KA-CAPTCHA: An Opportunity for Knowledge Acquisition on the Web

Bruno Norberto da Silva, Ana Cristina Bicharra Garcia
Computer Science Department, Fluminense Federal University
Passo da Patria 256-E
Niteroi, RJ 24210-240 Brazil
{bsilva,bicharra}@ic.uff.br

Abstract

Any Web user is a potential knowledge contributor, but it remains a challenge to make them devote their time contributing to some purpose. In order to align individual with social interests, we selected the CAPTCHA Web resource protection application to embed knowledge elicitation within the users' main task of accessing a Web resource. Consequently, unlike previous knowledge acquisition approaches, no extra effort is expected from users since they are already willing to use a CAPTCHA to perform some particular task. We present an application where we extract pictorial knowledge from Web users, and experiments suggest that our approach enables knowledge acquisition while still satisfying CAPTCHA's security requirements.

Introduction

Many features of the Web are based on knowledge extracted from humans. Evidence for such a claim includes Web search techniques, Wikis (Wikipedia) and recommender systems (Resnick and Varian, 1997). While Wikis build a more explicit knowledge acquisition with a more direct flow of information, some search mechanisms benefit from an implicit collaboration where search engines take advantage of the structure generated by people when they link one Web page to another (Brin and Page, 1998). Recommender systems present a combination of the implicit (Apte et al. 2002) and explicit (Avery et al. 1997) approaches.

Arguably, one aspect that popularized this cooperation between Web users and their applications is the fact that almost every Web user is a potential contributor. Nevertheless some issues still remain, possibly the most compelling one being the necessity to align individual with social interests, since one often finds the task of contributing to some common good to be boring or exhaustive.

One early approach to the problem of large scale knowledge acquisition is the CYC project (Lenat, 1995), where information is to be collected from selected trained experts to construct an ontology of commonsense knowledge. However, using a limited number of contributors might lead to a reduced knowledge base (Chklovski and Gil, 2005).

Subsequent endeavors had untrained volunteers as the source of knowledge to the system (Singh et al. 2002; Chklovski, 2003; Wikipedia). These approaches do not attempt to control the number or nature of knowledge contributors, which hopefully will be numerous and diverse. The Open Mind Common Sense project is an instance of this class, where Web users freely register to the system and perform an activity that covers the knowledge contribution task. For instance, one Open Mind user could face a hypothetical event like *Bob bought some milk* and then be asked to *write up to five things that someone should already know in order to fully understand the event* (OpenMind Commonsense). Hopefully, the system will be able to gather the new information supplied by the user with previous submissions to generate a large repository of commonsense knowledge.

Some differences distinguish efforts like the Open Mind initiative from the early CYC project. While the former has the potential to acquire a larger amount of knowledge by accommodating new contributors as they volunteer from the Web, the latter is able to provide better guarantees against malicious users. However, they all underperform as they fail to extract as much knowledge as people would be able to report. The reader is referred to (Chklovski and Gil, 2005) for a more detailed review of knowledge collection from volunteer contributors.

Recent investigations attempted to overcome this inefficiency by working out the psychological aspect of the interaction. (Rashid et al. 2006) presents an experimental study where humans are encouraged to contribute with more information to an online movie recommendation community by learning the value of their contribution. In (von Ahn, 2005), the author presents the idea of motivating people's participation in a knowledge acquisition effort by making the experience of contributing with some information pleasant. In his work, the problem of generating textual descriptions for images is attacked by designing games that people find very entertainment to play and by playing them they end up labeling images as a

side effect. The ESP game is an example of such a game, and can be described generally as a game where two random players are anonymously paired and both are shown the same image. Each player is then challenged to try to guess what her partner is typing in a short period of time; see (von Ahn, 2005) for a more complete description.
This entertainment approach has proven to be very successful and this idea is now known as an instance of Human Computation, which can be roughly described as the effort of making good use of the human cognitive power when people perform some peripheral task like playing games on the Web (von Ahn, 2005, Chapter 1).
The approach we introduce here can be regarded as an instance of Human Computation. We extend the existing CAPTCHA model (von Ahn, 2005, Chapter 2) used to protect Web resources against malicious agents (Figure 1) to enable knowledge acquisition from its users; i.e. at the same time a CAPTCHA would generate and send a test to some agent to verify if the agent is a human or a computer, this test could be carefully designed to allow (or even force) users to contribute some knowledge with their response.
We are motivated by the idea that since interaction with CAPTCHAs is decidedly present in the Web experience, if we could extract knowledge from users every time they solve a CAPTCHA test then maybe a relevant amount of information could be gathered by computers. We advocate that the CAPTCHA approach has several potential advantages over previous efforts, including the following:

