TagCaptcha: Annotating images with CAPTCHAs

Morrison, Donn Alexander; Marchand-Maillet, Stéphane; Bruno, Eric

TagCaptcha: Annotating images with CAPTCHAs

Donn Morrison
Viper Group
University of Geneva
Geneva, Switzerland
donn.morrison@unige.ch

Stéphane Marchand-Maillet
Viper Group
University of Geneva
Geneva, Switzerland
stephane.marchand-maillet@unige.c

Éric Bruno
Viper Group
University of Geneva
Geneva, Switzerland
eric.bruno@unige.ch

ABSTRACT

We introduce a method of annotating images for use in a retrieval setting by exploiting the need for CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart) online. Our system, called TagCaptcha, presents the user with a number of images that must be correctly labelled in order to pass the test. The images are divided into two subsets: a control or verification set for which annotations are known, and an unknown set for which no verified annotations exist. The verification set is used to control against the tags provided for the unknown set. If the user provides correct verification tags, the tags for the unknown set are promoted. An image with a promoted tag must be validated by other users before it can be classed as annotated and added to the verification set. Given a partially annotated database, the images can be incrementally annotated over time. We report usability results from a small user study as well as sample user tags from the online demonstration system.

1. INTRODUCTION

Image retrieval has long been plagued by limitations on automatic methods in that they cannot reliably extract semantic data from the low-level features. The result is that users must formulate awkward and inefficient queries in terms these systems can understand. Humans, on the other hand, have the ability to quickly and accurately summarise visual data. This dichotomy, named the *semantic gap*, is a fundamental problem in image retrieval.

The ability for users to specify keyword-based queries is more natural than traditional query-by-example (QBE) and other low-level feature-based retrieval systems. However, images must first be associated with corresponding keywords before this approach yields effective results. Research on automatic image annotation has largely focused on methods of learning co-occurrences between words and low-level features [1, 4]. However, these methods often produce less than adequate results and are limited to the vocabulary used to train the models, effectively stagnating keyword diversity.

In this paper, we propose a novel system of collecting image annotations based on the need for human verification on the web. Similar in principle to work by Ahn *et al.*, the idea is to exploit the requirement of users to pass tests [6, 8] in order to incrementally annotate images.

2. RELATED WORK

CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart) have proven to be a very useful tool in the fight against spam and other automated exploits on the Internet. For example, users registering for or using services such as web-based email accounts, online polls and message forums are often required to correctly answer a word-based CAPTCHA to verify that they are human. By validating that every user is human, the extent that these services can be exploited for illegitimate purposes is greatly reduced, since most of these exploits are performed automatically by programs.

CAPTCHAs are designed to be difficult to solve automatically yet require relatively little cognitive effort from humans. By inventing CAPTCHAs that require solving specific problems, such as image annotation, it is possible to use this collaborative cognitive effort where it would be otherwise discarded. Most currently employed CAPTCHA systems rely on users interpreting distorted text displayed in small images. Justification rests on the fact that humans can more easily read the text than automated programs. Recently, however, many deployed word-based CAPTCHAs have been broken by sophisticated machine learning attacks [5, 10].

Ahn *et al.* created system called *reCAPTCHA* that channels the necessary cognitive work associated with human verification into a useful purpose: correcting ambiguous portions of text scanned from books using optical character recognition (OCR) software. The authors justify their word-based CAPTCHA in light of the fact that they use two state-of-the-art OCR programs in an agreement-based approach to scanning text. Words that both OCR programs fail to agree on are used in the CAPTCHA to make use of human cognitive effort. Thus, if the CAPTCHA is broken automatically, state-of-the-art OCR will have advanced beyond that used in reCAPTCHA.

reCAPTCHA works by showing a user two words: a control word and an unknown word. The control word is used to validate that the user is human, as in a standard word-based implementation. If the user is validated as human, the unknown word is also assumed to be valid, pending agreement from other users.

