Rethinking the ESP Game

Ingmar Weber*
EPFL
Lausanne, Switzerland
ingmar.weber@epfl.ch

Stephen Robertson
Microsoft Research
Cambridge, UK
ser@microsoft.com

Milan Vojnović
Microsoft Research
Cambridge, UK
milanv@microsoft.com

September 2008

Technical Report
MSR-TR-2008-132

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

Abstract — The ESP Game [15] was designed to harvest human intelligence to assign labels to images - a task which is still difficult for even the most advanced systems in image processing [2, 8]. However, the ESP Game as it is currently implemented encourages players to assign "obvious" labels, which are most likely to lead to an agreement with the partner. But these labels can often be deduced from the labels already present using an appropriate language model and such labels therefore add only little information to the system.
We present a language model which, given enough instances of labeled images as training data, can assign probabilities to the next label to be added. This model is then used in a program, which plays the ESP game *without looking at the image*. Even without any understanding of the actual image, the program manages to agree with the randomly assigned human partner on a label for 69% of all images, and for 81% of images which have at least one "off-limits" term assigned to them.
We then show how, given any generative probabilistic model, the scoring system for the ESP game can be redesigned to encourage users to add less predictable labels, thereby leading to a collection of informative, high entropy tag[1] sets. Finally, we discuss a number of other possible redesign options to improve the quality of the collected labels.

Author Keywords
ESP Game, Image Labeler, Tagging, Flickr

ACM Classification Keywords
H.1.1 Information Systems: Systems and Information Theory; H.1.2 Information Systems: User/Machine Systems

INTRODUCTION

The ESP Game [15] is one of the best-known examples of successful "crowd-sourcing", where the human intelligence of thousands of contributors is harvested for a task which is still difficult for machines: labeling images [2, 8]. Assigning labels to images is useful as otherwise there is little chance to retrieve an image relevant for a certain query. In the original paper [15] evidence is presented that for a selection of 9 labels all images, which were assigned one of these labels, are indeed relevant for the corresponding query. However, it is at least questionable, how much a large image repository benefits if the label "car" is correctly assigned to an unlabeled image. Microsoft's Live Image Search[2] currently returns 150 million results for this query. If the main purpose of the ESP Game is indeed to label images for search purposes, then one can argue that adding *informative* tags such as "red bmw" or even "talbot 1923"[3] is more beneficial and that users adding such tags are adding the most value.

We show that the ESP Game in its most popular implementation fails to collect such informative labels.[4] Namely, we show that (i) the sets of tags already present can be generated from a low entropy distribution and (ii) new tags added by players are highly predictable given *only* the "off-limits" terms, which are the tags already assigned to the image. To demonstrate the second point, we implemented a small program, which we will refer to as "robot". This robot plays the ESP game *without deriving any knowledge from the image itself* and by only using the list of off-limits terms. Underlying both the entropy estimates and the labels used by the robot is a generative probabilistic model, derived from 13K images obtained while playing this game.

Google apparently noted these shortcomings and, in May 2007, introduced different scores for different labels according to the "specificity" of the term[5]. These scores vary between 50 and 150 points. However, as our analysis will show, (i) this is not a strong enough differentiation, (ii) it punishes terms too much which are globally unspecific, but which add relevant information for the particular context, and (iii) the current scores are *not* directly linked to the degree of predictability of a label. We show how to assign scores optimally for a simple model of the ESP game. Furthermore, we discuss a number of possible changes to the game, such as pairing experts for certain topics and introducing time limits before certain terms become "active".

The rest of the paper is structured as follows. In the following section we discuss work related to (i) the ESP Game, (ii) tag suggestion or (iii) tagging behavior of users with respect to navigability or other aspects of our work. Then, we discuss in more detail shortcomings of the current implementation of the ESP Game. The next section then presents our probabilistic model, which uses a Naive Bayesian setup. After this, we present both the implementation details and, more importantly, the results of our robot. The following section then uses the data obtained by the robot and looks at the entropy of the game, i.e. how diverse the sets of off-limits terms are and how much uncertainty exists concerning the labels entered by human players. Thereafter, we analyze optimal scoring schemes for the ESP Game, when the objective is to obtain high entropy tag sets. Finally, we discuss a number of extensions and other practical aspects of the ESP Game.

RELATED WORK

The ESP Game [15], where two players are randomly paired up and have to agree on appropriate labels for images, is often cited as a successful application of a game to use human intelligence and time for solving tasks, which are intractable by current computer technology. In this work we will show that, although the idea underlying the game is an extremely powerful one, more care needs to be taken in the design as the tags which are most likely agreed upon, are *not* the tags which add the most information to the system. The idea of a "bot" playing the game was also already mentioned

[1] We use the terms "label" and "tag" interchangeably.
[2] http://images.live.com
[3] http://en.wikipedia.org/wiki/Talbot
[4] We used the version licensed by Google and available at http://images.google.com/imagelabeler/, rather than the version available at http://www.gwap.com/gwap/gamesPreview/espgame/, as the first is vastly more popular.
[5] http://en.wikipedia.org/wiki/Google_Image_Labeler#History

in [15]. There the idea was to use recorded sequences of tags suggested by humans during previous rounds of play, when there are not enough human players in the system.

Games "with a purpose" have also been applied to obtain common sense facts ("Verbosity" [17]), to locate objects within an image ("Peekaboom" [18]), to tag music ("TagATune" [9]), to trace the shapes of objects ("Squigl"[6]), to elicit human-transcribed data for automated directory assistance ("People Watcher" [10]) and to get descriptions, rather than mere labels, for images ("Phetch" [16]). All of these games share the properties that (i) players share the common goal of "agreeing" on certain things, (ii) players are matched randomly, and (iii) no communication is allowed, as this would make the agreement trivial and prone to spamming. Our arguments apply at least partly also to the tagging of music (where labels such as "loud" or "fast" are probably of little use). Interestingly, the "Phetch" game for describing images avoids most of the problems discussed, as it implicitly involves other players trying to find a *particular* images for a given description. This avoids low-precision descriptions such as "A man standing next to a woman.".