- Unlike the case of knowledge collected from volunteers, the CAPTCHA approach does not rely on altruism to succeed. Our approach might succeed because people need to use the CAPTCHA to fulfill their agenda (i.e. to access some resource or to perform some task on the Web).
- Unlike the case of entertainment, we do not need to redesign our system from time to time in order to keep it attractive to people, as is the case with games. Again, people will not use our CAPTCHA to get distracted, surprised or amused. In fact, they would rather not use a CAPTCHA if they could.
- Because our method's appeal lies in the resource protected by the CAPTCHA and not on the CAPTCHA itself, one could save design and implementation time by applying the very same CAPTCHA mechanism to a number of different services, reaching a wider audience that could contribute with knowledge of a more diverse nature. We believe this is not the case with entertainment.
- Moreover, the effort our approach expects from our users is already being exerted with the current CAPTCHA design, but only in a fruitless way. Current instances of CAPTCHAs have become so difficult for humans (Chellapilla et al. 2005) that it can be viewed as a waist not to recycle this considerable effort into something positive.

Figure 1. A CAPTCHA must distinguish humans from computers.

In the next section, we will briefly review the concept of CAPTCHAs while section 3 introduces our extension to the CAPTCHA model. Section 4 describes an application of our proposal to the domain of image taxonomy and empirical evaluations are presented in Section 5. Finally, Section 6 ends this work with conclusions and a brief discussion.

The CAPTCHA Model

CAPTCHA stands for *Completely Automated Public Turing Test to Tell Computers and Humans Apart* (von Ahn, 2005, Chapter 2). It works by generating tests that humans are expected to easily pass but current computers will find it virtually impossible. A CAPTCHA is useful whenever there is need for protection of some resource against automated attacks, and since most Web resources need some sort of protection CAPTCHA has become very popular.
Examples of successful applications include email account registrations, which cannot be freely accessed due to risk that spammers will be able to spread even more unsolicited email, public database queries, which might be flooded by denial of service attacks, and SMS message sending services.
Among its additional properties, we highlight that a CAPTCHA must not rely on any privacy of data or code to secure its purpose. This means that a computational agent must fail a CAPTCHA test even if it has access to the data and procedure involved in the generation of the test. The challenge then is to build a test that is hard for computers but easy for authentic users.
Arguably, the most popular instance of a CAPTCHA is Gimpy (Figure 1), where the system generates an image containing a random sequence of letters and then challenges an agent to identify which letters are encoded in the image.
Because of the property of public data, the CAPTCHA is forced to introduce a random noise into the image in order to guarantee that computational agents will not be able to recognize its content. This was the case when this CAPTCHA was being developed, but subsequent efforts have overcome this test challenge (Mori and Malik, 2003), turning Gimpy into a broken CAPTCHA. Current implementations of Gimpy try to maintain security primitives by increasing the difficulty of the test, but the state of the art breaking algorithms would require such a difficult test that most humans (if not everyone) would be unable to pass it (Chellapilla et al. 2005).
It is important to stress that the current CAPTCHA test stands merely as an overhead to Web services while it does not aggregate value to the domain, making the cognitive

effort exerted by humans when taking the tests an unproductive experience.

This scenario poses a good opportunity for knowledge acquisition because people are facing hard tests and willingly giving their best shots to solve them. The problem lies in designing a CAPTCHA test that impedes computational agents to access unauthorized resources at the same that elicits some knowledge from users. In the next Section we present an architecture that enables CAPTCHAs to extract knowledge from humans.

Knowledge Acquisition with CAPTCHA (KA-CAPTCHA)

The architecture of current CAPTCHAs contains an agent that generates tests aiming at distinguishing humans from computers. The general CAPTCHA procedure starts by picking up some data from a public repository, adding some noise into the collected data and then generating a test based on these noisy data before sending it to some challenged agent.

This method can be abstracted as if the CAPTCHA agent retrieves some questions from a public database of questions and answers, rewrites the questions in a way to avoid cheating and then generates a test based on the set of modified questions.