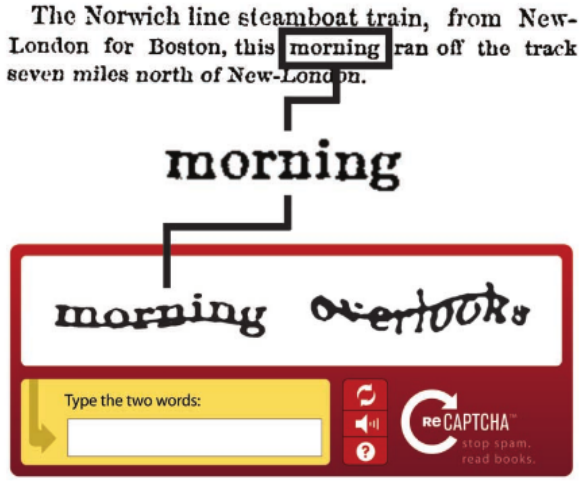

Figure 1: The reCAPTCHA interface. The user must enter both words correctly, however, only one word serves as the control word; the other word is unknown or ambiguous to the system.

Elson *et al.* developed an image-based authentication system called *Asirra* that relies on a large database of images of pets from various animal shelters [3]. In order to pass the CAPTCHA, the user must select all images depicting either cats or dogs from a set of random images from both categories. The system takes advantage of the fact that users can easily differentiate between semantically different visual content, while the problem is difficult for computers.

The image-based *IMAGINATION* CAPTCHA relies on a two stage process to verify that a user is human [2]. The first stage asks the user to locate the centre of an image in a collage of images. In the second stage, the user is asked to choose the keyword that correctly describes another image. Only when the user passes both stages in sequence does the user pass the CAPTCHA. In order to make it difficult for a system to simply index all of the images in the second phase, a series of random distortions are performed on the images to make automatic retrieval less effective.

Another related work is the ESP Game of Ahn *et al.* [7]. In this study, the authors attempt to create incentive for manual image annotation by reformulating the monotonous process as a game: two players are paired randomly and shown the same image. The two players provide guesses for possible tags and when the guesses match they are passed to the next round. The ESP Game demonstrated that given the right incentive, even as simple as a high-scores list, users are eager to participate. Annotations derived from users playing the game are reported to be of similar quality to those derived from using professional annotators, giving more weight to the utility of the approach.

Figure 2: The Microsoft Asirra interface. The user must select all images of a specific species.

3. TAGCAPTCHA

TagCaptcha is an image-based CAPTCHA where users are presented with a set of images and asked to provide one English free-text word that best describes the corresponding image, be it a general concept or a specific object.

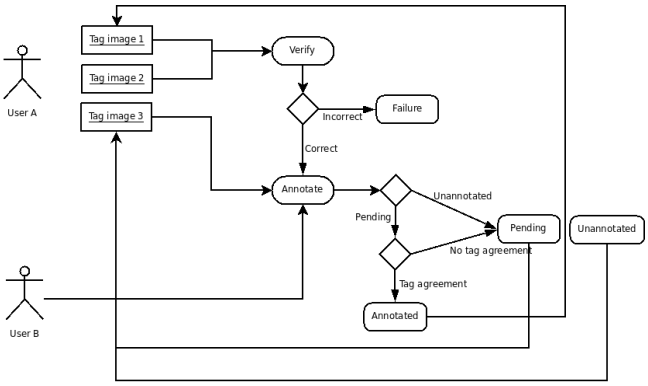

Figure 3: The TagCaptcha flow diagram. When an unknown image (image 3) is agreed upon by two users, the image is promoted to the verification set ("Annotated").

Similar to the reCAPTCHA system [8], a subset of the displayed images are selected from the verification set and have a set of known tags. The remaining subset comprises images for which the annotations are either unknown or pending approval from other users. In this way, the verification images are used to validate whether or not the user is human, and the tag guesses for the unknown images are used to annotate these images. A newly tagged image with no previous tags is promoted to a pending set until it can be verified by another user. Once a particular image has acquired two similar guesses from different users, it is promoted to the verifica-

tion set where it is used to control against new annotations. A flow diagram is shown in Figure 3.