Our proposal, to use probabilistic models and information theory to *quantify* the amount of information added by humans to a system, also has implications for the "Mechanical Turk" [3], where humans are given financial incentives to perform AI-complete tasks. In such a setting it might be desirable to compensate people more, if they take on tasks where predictions made by a machine are most inaccurate.

Closely related to our study of inferring the next label to be added, is the issue of *tag suggestion* [6, 13, 7, 14]. In fact, such schemes could also be directly employed by our robot to play the ESP Game. Though we did use a scheme from [6], trained on data from Flickr before we had enough data from the ESP Game to train our model on, we ultimately used only our model as it directly gave probabilities. The model we used is also similar to but different from a model used in [6]. The biggest differences are that (i) we give a clear definition of what exactly is modeled and (ii) we work not only with the *ranking* of tags but with the probabilities.

The idea of looking at the entropy of the tag distribution and its relation to navigability for collections of tagged objects was looked at in [4]. Whereas they looked at the entropy for *individual* tags over a *shared* collection of items, it would be interesting to investigate, using our model, how the entropy for *sets* of tags for *personal* collections of items relates to the navigability of such collections. This is closely related to what actually constitutes a "good" label for an object.

This later question was investigated in [11], where the focus was on which kinds of labels might be found useful by a user for at least *some* object, rather than a particular one. Related to the issue of the quality of labels is the question of *why* users tag [1] and how their usage patterns is influenced by other users and suggestions made by the tagging system [12, 14].

As far as our estimates of the entropy of the labels of the ESP Game are concerned, these estimates are always with respect to our model. An improved model would lead to a higher predictability and to an even *lower* estimate of the entropy. The possibility of using *humans* to obtain entropy estimates was discussed in [5]. Such an approach, involving volunteers, is also applicable in our setting, but requires modifications as, e.g., humans can be expected to know the whole English alphabet but cannot be expected to be aware of every imaginable tag.

SHORTCOMINGS OF THE ESP GAME

If one looks at how people label images via the ESP Game[7], then one quickly notices the following.

- There is a lot of redundancy in the tag sets. That is, often synonyms are present and images are labeled, e.g., as both "man" and "guy". Of all 496 (out of 14.5K) images labeled as "guy" 81% were also labeled as "man", although only up to five off-limits terms are shown.
- Even when tags are not exactly synonyms, they are often "to be expected" given the other tags, so that images are labeled as "water", "blue", "sky" and "clouds". E.g., 68% of all the 85 images labeled as "clouds" had also been labeled as "sky".
- There is a tendency to match on colors. See Table 1 for details on the distribution of colors in labels assigned by the ESP game.
- People tend to add more generic labels such as "building" as opposed to "terraced house".

The common reason for these points is that it is far more likely for two people to agree on a general term (and in particular on colors), than to agree on more specific terms. In a sense, experts or anybody deviating from the standard tagging behavior is punished by the system. Even if a player knows that a particular image depicts an oak, it will be pointless for her to enter this term, as the chances to agree with her partner are considerably smaller than for "tree", "plant", "leaves" or "green". Note that none of the assigned tags are wrong in any sense, but the question arises, whether one has to rely on humans to obtain them. It should also be made clear that people are indeed more likely to *search* for general terms than for concrete terms, so that terms such as "man" or 'tree" could be deemed more valuable for a search application. But for these general terms there are already millions of publicly available and searchable images online.

A PROBABILISTIC TAGGING MODEL

What Needs to be Modeled

We want to meaningfully assign a probability to the event that "term t will be added to the tag set S by humans presented with the image and with the set S". However, this as

[6] http://www.gwap.com/gwap/gamesPreview/squigl/

[7] We use the terms "ESP Game" and "Google Image Labeler" interchangeably, although all of our experiments were only run for the latter, more popular version of the game.

Label	%-age all	%-age images
black	3.3	14.7
red	2.2	9.8
blue	2.2	9.8
white	1.8	8.1
green	1.3	6.0

Table 1. Distribution of some color related tags among the off-limits terms for 14.5k images with at least one off-limits label. The first percentage refers to the total number of tag occurrences among the off-limits terms made up by the particular term. The second percentage refers to the number of images with at least one tag, which have already been assigned this tag. Over 10% of all off-limits labels are colors.

it stands is ill-defined for several reasons. First, it is unclear if "being added" is to be interpreted as "being added as the immediate next tag" or if it refers to "being ultimately added before the labeling process stops". Second, the "tag set S" can be both interpreted as a *set* or a *sequence*. Third, the setting and motivation of the humans might influence their decisions, depending on the future user of the tag. Fourth, humans will, of course, differ both with regard to expertise (understanding the image) and the kind and level of language used (putting things into words), so that the "by humans" is ill-defined.

Concerning the first point, we will focus on the event of term t being added *immediately next*. This corresponds more closely to the situation we are facing in the ESP game. As for the second point, we use the interpretation of a set without order being present. This kind of approach is imposed by the *sparsity* of the corresponding problem of assessing the probabilities involved. Essentially, a set contains an exponential number of subsets, which we can learn from, but a sequence only contains a linear number of subsequences (or quadratic, if one varies both the starting and end point of the subsequence). Furthermore, we will work with the bigger event of a set already present containing all tags in S, rather than being identical to S. This, as we will see, will avoid having to estimate the significance of a certain tag *not* being present, which is even harder to estimate from a comparatively small set of samples. The third and fourth issues relate to the source of the data used for training. As we mostly train on data obtained from the Google Image Labeler, the interpretation is that a term would be next in the list of off-limits terms, which in turn are derived from human players agreeing on a certain match. However, in the context of tag suggestion, we can also fit a model for a person's tagging past for, say, images on Flickr[8], in which case the interpretation changes. In the first setting, we are mostly dealing with an "average" player of the game, whereas in the latter setting the model would be for a particular user.