If it is the case that there are more questions than what is answered in the current database, one could imagine a CAPTCHA test where an agent collects the usual set of answered questions and, additionally, some questions whose answers are not yet known by the agent. All questions are then mixed together and sent to some challenged agent as one single test. Since the questions have been rewritten, the challenged agent will be unable to distinguish questions whose answers are unknown from the others and so will be forced to answer appropriately to the entire test. When the CAPTCHA receives a response to this test from the challenged agent, it will be able to verify the answers related to the previously answered questions. If they are all correct, then the challenged agent is believed to be a human. The CAPTCHA agent will then consider the answers associated with the remaining questions and possibly extract new knowledge from the interaction.

This metaphor is made concrete by the architecture depicted in Figure 2 (all content in black belongs to the traditional CAPTCHA architecture, while the blue part belongs to our extension). A CAPTCHA agent generates a test by retrieving data from a knowledge base and possibly combining it with data gathered from other sources (here represented by a Hypothesis generator). The CAPTCHA agent then generates a test and sends it to some Challenged agent, who reacts with a response to the test. The CAPTCHA agent then reviews the response and, in case of success, forwards the response to some Information extracting agent who will analyze it looking for new knowledge to feed into the original knowledge base.

The success of the knowledge extraction from the test's response is directly correlated with the design of the CAPTCHA test. Better designed tests should enable better knowledge acquisition at latter stages, but never in detriment of CAPTCHA's original security purpose or proper recognition of legitimate users.

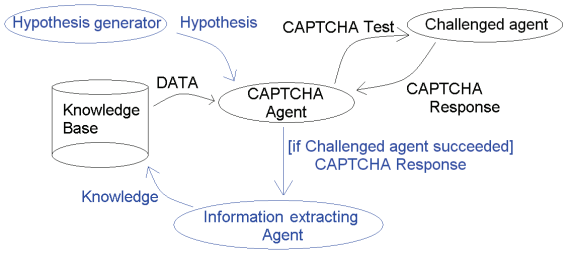


Figure 2. The CAPTCHA architecture, extended to enable knowledge acquisition.

An Application of KA-CAPTCHA to Pictorial Knowledge

In this Section, we present an application of the KA-CAPTCHA architecture. Our goal is to design a CAPTCHA that will help us describe images. This problem is difficult, since there is not a general computer algorithm that, given any image as input, returns a valid textual description of that image. To make things worse, one can note that most images have more than a single valid description and valid descriptions might even be conflicting.

One popular workaround applied by image search engines on the Web is to estimate a valid description to an image from metadata like the image's file name or key words from Web documents linked to the image. Unfortunately, this method sometimes leads to very bad results, demonstrating the need for better alternatives.

This problem has already been attacked using a different form of Human Computation in (von Ahn, 2005, chapter 3). We compare our approach to theirs in Section 6.

Knowledge modeling

Since we want to design a CAPTCHA that collects pictorial knowledge, we assume that a public knowledge base of images and possible descriptions of them is available (possibly the result of a machine learning technique or current image-searching services). This might be regarded as a very modest ontology, with only two classes (namely *Image* and *Label*) and one semantic relation *describes* connecting a label to an image.

In our model, we attach to each *describes* relation two measures. First, there is a *Confidence rank* that indicates how confident the system is that the respective label describes the image (under the view of the system's users). This measure is higher the more users indicate that the relation is indeed correct, and lower the more indications that the relation is incorrect.

As illustrated in Figure 3, we compare each Confidence to two thresholds. If a given Confidence lies beyond a *Certainty threshold*, the system would be convinced that the particular label really describes the content of the image (under the view of its users). If some Confidence

falls below a *Suspicion threshold*, the system would believe that the label does not describe the respective image (again, from the users' point of view). Whenever a Confidence lies between these two thresholds, this indicates that further investigation is needed in order to determine the nature of the relation between the label and the image.

Figure 3. The Confidence rank is a measure of how confident the system is about a given label-image relation.

The second measure we add to each label-image relation is a Support rank and it measures how many users have contributed with an indication about the pertinence of the label-image association. The Support can be interpreted as a signal of how mature the Confidence of a relation truly is.

If the Support falls below a predefined *Support threshold*, we assume that the relation between the label and the image has not been revised by a sufficient number of users. The aim of the Support rank is to protect the system against initial oscillations of the Confidence.

