tags. For each of these seeds, we query Flickr for the photos with that tag. We gather other tags that occur in this result list, and add them to our tag list. Furthermore, we also use the Flickr getRelatedTags function to expand our tag list. We repeat this procedure for each of the tags in our seed set and crawl a total of 7863 tags.

Given this set of tags and photos they occur in, it is straightforward to compute our language model statistics. We count occurrences (unigrams) using a simple tag search on Flickr, and count co-occurrences (bi-

grams) with a conjunctive tag query. In the case of cache misses, we simply revert to the relevant Flickr API calls.

## Language

The purpose of the language component of our system is to compute a set of tags that are likely to be relevant to a particular image, given the set of tags  $U$  already defined by the user. For each user-defined tag  $u_i \in U$ , we compute a list of related tags  $R_i$ . These lists  $R_i$  are then merged into a single output list  $O$  of recommendations. We discuss these two steps in turn.

### From user-defined tag to related tags

The complete freedom Flickr offers its users in regards to tagging leads to a huge but noisy and inconsistent tag space. Because of this immense size, tag co-occurrence is a natural metric to use. The *co-occurrence* between two tags is the number of images that contain both tags. By themselves, raw co-occurrence scores are not very meaningful since they ignore the frequency of the individual tags. To account for this we normalize these scores by the frequency of the tags. In the literature there exist two different classes of normalized metrics: symmetric and asymmetric (Sigurbjörnsson and van Zwol 2008). We consider one of each.

The symmetric metric we use is the Jaccard coefficient, which is defined as follows:

$$P(t_j|t_i) := \frac{|t_i \cap t_j|}{|t_i \cup t_j|} \quad (1)$$

Unlike raw co-occurrence, the Jaccard coefficient is a meaningful measure. However, it tends to measure how similar two tags are: if two tags have a high Jaccard score, then they almost always occur in the data set as

a pair, and one will almost never occur in the absence of the other. Our goal is not to find *similar* tags, but to find *relevant* ones. Asymmetric metrics are one way of doing this.

The asymmetric metric we use is given by:

$$P(t_j|t_i) := \frac{|t_i \cap t_j|}{|t_i|} \quad (2)$$

As the notation suggests, we can interpret this measure as the probability of a photo being annotated with  $t_j$  given that it is annotated with  $t_i$  (Sigurbjörnsson and van Zwol 2008).

By looking at pairs of words scored by both the Jaccard coefficient and the above asymmetric metric, we observed that the asymmetric metric was much more suited to our task than its symmetric counterpart. As mentioned above, the Jaccard coefficient tends to score similar or synonymous tags highest, which from our perspective is less interesting. We chose to use the asymmetric coefficient for measuring co-occurrence, and throughout the rest of this paper references to “co-occurrence score” will mean the asymmetric calculation.

As mentioned above, our decision to use a co-occurrence metric was partially motivated by Flickr’s enormity. However, crawling all of Flickr wasn’t feasible. Given a tag  $t_i$ , finding the relevant tags  $R_i$  corresponding to it would have required a number of searches on the order of the number of tags in Flickr (around 3.7 million (Sigurbjörnsson and van Zwol 2008)). Doing this for any significant number of tags would require an astronomical number of calls to the Flickr API and computer time. Instead, we use a method in the Flickr API that returns some of the most relevant tags for any given tag. This function is computed by deep in-house analysis of the full tag graph. For each user-defined tag  $u_i$ , we only consider these tags as possible candidates for inclusion in the corresponding list of related tags  $R_i$ . This allows us to cut down on the number of other tags to consider from millions to dozens, while still using the aforementioned co-occurrence metrics to decide which tags are most relevant.

### Merging the lists of related tags

After coming up with a list  $R_i$  of relevant tags (each with a score) for each user-defined tag  $u_i$ , the last step is to aggregate them together to form the final recommendations  $O$ . In all of the following methods, we compute a final score for each tag then sort the tags in descending order.

1. **Vote** The vote method is the simplest possible method and we use it as a baseline. A user-defined tag  $u_i$  “votes” for tag  $t$  if  $t$  appears in  $R_i$ . The final score for a tag  $t_i$  is the total number of votes it receives.
2. **Sum** In the sum method, a tag’s final score is the sum of its scores in all lists  $R_i$  in which it appears. So within a list  $R_i$ , the related tags are given votes

weighted by their score instead of all related tags getting equal votes.