As the whole point of the model is, to measure what can be predicted *without* using the image, we will only use the set S and no other information related to the image. In fact, even human players are also often influenced by the set S of off-limits labels. This influence is strongest for images depicting unidentifiable or blurry objects. Here the player uses the clues of the off-limits labels, to "understand" the image, which leads to an automatic reinforcement of the previous interpretation.

[8] `http://www.flickr.com`

How we Model it

The probability we are interested in can be written, using Bayes' formula as follows.

$$\begin{aligned} & P(\text{`}t \text{ is next label'}|\text{`set } T \text{ already present'}) \\ &= P(\text{`set } T \text{ already present'}|\text{`}t \text{ is next label'}) * \\ & \quad P(\text{`}t \text{ is next label'})/P(\text{`set } T \text{ already present'}) \end{aligned}$$

For our applications we assume, as mentioned above, the probability that *some* label will be added next is 1.0. This allows us to drop the denominator from consideration as we know that the expression, summed over all possible terms t, has to yield 1.0. Of course, if we are only interested in a *ranking* of the probabilities for the terms t, then this assumption is not required.

Now the probability of $P(\text{`}t \text{ is next label'})$ can be empirically estimated by the number of occurrences of the tag t among the observed tag sets[9], divided by the total number of observed tags.

So the only relevant and non-trivial probability to estimate is $P(\text{`set } T \text{ already present'}|\text{`}t \text{ is next label'})$. Again, if we had enough training data for every possible set T, we could directly estimate this probability. But given the unavoidable problem of data sparsity in the face of an exponential number of possible sets, we make the following conditional independence assumption.

$$\begin{aligned} & P(\text{`set } T \text{ already present'}|\text{`}t \text{ is next label'}) \\ &= \prod_{t_i \in T} P(\text{`}t_i \text{ is already present'}|\text{`}t \text{ is next label'}) \end{aligned}$$

This is the usual "naive" assumption in a Naive Bayes classifier. For arbitrary labels, it will not hold true. E.g., given that "jaguar" is added next, the probability of "car" being present is clearly *not* independent of the probability of "cat" being present. Generally, this assumption leads to a certain leveling effect of underestimating high probabilities and overestimating low probabilities. E.g., the probability of "stars, stripes" being present given that "flag" is next would be underestimated (assuming that "stars" and "stripes" often occur *together* with "flag"), while the probability that these two terms are present given that "bright" is next would be overestimated (assuming that both "stars" and "stripes" often occur with "bright", but rarely both at the same time). Still, this assumption is a good compromise between being close to reality and not requiring an unrealistic amount of

[9] This number will be equal to the number of tag sets containing t, as no tag set contains any label twice.

training data. Ultimately, its use can be empirically justified by the performance of the robot playing the ESP Game.

The individual probabilities $P(\text{`}t_i \text{ is already present'}|\text{`}t \text{ is next label'})$ are now estimated by dividing the number of tag sets, or rather sequences, in which the label t_i occurs *before* t by the total number of tag sets containing t. Note that these estimates are the maximum likelihood estimators. Although the conditional independence assumption has vastly reduced the problem of sparsity, there is still the risk of prematurely estimating a probability $P(\text{`}t_i \text{ is already present'}|\text{`}t \text{ is next label'})$ to be zero, due to lack of observed tag sequences. Therefore, the probability P is replaced by a *smoothed* variant $\tilde{P}$ using a mixture model based on a combination with a simple unigram model.

$$\begin{aligned}&\tilde{P}(\text{`}t_i \text{ is already present'}|\text{`t is next label'})\\ &= (1-\lambda)P(\text{`}t_i \text{ is already present'}|\text{`t is next label'}) + \\ &\quad \lambda P(\text{`}t_i \text{ is already present'})\end{aligned}$$

The $P(\text{`}t_i \text{ is already present'})$ is estimated as the number of observed tag sets containing t_i divided by the total number of tag sets. Note that in the mixture model above a $\lambda = 1.0$ corresponds to an assumption of *full independence* between the terms in a set. Also note that for $0 < \lambda \leq 1$ and for any previously seen tag t_i this probability estimate will never give zero. In all of our experiments, we used a value of $\lambda = 0.85$. This value was chosen using a validation set of images, not part of the test set used to estimate the predictive performance of the model. With this smoothing, we then obtain the following probabilistic model.

$$\begin{aligned}&P(\text{`}t \text{ is next label'}|\text{`set } T \text{ already present'})\\ &= \prod_{t_i \in T} \tilde{P}(\text{`}t_i \text{ is already present'}|\text{`}t \text{ is next label'}) \quad (1)\\ &\quad * P(\text{`}t \text{ is next label'})/C\end{aligned}$$

Here, $C = \sum_t \prod_{t_i \in T} P(\text{`}t_i \text{ is already present'}|\text{`}t \text{ is next label'}) * P(\text{`}t \text{ is next label'})$, as the model assumes that *some* label t will be applied next. Again, when it comes to *ranking* of the labels t, the constant C is irrelevant. It is only needed if we want to interpret the result as a probability. Also note that in settings where T is empty, we use the probability $P(\text{`}t \text{ is next label'})$.

Ultimately, the actual model used is not crucial as our main objective was simply to show that the labels on the ESP game are predictable using *only* the off-limits terms. An improvement in future models would only make this claim *more* true and it would lead to even *lower* estimates of the entropy. Similarly, the predictive power should improve further with more training data, which would also reduce the need for smoothing and hence result in a lower value of λ.