The justification for an image-based CAPTCHA is evident given the semantic gap problem in computer vision (deriving semantic information from the low-level features of images): if TagCaptcha can be solved automatically, then the state-of-the-art will have been improved and the semantic gap will have been narrowed.

3.1 WordNet soft match

Due to the inherent subjectivity involved in describing images, we introduce a two-step matching strategy to reduce rejection of otherwise appropriate image tags. First, the image tags and user supplied guess are stemmed and compared in their base form. If these do not match, the words are compared in WordNet using the similarity measure developed by Wu & Palmer (WUP distance) [9]. WUP distance is defined as $D_{wup} = \frac{2*depth(lcs)}{depth(s1)+depth(s2)}$ where lcs is the least common subsumer of words $s1$ and $s2$ and $D_{wup} \in \{0, 1\}$. As a side effect, many proper nouns (and any other words that do not exist in WordNet) and spelling mistakes are excluded. As a result of soft matching, semantically close image tag guesses are accepted. For example, many users may not be familiar with mammal taxonomy and may incorrectly tag an image of a wolf as a dog. Using the WUP distance measure, wolf and dog have a similarity of 0.93, and under this soft matching technique the guess could be accepted as valid, depending on the value of the soft match threshold t_{wup}.

The selection of the value of the soft match threshold is a trade-off between usability of the system (i.e. how semantically different a valid guess can be) and vulnerability to automated attacks. Constraining the threshold to only match exact keywords ($t_{wup} = 1$) limits automated attacks at the cost usability. We can see the effect of varying the soft match threshold in Figure 4, where the probability of a successful guess was measured on the image vocabulary used in the demonstration system (see below).

3.2 Resistance to automated attacks

TagCaptcha's resistance against attacks is a function of the threshold parameter t_{wup}, the size and distribution of the verification image vocabulary, the size and distribution of the image database, and the number of verification images displayed to the user. Previous studies propose that a reasonable random guess success rate for a CAPTCHA should be lower than 1×10^{-4} [3]. As a rough estimate, the demonstration implementation of TagCaptcha allows the probability of a single valid random guess to be $P(g_v) = 0.095$ at $t_{wup} = 0.50$. With two verification images $P(G_v) = P(g_v)^2 = 9 \times 10^{-3}$; somewhat higher than 1×10^{-4}. However, by setting $t_{wup} = 0.76$, we can achieve $P(G_v) \leq 1 \times 10^{-4}$ for two verification images, albeit at the cost of usability. $P(G_v)$ can be further reduced by adding more verification images, although a separate user study should investigate this effect.

From an automated attack perspective, as discussed by Elson *et al.* [3], the major deficiencies of image-based CAPTCHAs are both the public availability of the image dataset and the vulnerability to statistical attack by sampling enough

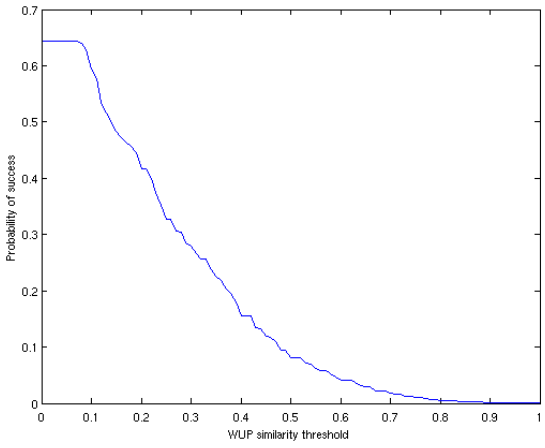

Figure 4: The effect of varying the threshold t_{wup} on the WUP WordNet distance measure versus the probability of a valid guess for one verification image in the database. At $t_{wup} = 0.50$, the probability of guessing a tag correctly is 0.095 for one image and decreases as we increase t_{wup}.

tests to learn a significant portion of the image database. As our system aims to annotate images for retrieval, we assume that our image database would be publicly available. In order to protect against attacks which learn the images in the database, we could employ a strategy similar to that in the IMAGINATION CAPTCHA where images are distorted enough such that the semantic content is left intact but the low-level features are augmented sufficiently to reduce the effect of retrieval algorithms [2]. For the purposes of the demonstration system, however, this has not been introduced.