3. **Promotion methods** Sigurbjörnsson and Zwol introduced a “promotion” score based on the overall frequency of tags in Flickr (tags are penalized for being either too frequent or too infrequent) and the rank in which tags appear in the  $R_i$  lists. This can then be aggregated by vote or sum, as above (Sigurbjörnsson and van Zwol 2008).
4. **Aggregation of the above** We aggregated all four of the above methods using Borda voting (see the Rank Aggregation section).

## Vision

Our vision system learns a predictive model of keyword tags from an image feature representation. We use the Automatic Linguistic Indexing of Pictures - Real Time (ALIPR) system developed by Li et al. (Li and Wang 2006). This system has several desirable properties: it uses a fairly shallow feature representation which is not fitted specifically to any domain, and furthermore ALIPR is fairly quick, taking an average of 1.4 seconds to predict tags for a novel image.

We use the ALIPR system as a black-box, where we input an image from Flickr and get back a list of 50 tags and their corresponding confidence scores as output by the model<sup>2</sup>.

Space considerations prevent us from describing this component in great detail, but the feature representation is a mixture of LUV color features and texture features which are formed from wavelet coefficients in high frequency bands from the original pixel data. This results in a 6-dimensional feature vector for each pixel. ALIPR next clusters these pixels into contiguous image segments. Using this feature representation, ALIPR next learns a generative model of  $p(\text{image}|\text{tag})$ , and chooses tags which maximize the posterior of the image features.

However, ALIPR is only trained on 332 keywords from the Corel image dataset, a commonly used gold-standard image/keyword corpus (Blei and Jordan 2003). Although they evaluate their system against Flickr in (Li and Wang 2006), it’s not clear why they do not train on all of Flickr. This severely limits the impact that the vision component can have on our overall system, as there is oftentimes not much overlap between the language and vision outputs.

## Rank Aggregation

Once the vision component and the language component have outputted their recommendations, the system must aggregate these two ranked lists in some logical way. This problem crops up in many natural settings in information retrieval (for example aggregating search engine results). We use a simple algorithm from the IR

<sup>2</sup>Thanks to James Wang for giving us access to the ALIPR API.

literature called *Borda voting*, which originated in election theory (Borda 1781). In each list (of size  $n$ ), the tags are assigned a decreasing number of points. The top-ranked tag is given  $n - 1$  points, the second-ranked tag is given  $n - 2$  points and so on until the last tag which is given no points. Then the points for each tag are summed and the tags are sorted in decreasing order. If the lists have different numbers of elements, we set  $n$  to be the size of the longest list.

## Evaluation

In this section we present our quantitative evaluation of our tag prediction methods. We started with a set of 28 seed tags, and for each seed tag we randomly chose around 35 images with this tag which had at least 9 tags. This yielded a test set of 924 images. For each test image, we randomly held out one fifth of the tags, treating them as the “test” tags. Since Flickr places no restrictions on the tags users can annotate their pictures with, many tags are very infrequent and virtually “unpredictable”. We filter these out by calling Flickr’s `getRelated` function on the test tags and filtering tags that have no related tags. This heuristic is based on the observation that tags with no related tags in the tag graph are typically obscure and infrequent tags.

We then ran each prediction method on the test images and compared the output to the held-out test tags. A predicted tag is judged as incorrect if it is not present in the held-out set. Since the held-out tags need not be the only tags which could apply to an image, our evaluation method actually gives a lower bound on our system’s performance.

To evaluate the results, we used standard Information Retrieval metrics: the Mean Reciprocal Rank (MRR), the success at rank  $k$  ( $S@k$ ), and the precision at rank  $k$  ( $P@k$ ). The MRR is defined as the reciprocal of the rank of the first relevant (i.e. held-out) tag, averaged over all test photos. Success at rank  $k$  is 1 if a held-out tag was ranked in the top  $k$  results and 0 if not, averaged over all test photos and precision at rank  $k$  is the number of held-out tags ranked in the top  $k$  results, again averaged over all test photos.

Figure 2 displays our results. Surprisingly, the sum method consistently outperformed the baseline vote method, indicating that the asymmetric co-occurrence scores we used are appropriate and helpful in evaluating how relevant tags are. In contrast to the results in (Sigurbjörnsson and van Zwol 2008), the promotion methods actually hurt performance. The vision component did not perform well under our evaluation metrics, probably because the vision component’s vocabulary is restricted to the Corel image dataset’s 330 tags. This restriction severely limited the vision component’s possible success under the IR metrics: only 237 of the 924 test images contained held-out tags in the vision component’s vocabulary. If the vision component had been trained on Flickr (or if we had used human evaluation) the vision component would probably have received much higher scores. Because of this,

the entire system (both vision and language components aggregated using Borda voting) underperformed the language component by itself. This suggests that the aggregation method we used was too crude.