A ROBOT PLAYING THE ESP GAME

To show that in the current implementation of the ESP Game, tags added are predictable from the tags already present, we used the model presented in the previous section to implement a robot which plays the game *without extracting any information from the image itself*.

Implementation Details

We used the Watir[10] library for Ruby[11] to have a scripting interface to the Internet Explorer. Using this library we could get the current status of the game and automatically add new labels or pass as appropriate. The input rate, that is the number of labels entered within the time limit of 120 seconds, was throttled to play more human-like and not add, say, 100 tags in one second. Therefore we always waited a few seconds before adding any tag for a new image and, similarly, we waited a couple of seconds before adding an additional tag or "reacting" to a user's request to pass. Averaged over all the $2,600$ games played, our robot suggested around 4.3 labels per image, before (i) finding an agreement, (ii) passing or (iii) running out of time. This corresponds to an input rate of 4.4 seconds per label entered, compared to 5.1 seconds for the human players. We set an upper limit of 10 to the number of labels added to any image, at which point we would ask the partner to pass. Figure 1 shows a screen-shot of the robot playing the game. In order to avoid tracking of our robot and to limit the possibility of "personalized" images during the games, we waited between 5 and 10 minutes between two consecutive games and we removed any identifying cookies. At the end of the game, all suggestions made by the partner are revealed and this information is then also recorded by our robot.

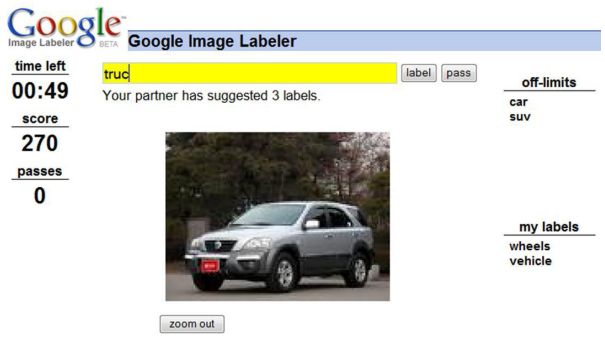

Figure 1. A screen-shot of our robot playing the ESP game. Only using the off-limits terms "car" and "suv", it has produced the list "wheels, vehicle" and is entering "truck". This will then lead to a match for 120 points.

The Cold Start

Before we had obtained a sufficient number of tag sets from the ESP game to train a good model, we used the "global" tag suggestion scheme from [6] in combination with tagging data obtained from Flickr. This way, we avoided obtaining the initial data set by human play. As this initial scheme did not perform as well (for the task of playing the ESP game) as our later scheme based on the model corresponding to

[10] http://wtr.rubyforge.org/
[11] http://www.ruby-lang.org/

Equation 1 we do not report any performance numbers for this first phase.

Different Playing Strategies

Maximizing the number of matches. Without knowing the scores awarded by the Image Labeler Game to different labels, the only reasonable strategy seems to be to try to maximize the total number of agreements with the partner while playing. If we were to add only a *single tag*, then it would be optimal to add the label for which our model output the highest probability estimate. However, as we have the possibility to add multiple tags, until we either find a match with our playing partner or until we pass, at least two different strategies are possible. One is to simply rank the terms by the probability of being added next, as output by the model. This ranking is then *not* changed while subsequent terms are added. This strategy we call *MaxMatches*. If one were to also estimate the significance of a certain label *not* being present, then one could update this list at each step, conditioned on the assumption that the labels suggested so far to the partner will *not* be selected next. E.g., if for the off-limits term "jaguar" the label "cat" had already been suggested, such an approach would favor suggesting "car" next, rather than reinforcing the "cat" interpretation by suggesting a tag such as "zoo". However, we only experimented with *MaxMatches* as synonyms, which would be avoided by this approach, are indeed desirable in our setting.

Maximizing the number of points. Once we have played the game successfully for some time, we can learn the number of points awarded, especially for the common words, such as colors or general terms such as "woman" or "man". For previously unmatched terms, a prior estimate of 140 is fairly accurate in most cases, as (i) the maximum number of points awarded is 150, (ii) if the term was frequent (and would give less than, say, 100 points), we would have probably had at least one match using this term, and (iii) by far the most terms ever agreed on gave 140 points. See Table 2 for details on the awarded points in relation to the number of matches. Using this knowledge about the point value of a label, we can weight the probability estimates by this number of points. Thus, we would prefer a high-scoring term over a low-scoring term, even if the latter is slightly more likely to be agreed upon than the first. This strategy we call *MaxPoints*.

Results

Table 3 gives a summary of the robot's performance for both playing strategies. Most notably it achieves to find a match for roughly 80% of images, which have at least one off-limits label. For images, where a match was found, the match was usually between 2nd and 3rd in the list of suggestions made by the human. This indicates that the robot does indeed manage to "read the human's mind" and does not match on unlikely terms. The differences between *MaxMatches* and *MaxPoints* are rather small. On a per-match basis *MaxPoints* does indeed score higher, but as it takes more labels to find a match, its overall score in the limited time is worse. The fact that the performance difference is so small is due to the fact that in 95% of cases, the two methods agree for the highest ranking label and, on average, they agree on 4.8 out of the first 5 labels. This high degree of similarity is in turn explained by a rather mild (negative) correlation between a high Google Image Labeler score and an overall probability/frequency of only $-.28$ for the 50 most frequent terms. If this correlation was stronger, then the two strategies would be expected to differ significantly more.

Points awarded	Distinct words	Matches	Example
150	1	1	a joy
140	225	360	actress
130	42	68	fight
120	26	86	cloud
110	17	26	kid
100	16	39	music
90	6	13	plant
80	7	50	guy
70	1	1	beach
60	4	10	ocean
50	29	270	black

Table 2. Distribution of points when the *MaxMatches* strategy is employed. The points awarded for a match are independent of the strategy, but the distribution can change.