The interpretation of both measures is summarized in Table 1, where the '+' means the system regards the respective relation as true, '-' means the system regards the relation as false and '*?*' means the relation is yet to be further verified by users.

Table 1. Interpretation of the ranks attached to each pair label-image in our database.

Confidence rank / Support rank	Above Certainty threshold	Between thresholds	Below Suspicion threshold
Below Support threshold	?	?	?
Above Support threshold	+	?	−

The KA-CAPTCHA procedure

Whenever a new test is needed, our CAPTCHA will consult its knowledge base and collect a random label from it. Next, it retrieves a group of images from the repository: some that it knows relates to the label (the '+' condition from Table 1), some it knows do not relate to the label (the '-' condition), and some whose pertinence to the label it wants to verify ('*?*' in Table 1). All are randomly distorted due to the openness of our data and presented to the user as seen in Figure 4.

In our test, the selected label is displayed on the left and the images retrieved from the knowledge base are organized by columns on the right. One special column will contain all those images whose relation to the current label we want to verify, while all the others contain the remaining images whose relation to the label the system is confident about, i.e. they certainly do or do not relate to the label.

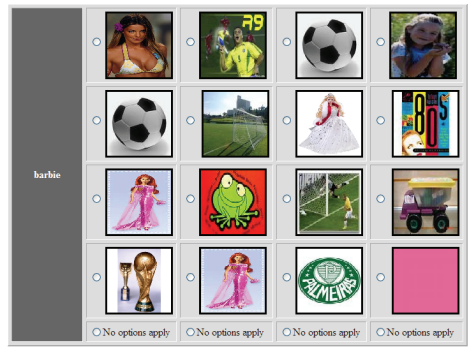


Figure 4. A KA-CAPTCHA designed to extract pictorial knowledge from its users.

The index of the special column in randomly chosen in each iteration, so that the user won't be able to distinguish it from all the other columns present in the test.

The introduction of a special column (where we are not sure there is at least one correct answer) forces us to provide the user with the option *No options apply*, which indicates that all images shown in the respective column are not described by the label on the left. Since the user should not distinguish a special column from all the others, we add the *No Options apply* alternative to all columns.

The user is then asked to identify, from each column available in the test, which image relates to the label. Since the existence of the special column is ignored by the users, we expect them to answer all columns with the same dedication, thus contributing with some knowledge with the same effort they would exert at trying to pass a regular CAPTCHA.

After the user submits a response to our CAPTCHA test, we initially consider all those columns whose correct answer we know a-priori. If the user fails to recognize some correct answer then it is considered a computer and access to the resource protected by our CAPTCHA is denied.

If all provided answers were correct, the user is viewed as a human and access to some resource is granted. Additionally, the system is now able to extract knowledge from the special column, and it does so by considering whatever option provided by the human in the special column to be correct.

With this new indication from users, the Support and Confidence ranks of our knowledge base are updated. The Support rank is increased by one if its corresponding image was selected by the user or if the *No options apply* alternative was checked. The Confidence is updated in a Bayes fashion, given the likelihood that a correct (incorrect) relation would be regarded as false (true) by an average user.

If the user chooses one of the images from the special column, and not the *No options apply* option, we do not update the rank of the remaining images.

The exact number of columns (call it *#c*) and the number of images per columns (*#i*) generated in this CAPTCHA test are configurable in the system. Since a computer might try to pass our test using random responses, such strategy

would yield a successful hit every $(\#i+1)^{\#c-1}$ attempts. So, *#i* and *#c* could be adjusted to reach any secure level of protection.

Empirical evaluation

We conducted an experiment to evaluate our approach against two measures. First, we wanted to confirm if our system is indeed capable of extracting valid knowledge from humans. Second, we wanted to verify how easily our CAPTHA can be solved by legitimate users.

We reproduced the scenario presented in the previous Section where a CAPTCHA agent is to collect data from a knowledge base of labels and images. The first stage of the experiment served to generate this knowledge base. A second phase represents the actual use of CAPTCHAs by volunteers and a final stage was designed to evaluate the data extracted from steps 1 and 2. There were no overlap of volunteers, i.e. each participant took part in one, and only one, stage of the experiment.

During the whole test, we used a Certainty threshold of 0.8, Suspicion threshold of 0.2 and a Support threshold of 2. The likelihood of a user recognizing a correct relation as false and recognizing a false relation as true were both estimated as 0.2. The initial value of all Confidence measures was 0.5. We used 5 images per column, with 4 columns per test.