## Conclusion

In this paper we presented TagEz, a system for tag recommendation incorporating both language and vision components. In the language component, a natural asymmetric co-occurrence metric is used for quantifying how related one tag is to another. We narrow our attention of potentially related tags to only those that are relatively “close” in the tag graph, which is computed by an in-house Flickr function. Then we aggregate these lists together using various methods and output one final ranking. The vision component use basic texture and image segmentation to recommend potential tags. These two disparate recommendations are then aggregated into one final list using a rank aggregation algorithm borrowed from election theory.

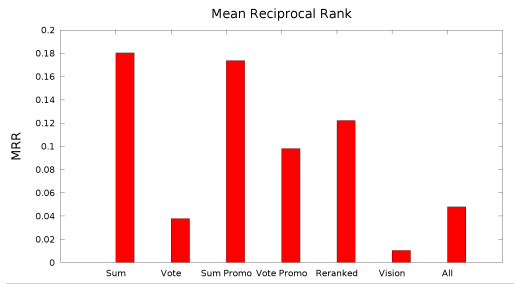
We evaluated our system using a test set of 924 images with some of the original tags for each image held-out. Then we compared the system’s output with the held-out tags only counted direct matches as successes (thus our evaluation was a lower bound). We used standard Information Retrieval metrics to quantify the quality of our tag recommendation methods.

We found that the sum method of aggregating related tag lists within the language component outperformed all other tag recommendation methods, including the combination of the language and vision components. The main reason for the underperformance of the vision component is that it was not trained on Flickr. The aggregation method we used, Borda voting, also disregarded the scores that were given to it.

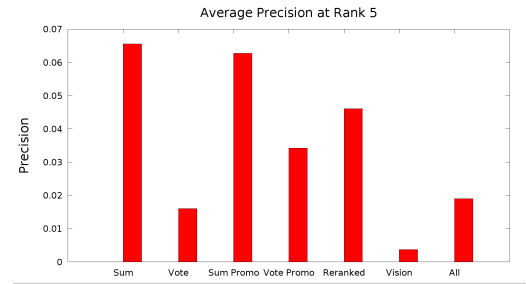
TagEz leaves several areas open to improvement. Firstly we could train the vision component on all of Flickr so that it can handle all tags. TagEz uses a primitive method of aggregating the two components’ recommendations; refined aggregation would improve performance. Another possible extension is to use the output of the language component to narrow down candidate tags for the vision component.

## References

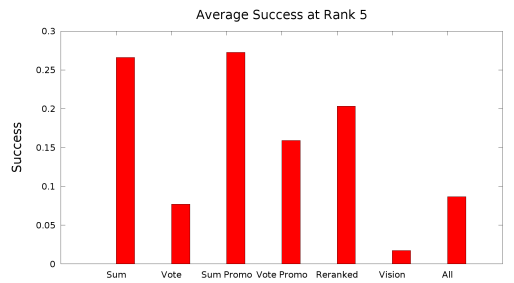
- Barnard, K.; Duygulu, P.; Forsyth, D.; Freitas, N. D.; Blei, D. M.; K, J.; Hofmann, T.; Poggio, T.; and Shawe-taylor, J. 2003. Matching words and pictures. *Journal of Machine Learning Research* 3:1107–1135.
- Blei, D. M., and Jordan, M. I. 2003. Modeling annotated data. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, 127–134. New York, NY, USA: ACM Press.
- Boodman, A. 2005. Greasemonkey. <http://www.greasespot.net/>.
- Borda, J. C. 1781. Memoire sur les elections au scrutin. In *Histoire de l’Academie Royale des Sciences*.



(a) Mean Reciprocal Rank



(b) Precision at Rank 5



(c) Success at Rank 5

Figure 2: Experimental Results

Heymann, P.; Ramage, D.; and Garcia-Molina, H. 2008. Social tag prediction. In *Proceedings of the 31st annual international ACM SIGIR conference*.

Li, J., and Wang, J. Z. 2006. Real-time computerized annotation of pictures. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, 911–920. New York, NY, USA: ACM.

Sigurbjörnsson, B., and van Zwol, R. 2008. Flickr tag recommendation based on collective knowledge. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, 327–336. New York, NY, USA: ACM.