	Strategy	
	MaxMatches	*MaxPoints*
Number of		
- games	205	208
- images	1,335	1,238
- images^{o12}	1,105	1,008
%-age w match		
- all images	69%	67%
- only images^{o}	81%	78%
%-age tags matched	17%	15%
Average score		
- per game	467	437
- per image	72	73
- per image^{o}	85	88
- per match	104	109
Av. labels entered		
- per image	4.1	4.5
- per game	26.7	27.0
Agreement index		
- mean	2.6	2.8
- median	2.0	2.0

Table 3. The percentage of tags matched refers to the number of tags entered by the robot, leading to a match, divided by the total number of tags entered by the robot. By "agreement index" we mean the index, starting at 1 in the partner's list of suggestions, for which we found an agreement. A low agreement index indicates that we did not rely on the partner to enter dozens of tags. Only images with a match are taken into account for this. Note that the average score per match is (slightly) higher for the *MaxPoints* strategy.

Is our "Robot" Playing with Other "Robots"?

There is the possibility that the other player is actually *not* a human. However, we claim that this is unlikely.

First, if prerecorded game play of an actual human player is used, as already proposed in [15], the labels still come

from humans and our line or reasoning is still valid. Second, there does indeed seem to be some human input, as the labels entered by the other player frequently contain misspellings such as "cemetary" or "limosine", often immediately followed by the correct spelling. Third, at least some of the games seem to be clearly played in a live setting as, when our robot fails to find "obvious" matches, in particular for images without any off-limits terms, the other player often resorts to entering various insults, including politically extremely incorrect terms, which are unlikely to be entered by any robot.

INFORMATION CONTENT OF TAG SETS

We have repeatedly stated that the labels entered by humans in the current implementation of the ESP Game are highly predictable, given only the off-limits terms. This in turn then leads to fairly general set of tags. In this section, we will quantify this claim by looking at the *entropy* of the labeling game.

We want to measure, how much information was added at each step, as the list of off-limits terms grew from empty to (up to) 5 terms. For each label position, we measure the information defined as $-\log_2 p(t)$, where $p(t)$ is the probability of tag t being added next as predicted by Equation 1, which uses the previously added labels. The unit of this information quantity is "bits", as the information measures, how many bits would be required to encode the particular event. The entropy of a random source is then defined as the expected amount of information. More correctly, we are dealing with the *empirical entropy*, which is simply the average information. Note that the numerical estimates of the information and the entropy would change for a different probabilistic model.

One potential problem is that the term to be added next might not be in the list of previously seen labels. Hence, it would be assigned a probability of 0.0. To avoid this problem, we must adapt the empirical estimate of $P(\text{`}t \text{ is next label'})$ to take into account the possibility that t might fall outside the known vocabulary. To achieve this, we down-weight the total probability mass assigned to terms in the vocabulary (which so far summed to 1.0). This way, we have some probability mass "left" to accommodate the possibility of unknown tags appearing. Concretely, we allowed an unseen label to be generated next with a probability equal to the probability of the rarest tag being next (roughly $19/1,000,000$). This is an *underestimate*, as for the given vocabulary size the probability that the next occurrence of a label will be an unknown label is empirically observed to be around 1%. Thus, all of our entropy estimates are in fact *overestimates*.[13]

Table 4 shows how later labels among the off-limits terms become more and more predictable. If one assumes, that the order of the off-limits terms corresponds to the order in which they were added by players, then one can conclude that there is an effect of "diminishing returns", where later terms add less and less information to the set already present.

[13]It is not immediately obvious that assigning lower probabilities than the true probabilities to rare events, and therefore higher probabilities to more frequent events, does indeed give an overestimate of the entropy, but this can be easily shown.

Av. information per position of label in tag set				
1	2	3	4	5
9.2 (9.2)	8.5 (7.8)	8.0 (6.8)	7.7 (5.9)	7.5 (4.6)

Table 4. A model was trained on $13,000$ tag sets. The empirical entropy, i.e. average information ($-\log_2(p)$), is then reported for each label over a test set of $1,546$ tag sets. As the previous labels of a set are used to predict the next label, the labels become more and more predictable. The numbers in parentheses are the results when the model is tested on the training set. An equidistribution over all seen $4,958$ words would correspond to 12.3 bits. If terms were independent of the previous labels and followed a unigram model, the information at any position would be 9.3 bits.

To see if humans work their way towards less and less predictable terms, as they think of more labels to add, we did the following. We looked at the information gain of the labels suggested by humans, with respect to the off-limits terms, as a function of their position among the tag sequence. Table 5 shows that the information does indeed go up for later labels.

Av. information per position of human suggestions				
1	2	3	4	5^+
8.7 (7.8)	9.4 (8.5)	10.0 (9.1)	10.6 (9.7)	11.7 (10.7)

Table 5. The average information, when all the off-limits terms are used to predict the next label, goes up with every additionally suggested label. This indicates that (i) a player has to think of less and less obvious tags to suggest and that (ii) the notion of "obvious" is indeed correlated with the notion of "high probability" in our model. The numbers in parentheses refer to the setting where only images with at least one off-limits term are considered.

IMPROVED SCORE ASSIGNMENTS FOR THE ESP GAME

The main mechanism to steer the players, from the game designer's point of view, is the assignment of points to tags. A valid question is, what should the game designer steer the players towards? Our answer is "towards high entropy tags".