Stage 1

Participants. Two volunteers from Brazil, ages 24 and 25 years old, took part in this stage for free. One was a male engineering graduate student and the other a female literature undergraduate. Participants had no previous knowledge of the work described in this paper or the participation of each other.

Procedure. At first, we collected the 15 most popular queries made to Google from Brazil during the month of September 2006 - data is available in (Google Press Center). Then, we presented each query as a label and asked our volunteers to retrieve from the Web, for each label, 10 images they believe related to the label and 10 images they believe did not relate to the label. Volunteer 1 managed to collect 101 images, while Volunteer 2 collected only 63. We labeled each group of images as Knowledge base 1 and Knowledge base 2, respectively.

Stage 2

Participants. 143 Brazilian computer science, physics and engineering graduate and undergraduate students indirectly took part in this stage, for free. Their respective professors, all belonging to our Computer Science department, supported this study by providing us their students' test grades. No participant had any knowledge of the research described in this paper.

Procedure. As to attract users to our CAPTCHA, we designed a small application where students would be able to consult their grades online in a private way. In our Computer Science department, student's grades used to be published on a piece of paper attached to a wall, with no attempt to preserve grades' confidentiality. With the advent of our system, professors were able to appropriately address student privacy.

We collected students' grades twice. Initially, in the middle of the semester when students accessed the system to view their grades to mid-term tests (when we used Knowledge base 1), and then at the end of the semester, when the system was accessed for information about final grades and class status, during which time we used Knowledge base 2.

Students were told by their instructor that the only way they could know their test score was through our system, but weren't notified of the presence of the CAPTCHA, how to solve it or what its objective was. All information they had about the system was its Web address and instructions in its interface (which was the Portuguese equivalent of "*Select, for each column on the right, the alternative that is best described by the label on the left*").

Stage 3

Participants. Two male volunteers from Brazil, 24 years old, took part in this final stage for free. Both were undergraduate students, one studying law and the other veterinary medicine. Both had no knowledge of the research underneath the experiment.

Procedure. In this stage, participants were presented with all the labels and images collected from stage 1 and were asked to assign, for each image, which labels describe the image. They were instructed to assign any number of labels they thought described the current image, including none if that was the case.

Results

After the second stage of our experiment, we were able to evaluate the knowledge extracted from volunteers by our CAPTCHA. Figure 5 shows a representation of these results, where each bar represents the final status of each knowledge base.

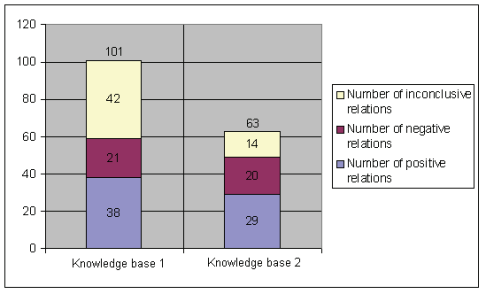


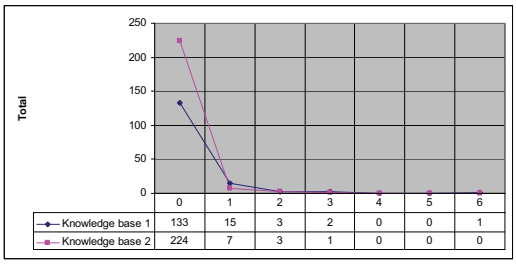
Figure 5. A representation of the knowledge extracted by our CAPTCHA from volunteers.

Knowledge base 1 was formed by 101 relations between images and labels. After all students had consulted the system for their mid-semester grades, we observed that 38 relations were regarded by the system as true, 21 had been considered false and 42 were still inconclusive (the system couldn't yet decide whether they were true or false). Comparing this knowledge collected by our CAPTCHA to

data collected from volunteers of stage 3, we noticed two false-negative relations (i.e. relations the CAPTCHA regarded as false but latter volunteers regarded as true) but no false-positives (i.e. all relations the CAPTCHA regarded as true were agreed by our volunteers).

Knowledge base 2 was initially composed of 63 relations. After the experiments, 29 of these were considered valid relations, 20 were considered false and 14 were still inconclusive. When comparing this data to the information extracted from stage 3, we observed two false-negatives but no false-positives.