Due to the fact that the image database in TagCaptcha is currently static, an attack could be constructed on the basis of the keyword distribution of the verification images. Based on successfully manually solving several hundred TagCaptcha tests, a malicious script could be directed to attack TagCaptcha with the most commonly accepted keywords, yielding a statistically higher successful attack rate.

In addition to the use of random image distortions, we are currently investigating ways to curb these vulnerabilities.

3.3 Demonstration system

The demonstration version of TagCaptcha[1] uses a database of 8647 images from the Corel collection spanning 87 image categories each containing approximately 100 images. Some example categories include: *nature textures, tigers, flowers closeup, reflective effects, everyday objects, cheetahs leopards and jaguars, tropical sea life, fabulous fruit, apes, etc.* Because some categories contain images which are difficult to describe, for example, textures, we manually removed these images to improve usability. The image collection has a vocabulary of 2,353 unique words. Each image has approximately 4 keywords describing the depicted visual content,

[1]Demo URL: `http://dolphin.unige.ch/tagcaptcha/`

and therefore a subset of the images can be used to seed the TagCaptcha test as the initial verification set. This set was created by randomly selecting 10% of the images and keeping their original tags.

For soft matching using WordNet, the minimum WUP distance threshold for a valid match was emphirically set to $t_{wup} = 0.50$. Upon entering valid tags and clicking the "Guess" button, the user will be notified of the success or failure and returned to the initial TagCaptcha page to perform another test. We left a debugging option available so users of the demonstration can determine why their guesses pass or fail.

Figure 5 shows the TagCaptcha demonstration interface. Because image thumbnails are used to display the images, the user has the option of moving the mouse cursor over the thumbnail to display a larger version.

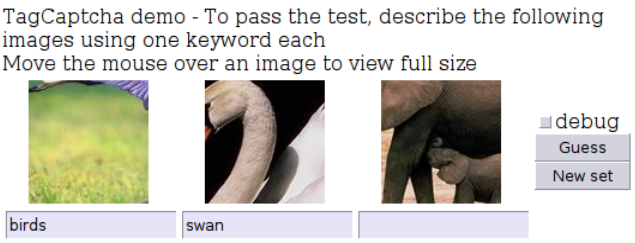

Figure 5: The TagCaptcha demonstration interface showing three randomly chosen images: two verification and one unknown image. The thumbnails represent a small portion of the image; the user can mouse-over each image to view a larger version.

3.4 User study

We conducted a small user study involving 12 participants from our laboratory. The objective of the user study was to obtain a usability estimate. In other words, we wanted to know how difficult it was to pass the test. Each participant was asked to solve 20 instances of the TagCaptcha test. The average success rate was 70%. The result is quite low compared to word-based CAPTCHAs where user success rates are usually above 90%. There are several factors contributing to this result.

First, tagging images is highly subjective and images containing multiple or complex concepts will have lower success rates than those with simple or straightforward concepts. Word-based CAPTCHAs, on the other hand, present a definite and objective test, allowing users to easily provide the correct answer, assuming the text distortion does not hamper readability. Likewise, context and personal experience play a large role. Users being unfamiliar with certain animal species, for example, would perform poorly on images containing animals.

Language difficulties also contribute to low success rates since users may not immediately know the words in English to describe the displayed image. Some users reported having to look up definitions before guessing, and others reported choosing a known higher level concept when they could not immediately identify the correct translation (i.e. labelling an image of a deer with "animal"). An approach to dealing with language problems would be to include a translation engine which would accept guesses in other languages, based on the language setting of the web browser, for example.