First, note that the actual model used is irrelevant for our argumentation, and does not need to be accurate in any sense. It only has to capture the system's current beliefs about which tag will be added next, that is, what it could have done *without* any human intelligence. Second, for the application of image search, the number of relevant images returned for a very precise query ("steyr car 1920") is much more of a problem than precision for a more general query ("cute dog"). Third, let us remind the reader that, with respect to a given model, the mathematical notion of *information* ($-\log_2(p)$) quantifies the usual notion in terms of the number of bits needed to encode a particular label (see the previous section). Thus, if we assume that the model does indeed assign higher probabilities to more general terms, then the information is a natural measure of how much new knowledge the user actually added to the system. Anecdotal evidence that labels with a higher $-\log_2(p)$ are indeed more informative is given in Table 6.

Modeling the Game

Although the ESP Game certainly is a "game" with players, who make moves (i.e. add suggestions) and who get certain payoffs depending on the other player's moves, a simple formulation of the ESP Game in the framework of traditional game theory does not lead to any meaningful result. The reason for this is that, if players are assumed to play rationally with the objective to maximize their points for a match, and if the list of possible moves (i.e. labels to add) and the corresponding payoffs are known to them, they will always add the label with the highest payoff. Even if there are ties, adding the lexicographically smallest, highest scoring label will (trivially) be a Nash Equilibrium[14] (just as any other strategy leading to a match). Note that game theory traditionally gives more insights (and is more applied) in *non-cooperative* settings.

To avoid these problems, we assume that players play rationally with respect to certain probabilistic assumptions. Concretely, we assume that from player A's perspective, who can be either of the two players, player B plays *randomly*. However, player A has certain beliefs about the probability that a certain term will be used by player B for a given image. This model comes close to how the game is perceived by players as, arguably, they either consciously or unconsciously judge the probability that the other player will add a certain label. Also note that in practice the set of words, deemed to have a non-zero probability of being added by the other player, would be fairly small and not contain more than, say, 20 different labels for any image.

Given this probabilistic modeling approach, how would a rational player make her move? Even for a fixed scoring scheme, known to both players, there are still other factors which would need to be taken into account.

- The probability that player B will add a label.
- The order in which labels are added by player B.
- The time to type a label for player A.
- The number of labels suggested by player A.
- The time to type a label for player B.
- The number of labels suggested by player B.

Completely neglecting the time factor would lead to a simple strategy: add *all* possible terms, ranked in descending order of points for a match. Assuming that the order is taken into account in the case of multiple matches, this gives the highest possible score. Clearly, this oversimplification is neither close to reality nor does it give any insights. However, taking all time-related factors into account leads to a model, which requires several additional parameters and is no longer tractable.

For our analysis, we therefore focus on the *one-shot* version of the game, where both players are allowed to make a *single* suggestion and there is no time limit involved. Of course, players are not allowed to communicate in any way. Each player knows the scoring scheme used by the system or, at least, she knows the scores awarded to labels which she deems could be added by the partner with a non-zero probability. However, as mentioned above, each player assumes that the other player plays *randomly*.[15]

[14] http://en.wikipedia.org/wiki/Nash_equilibrium

0th Scoring Scheme: The Original Proposal

In the original description of the game [15] a *unit* scoring scheme was proposed, assigning the same number of points to any match. This scheme is still used on http://www.gwap.com/gwap/gamesPreview/espgame. It is easy to see that such a scheme results in people adding the terms with the *lowest* information all the time. Google's current scheme tries to rectify this, by assigning between 50 and 150 points according to a notion of "descriptiveness", but (i) it does not go far enough and (ii) is context independent, so that even adding "blue" as the first tag to something related to "blue note records" would only give a player 50 points for the globally frequent tag "blue", although it adds quite a lot of information in the context of "note records". The current scheme does not seem to be related to the notion of "information added", as even the common and predictable label "sexy" is awarded 140 points.

1st Scoring Scheme: Learning the Distribution

Player A wants to maximize (what she believes to be) the expected score, namely:

$$P(\text{'Player B chooses } t\text{' [according to A's beliefs]}) * (\text{'score awarded for } t \text{ given the current tag set'})$$

We are not assuming that the player is actually correct in her beliefs. Suppose now, from the system's point of view, that $P(\text{'Player B chooses } t\text{'})$ is estimated by the current set of beliefs of the model. Then, as a reasonable scoring system, we could set:

$$\text{Score of } t = 1/P(\text{'Player B chooses } t\text{'}) \quad (2)$$

Here, this probability refers to the estimate according to the current model of the system. Given this scoring, a rational player would choose the tag with the biggest ratio of $P(\text{'Player B chooses } t \text{ [according to A's beliefs]})/P(\text{'Player B chooses } t\text{' [according to the beliefs of the system]})$.

An interesting observation about this is the following: suppose the system's model is regularly updated in light of the tags suggested (but not necessarily agreed on) by the players. Then, if we assume that the rational players do not change

[15] Again, if we assumed that the scores are (i) known to both players, (ii) that both players know that the other players knows the scores, (iii) that both players play rationally, and (iv) that they assume that the other player plays rationally, then they would both always add the highest scoring label.

their beliefs about the distribution of the other player, and if we assume that the player's assumptions only depend on the same information available to the system (namely, *only* the already given tags and not the object itself), then the system's model would converge to the players' beliefs, as they will always play labels, which are undervalued by the system and which, as the system constantly updates its beliefs, will then be assigned a higher probability mass.

Note that even if the assumption that the actual image is indeed irrelevant given the present tags were true, this property of convergence is still not desirable. After all the system's goal is *not* to learn the "normal" distribution (which contains many colors and synonyms for "breasts"), but we want to motivate the players to move away from this distribution (in order to maximize the entropy of the tag sets).

In practice, the probability beliefs of the players will of course *not* only depend on the offlimits terms. This is most obvious for the very first label, where any probabilistic model can do no better than using a context independent prior distribution, which will deviate a lot from the labels considered plausible by a player. So generally this scoring system promotes labels which are "obvious" given the image, but not obvious given only the other labels. E.g., given only the label "jaguar" both "car" and "cat" might have a probability of $1/2$ of being added next. But to somebody looking at the picture, one of these probabilities will be zero, whereas the other one is high.