One other claim we must verify is the ability of the average user to succeed on our tests. Figure 6 presents a graph indicating how many attempts our users have made before passing the CAPTCHA (0 means the user passed the test at first attempt, 6 means the user failed the test 6 times before finally succeeding). Eventually, every user passed the CAPTCHA test.



	0	1	2	3	4	5	6
Knowledge base 1	133	15	3	2	0	0	1
Knowledge base 2	224	7	3	1	0	0	0

Figure 6. Number of attempts made by volunteers before passing the KA-CAPTCHA.

Access to our CAPTCHA using Knowledge base 1 resulted in users passing the test in 81.48% of the attempts. Knowledge base 2 was used in a moment when users were more familiar with the system, as now they achieved the higher rate of success of 93.08%.

Conclusions

In this paper we introduced KA-CAPTCHA, a new approach to knowledge acquisition on the Web. We extended the CAPTCHA model and tried to extract knowledge out of the overhead generated by CAPTCHAs.

Our approach has the advantages of not requiring the advertising of a new tool or service, the potential to reach a more diversified set of users and not introducing any new burden into people's agenda on the Web.

We also have presented an application of KA-CAPTCHA where pictorial knowledge is to be extracted from humans. Empirical results suggest that this approach has the potential of extracting valid knowledge from users and still satisfy security needs.

We acknowledge that, when compared with other approaches to pictorial knowledge acquisition (von Ahn, 2005), our approach's performance is lower. However, we believe that the non-decreasing demand for Web resources might allow our method to attract larger amounts of knowledge in the long run.

On the other hand, we see our approach not as a substitute to existing knowledge acquisition efforts, and perhaps our method could be combined with existing solutions to achieve even better results. For instance, a very straightforward mixture of this approach with the single-player version of the ESP game might result in an efficient CAPTCHA that is fun to play as well.

We also envision many applications of KA-CAPTCHA that are not related to pictorial information. For instance, a CAPTCHA that acquires commonsense knowledge from Web users might contribute to the development of better computational agents. Another possibility would be the design of a test where Web designers could learn from visitors their general perception of a Web page. This might help detect problems with the affordance of a Web site.

We are currently working on a KA-CAPTCHA that can be experienced by visually impaired people as well.

References

von Ahn, L. 2005. Human Computation. PhD dissertation, Department of Computer Science, Carnegie Mellon University.

Apte, C.; Liu, B.; Pednault, E.; and Smyth, P. 2002. Business Applications of Data Mining. *Communications of the ACM*, 45(8):49-53, August 2002.

Avery, C.; Resnick, P.; and Zeckhauser, R. 1997. The Market for Evaluations. *American Economic Review* 89(3): 564-584

Brin, S.; and Page, L. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* 30:107-117

Chellapilla, K.; Larson, K.; Simard, P.; Czerwinski, M. 2005. Designing Human Friendly Human Interaction Proofs (HIPs). In *Proceedings of ACM CHI*, 2005, 711-720

Chklovski, T. 2003. LEARNER: A System for Acquiring Commonsense Knowledge by Analogy. In *Proceedings of Second International Conference on Knowledge Capture* (K-CAP 03) 4-12

Chklovski, T.; Gil, Y. 2005. An Analysis of Knowledge Collected from Volunteer Contributors. In *Proceedings of the Twentieth National Conference on Artificial Intelligence* (AAAI-05), 564-570. Menlo Park, Calif.: AAAI Press

Google Press Center. http://www.google.com/press/zeitgeist.html

Lenat, D. 1995. CYC: A Large Scale Investment in Knowledge Infrastructure. *Communications of the ACM* 38(11):33-38

Mori, G.; Malik, J. 2003. Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA. In *IEEE Computer Vision and Pattern Recognition*, 134-141, Madison, Wisconsin, June 2003

OpenMind Commonsense. commonsense.media.mit.edu

Rashid, A M., Ling, K., Tassone, R. D., Resnick, P., Kraut, R., Riedl, J. 2006 Motivating Participation by Displaying the Value of Contribution. In *Proceedings of ACM CHI 2006*, 955-958

Resnick, P., Varian, H. 1997. Recommender Systems. *Communications of the ACM*, 40(3):56-58

Singh, P., Lin, T., Mueller, E., Lim, G., Perkins, T., Zhu, W. 2002. Open Mind Common Sense: Knowledge acquisition from the general public. LNCS, 2519:1223-1237

Wikipedia. http://www.wikipedia.org