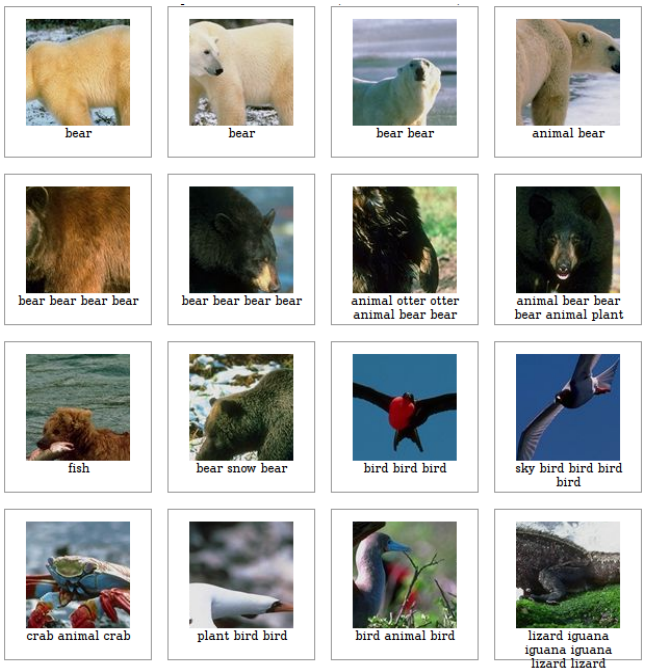

Figure 6: Example user-provided annotations from the TagCaptcha demonstration setup.

Finally, uncorrected spelling errors, which can be related to language difficulties, were reported in some cases. One user reported knowingly misspelling "stalactite" for an image depicting a cave and failed the test. In an attempt to reduce failures due to spelling errors, we have added a spelling helper to the interface that displays a matching list of words as the user types.

3.5 Annotation quality and coverage

Figure 6 shows a random selection of the most recent user-provided tags captured by the TagCaptcha demonstration implementation. Word frequencies are saved by simply appending new valid guesses to the list of annotations for each image. Evidence of mismatched tags can be seen in row 2, column 3, where an image of a bear was mistaken to be that of an otter. This has presumably happened because the first tag was "animal", which is semantically similar to both "bear" and "otter". This highlights the problem of allowing soft matches: while allowing semantically similar tags to improve usability, we may decrease the quality of the image annotations.

Row 2, column 4 shows an example where a user has tagged the image with a concept that does not appear to be the focus of the image. The first five tags are related to the apparent object ("bear", "animal"), whereas the sixth tag relates to a secondary object, "plant", which appears in the foreground of the full sized image.

In general the tags correspond quite well to the images and appear to be of a similar quality to those gleaned from users playing the ESP Game [7]. Although the variety in annotations seems quite low, this can be seen as a positive result given the problem of user subjectivity. An input constraint such as the ESP Game's list of taboo words could be useful in improving tag variety.

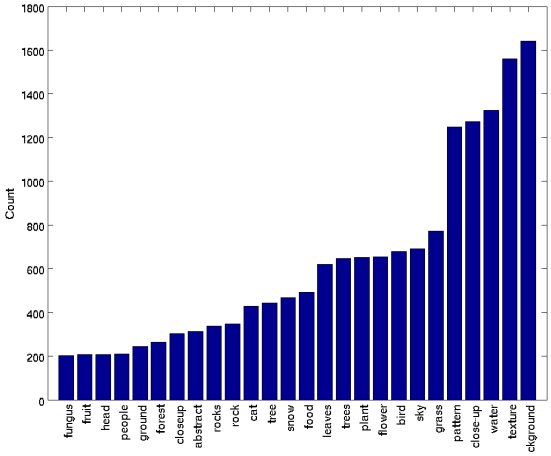

Figure 7: 25 most frequent ground truth tags from the Corel dataset used in the TagCaptcha demonstration.