Still, this scoring does not motivate people to necessarily add high entropy tags. E.g., when already several labels have been added, the system's model is expected to agree well with the player's beliefs and *any* tag then gives the same expected score.

2nd Scoring Scheme: Rewarding a high information gain

Whereas the scoring scheme in Equation 2 motivates players to tell us directly, where the system's beliefs are most incorrect, it does not motivate them directly to add informative tags. Let us therefore consider the following scoring variant.

$$\text{Score of } t = -\log_2(P(\text{`Player B chooses } t\text{'}))/P(\text{`Player B chooses } t\text{'}) \quad (3)$$

Again, the probability estimates are with respect to the beliefs of the system. If for a given image the system was correct about its beliefs (compared to the player's beliefs), a rational player would add the most information. That is, she will add the tag which is *least* expected by the system, but which still has a non-zero probability of being added by the other player (according to her beliefs).

If the two probability estimates differ (c.f. the "jaguar" example above), two objectives are mingled. First, adding a tag which adds a lot of information and, second, teaching the system about the current beliefs by adding a label which is underestimated by the system. One important thing to note about this scoring method is that if the player simply *assumes* the system estimates the other person's probability to add a given tag accurately, then she can be convinced that it is in her own interest to add the tag, which adds the most information and which her partner still might possibly add as well. If the system really *only* cares about adding tags with the highest information gain, then the following scoring variant is best.

$$\text{Score of } t = [-\log_2(P(\text{`Player B chooses } t\text{'}))]^k \quad (4)$$

For a large exponent k, it is then optimal for the player to add the tag with the highest information gain, given that the (estimated) probability of her partner also adding it is non-zero. The large exponent simply ensures that the actual probability of a match, as long as it is non-zero, does not change the order of preference.

Evaluation of Scoring Schemes

Although it would be undeniably preferable, to evaluate the qualitative differences between scoring variants in a user experiment, we consider this out-of-scope and limit ourselves to anecdotal evidence. Table 6 shows the given off-limits terms, the order of the suggested terms used by a human (which was recorded during plays with our robot), the ranking of the terms according to points awarded by the Google Image Labeler and the ranking according to Equation 3 for five images[16]. Recall that a rational human player would weight the awarded scores by the probability with which she believes her partner to add this label. If the player's estimates of the probabilities are similar to our model, then the ranking of the last column will remain unchanged.

For both of the first two images, the label entered first by the human does not add any new information ("kitten" $\Rightarrow$ "cat" and "man" $\Rightarrow$ "guy"). Equation 3 rectifies this. Similarly, the least predictable terms for the other images are "baby" (for a baby penguin), "cross" (for a cross above a burning ground) and "castle/building" (for a monastery on an isolated cliff), and all of these are ranked highest in the entropy-based ranking. Other terms suggested by the human, which simply reinforce already present terms, are low in the ranking.

THE ESP GAME IN PRACTICE

Transparency and Effect of Scoring Schemes

[16]Image 1: http://www.chrisspagani.com/cartoons/hammie-2s.jpg, Image 2: http://a776.ac-images.myspacecdn.com/images01/40/s_47dac64ce46da910b5b4004bf0646e0f.jpg, Image 3: http://animals.nationalgeographic.com/staticfiles/NGS/Shared/StaticFiles/animals/images/primary/emperor-penguin-baby.jpg, Image 4: http://dancingokra.com/Earlywork/Art/Armageddon.jpg, Image 5: http://www.leopalmerphotography.co.uk/ta%20meteora%20monastiria.jpg

	Off-limits	Scoring order		
		Human	Google	Eqn. 3
Image 1	cartoon ears kitten kitty orange	cat drawing	cat drawing	drawing cat
Image 2	drawing hat man sketch glasses	guy art	guy art	art guy
Image 3	white wings bird cold penguin	snow baby ice	ice baby snow	baby snow ice
Image 4	fire painting picture smoke art	burn burning cross flames	flames burning* burn* cross	cross burn burning flames
Image 5	mountain mountains	green house rock cliff painting	cliff rocks castle* rock building	castle building rock house cliff

Table 6. Some examples to illustrate the benefit of an entropy-based scoring scheme. A * indicates that the score is tied with the score for the word above, as a small set of discrete scores is used in the Google Image Labeler. The lists for the last example are truncated.

In practice, it is questionable, (i) if typical players appreciate a more involved scoring scheme and (ii) if the scoring scheme has an effect on their behavior at all. Of course, even in the current system the scoring scheme is not transparent and players have no way of knowing, why the system awarded, say, 70 rather than 140 points for a certain match. With the systems above the idea is that players could (and should) be told in detail about the scoring mechanism. E.g., every time they enter a suggestion the system should immediately show its score, possibly along with the probability estimate. However, many players might still prefer to go for many, easy matches, as finding a match is simply emotionally rewarding, even if this leads to a lower overall score.

Motivating Players

The more advanced scoring systems above, in particular Equation 3, have a nice selling point. Players could be (correctly) told that the goal of the game is to outwit the machine. E.g., if two players find an obvious match, they would (i) get fewer points and (ii) a smug message could be displayed "Haha! I saw that coming!". On the other hand, if they agree on an informative term, the system could say "Oh, you caught me by surprise!", and the players would be awarded more points.

Educating Players via Robots

Another interesting aspect to consider is, how much can be gained if the system deliberately uses robots and people would not necessarily play with a human. Rather than having someone wait for 10 minutes, when there are not enough players in the system, a player could then be given images, which are already labeled (though the labels are hidden), as proposed in [15]. Similarly, we could use such an approach to *teach* players not to use obvious combinations. Over time, they might simply learn (from the robot players, which they assume to be humans) that adding "lady" to an image already labeled as "girl", "woman" and "model" will *not* lead to a match at all. The same could possibly be achieved if the system simply (secretly) ignores such obvious labels, even if entered by both human players.

Timing Mechanism

Rather than via the scoring scheme, there are also other ways of enforcing more informative tags. E.g., terms could come with a certain time limit, before they are *activated*. That is, an informative term such as "frigate bird" would still lead to an immediate match, if entered by both players, but it takes, say, 10 seconds before the term "black" becomes active and can lead to a match. This would cancel the "emotional reward" of a quick match for obvious labels.

Hiding Offlimits Terms

Players could be encouraged to aim for non-obvious matches by *hiding* the offlimits terms. They would then only be told that there are, say, four hidden taboo words from previous rounds of the game. If they agree on such an unknown taboo term, the round would immediately be over and they get zero points. If they agree on a non-taboo term, they would be awarded a unit score, independent of the match. In such a setting, both players would probably start with less obvious labels, as they are less likely to be already present, and then work their way up to more predictable ones.

Linking Experts

With a scoring system as above (c.f. "Improved Score Assignments for the ESP Game"), tag sets such as "car, automobile, black, wheels, auto" could be replaced by more informative ones such as "black bmw, station waggon, silver wheel covers, parked car". However, there is still the problem that if one player does not know, say, the name of a particular person depicted, then this image can only be labeled with general tags such "woman", "man" or "person". This problem could be partly solved if also matches between *different* instances of a game are used. Then two experts would not have to play together, but it would suffice if both of them, at different times, enter the label "horst köhler"[17] for a particular image. However, (i) this makes the system more vulnerable to spam, (ii) experts are less likely to enter such terms to begin with, if they have to assume they are paired with a less knowledgeable person, and (iii) this would make the scoring either unfair (if experts are not rewarded) or asymmetric (if one player gets awarded points but the other does not). Therefore, it is preferable (i) if domain experts are paired and (ii) if they only label images

[17] The current president of Germany.

related to a certain topic.

Identifying Experts

The status of an "expert" on a topic could be either indicated by a player herself, as she agrees to, say, only label images related to music bands, or it could be learned by the system. One way to achieve the latter would be to have a quiz where, e.g., images from the Wikipedia are shown to a player and she has to correctly label as many people/countries/bands as possible. As players prove their expertise over time, they could go up in a ranking and only be paired with players of similar expertise. The topic of an image could be detected from the first off-limits terms. So if an image is already labeled as "tennis player", it would then be shown to sports experts in a hope to find more precise labels.

REFERENCES

1. M. Ames and M. Naaman. Why we tag: motivations for annotation in mobile and online media. In *Conference on Human Factors in Computing Systems (CHI'07)*, pages 971–980, 2007.
2. K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3(6):1107–1135, 2003.
3. J. Barr and L. F. Cabrera. Ai gets a brain. *Queue*, 4(4):24–29, 2006. `http://www.mturk.com/`.
4. E. H. Chi and T. Mytkowicz. Understanding the efficiency of social tagging systems using information theory. In *Conference on Hypertext and hypermedia (HT'08)*, pages 81–88, 2008.
5. T. Cover and R. King. A convergent gambling estimate of the entropy of English. *IEEE Transactions on Information Theory*, 24(4):413–421, 1978.
6. N. Garg and I. Weber. Personalized, interactive tag recommendation for flickr. In *Conference on Recommender Systems (RecSys'08)*, pages x–x, 2008.
7. N. Garg and I. Weber. Personalized tag suggestion for flickr. In *International Conference on World Wide Web (WWW'08)*, pages 1063–1064, 2008.
8. J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *International Conference on Research and Development in Information Retrieval (SIGIR'03)*, pages 119–126, 2003.
9. E. Law, L. von Ahn, R. Dannenberg, and M. Crawford. Tagatune: A game for music and sound annotation. *International Conference on Music Information Retrieval (ISMIR'07)*, pages 361–364, 2007.
10. T. Paek, Y.-C. Ju, and C. Meek. People watcher: A game for eliciting human-transcribed data for automated directory assistance. In *Annual Conference of the International Speech Communication Association (INTERSPEECH'07)*, pages 1322–1325, 2007.
11. S. Sen, F. M. Harper, A. LaPitz, and J. Riedl. The quest for quality tags. In *Conference on Supporting Group Work (GROUP'07)*, pages 361–370, 2007.
12. S. Sen, S. K. Lam, A. M. Rashid, D. Cosley, D. Frankowski, J. Osterhouse, F. M. Harper, and J. Riedl. tagging, communities, vocabulary, evolution. In *Conference on Computer Supported Cooperative Work (CSCW'06)*, pages 181–190, 2006.
13. B. Sigurbjörnsson and R. van Zwol. Flickr tag recommendation based on collective knowledge. In *International Conference on World Wide Web (WWW'08)*, pages 327–336, 2008.
14. M. Vojnovic, J. Cruise, D. Gunawardena, and P. Marbach. Ranking and suggesting tags in collaborative tagging applications. Technical Report MSR-TR-2007-06, Microsoft Research, February 2007.
15. L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Conference on Human Factors in Computing Systems (CHI'04)*, pages 319–326, 2004.
16. L. von Ahn, S. Ginosar, M. Kedia, R. Liu, and M. Blum. Improving accessibility of the web with a computer game. In *Conference on Human Factors in Computing Systems (CHI'06)*, pages 79–82, 2006.
17. L. von Ahn, M. Kedia, and M. Blum. Verbosity: a game for collecting common-sense facts. In *Conference on Human Factors in Computing Systems (CHI'06)*, pages 75–78, 2006.
18. L. von Ahn, R. Liu, and M. Blum. Peekaboom: a game for locating objects in images. In *Conference on Human Factors in Computing Systems (CHI'06)*, pages 55–64, 2006.