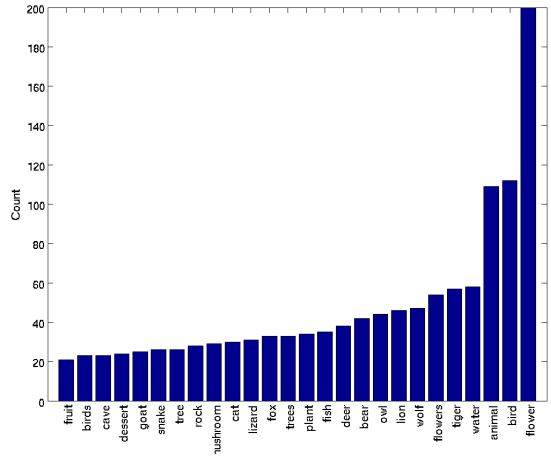

Figure 8: 25 most frequent user-provided tags from the TagCaptcha demonstration.

Figures 7 and 8 show the keyword distributions for the 25 most frequent tags for the original Corel annotations and the user-provided respectively. The most frequent original annotation, "background", does not feature at all in the user-provided set as it is abstract and of limited descriptive use. Prominent user-provided tags such as "flower", "bird", and "water" are also frequent in the original set, showing that there is a tendency for the user-provided tags to model the underlying image data.

4. CONCLUSION

We introduced a novel method of exploiting the need for human verification on the web in order to annotate images for improved keyword-based retrieval. Based on the success and acceptance of reCAPTCHA for correcting OCR errors in scanned text, we believe that TagCaptcha offers a viable means for eliciting diverse and objective image annotations for use in image retrieval applications.

We reported the results of a user evaluation to determine an estimate of usability. It was found that users were able to pass the test 70% of the time. Despite being significantly harder than word-based CAPTCHAs, we see several areas that can contribute to increased performance. Subjectivity (as well as context and personal experience), language, and spelling all contribute the low usability score. Subjectivity is a difficult problem to solve, as it is inherently dependent on each users' personal experience and context. Language and spelling, while less significant, can be accounted for by introducing lexicons for other languages and incorporating a strict spelling guide as users enter guesses.

5. ACKNOWLEDGEMENTS

This research was funded in part by the Swiss National Science Foundataion (SNF) through IM^2 (Interactive Multimedia Information Management) and by EU-FP7-ICT.1.5 NoE PetaMedia.

References

K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. Jordan, *Matching words and pictures*, Machine Learning Research **3** (2003), 1107–1135.

Ritendra Datta, Jia Li, and James Z. Wang, *Imagination: a robust image-based captcha generation system*, MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia (New York, NY, USA), ACM, 2005, pp. 331–334.

Jeremy Elson, John R. Douceur, Jon Howell, and Jared Saul, *Asirra: a CAPTCHA that exploits interest-aligned manual image categorization*, CCS '07: Proceedings of the 14th ACM conference on Computer and communications security (New York, NY, USA), ACM, 2007, pp. 366–374.

Florent Monay and Daniel Gatica-Perez, *Modeling semantic aspects for cross-media image indexing*, IEEE Trans. Pattern Anal. Mach. Intell. **29** (2007), no. 10, 1802–1817.

G. Mori and J. Malik, *Recognizing objects in adversarial clutter: Breaking a visual captcha*, CVPR, vol. 1, 2003, pp. 134–141.

Luis von Ahn, Manuel Blum, and John Langford, *Telling humans and computers apart automatically*, Commun. ACM **47** (2004), no. 2, 56–60.

Luis von Ahn and Laura Dabbish, *Labeling images with a computer game*, CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems (New York, NY, USA), ACM Press, 2004, pp. 319–326.

Luis von Ahn, Benjamin Maurer, Colin Mcmillen, David Abraham, and Manuel Blum, *reCAPTCHA: Human-based character recognition via web security measures*, Science (2008), 1160379+.

Zhibiao Wu, *Verb semantics and lexical selection*, 1994, pp. 133–138.

Jeff Yan and Ahmad S. Ahmad, *Breaking visual captchas with naive pattern recognition algorithms*, Computer Security Applications Conference, 2007. ACSAC 2007. Twenty-Third Annual, 2007, pp. 279–291.