Automatic Generation of Game-based CAPTCHAs

Hong Yu and Mark O. Riedl
School of Interactive Computing, Georgia Institute of Technology
{hong.yu, riedl}@cc.gatech.edu

ABSTRACT

A Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) is a challenge-response test used on the Internet to prevent bots from accessing web services that are designed for humans. In this paper, we propose Automatic Game-based CAPTCHA Generation (AGCG), in which an AI system generates games that, when played, distinguish between humans and bots. The game-based CAPTCHA takes advantage of not only the bots' difficulty in performing pattern/object recognition, but also their lack of commonsense knowledge. Thus it is more secure but remains easy and fun for humans, compared to traditional visual based CAPTCHAs. Furthermore, our AGCG system is capable of learning new commonsense knowledge based on users' response in the game-based CAPTCHAs.

Keywords

CAPTCHA, Gamification, Procedural content generation, Commonsense knowledge

1. INTRODUCTION

A Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) is a type of challenge-response test used in computing to determine whether or not the user is human. The CAPTCHA is widely used on the Internet to prevent bots from accessing web services that are designed for humans. They are usually in a format of a visual or audio pattern recognition task created based on problems that are easy for humans but difficult for bots to solve [9]. One of the most common CAPTCHA tests is Google reCAPTCHA[1], which not only verifies humanity to prevent bot attack, but also uses human effort to digitize text, annotate images, and build machine learning datasets. The human responses to the reCAPTCHAs are used to preserve books, improve maps, and solve other problems that are difficult for computers.

[1]http://www.google.com/recaptcha/intro/index.html

Developments in artificial intelligence have increased bots' abilities to complete the standard visual and audio tasks, thus making it difficult for CAPTCHAs, even the Google reCAPTCHA, to distinguish humans from bots [5]. While researchers are developing new visual and audio CAPTCHAs to prevent bot attack and improve security for online computer systems, the increased noise and distortion in the CAPTCHAs also make it more difficult for humans, especially non-English speakers, to pass the test [6].

In this work, we propose a new type of game-based CAPTCHA that is easy for humans while difficult for bots. The game-based CAPTCHA verifies humanity through asking the users to play a quick, simple game, such as selecting proper toppings from a set of images to add to a pizza. Our CAPTCHA games take advantage of not only the bots' difficulty performing pattern/object recognition, but also their lack of commonsense knowledge. Thus it provides an additional layer of security while remains easy and fun for humans, compared to traditional visual based CAPTCHAs.

Making games is difficult. Even simple CAPTCHA games can be time and effort intensive to make. If only a small amount of the game-based CAPTCHAs are created, a simple brute-force attack can be used directly to crack the CAPTCHAs. Unfortunately, it seems unrealistic for humans to author a large enough pool of games manually. In this paper, we propose a fully automatic AI approach to create the game-based CAPTCHAs. Our Automatic Game-based CAPTCHA Generator (AGCG) relieves the authoring effort for creating the game-based CAPTCHAs.

The AGCG needs a commonsense knowledge source to automatically generate the game based CAPTCHAs. The knowledge source could be a private knowledge database for commercial CAPTCHA service providers. In this research, we use ConceptNet[2] to bootstrap our knowledge database because ConceptNet is freely available for research purpose. Similar as Google reCAPTCHA, our AGCG is capable of learning new concept relationships that do not exist in current knowledge database. We apply a node similarity algorithm—SimRank [7]—to retrieve candidate concept relationships from the relationships of similar concepts in the database. These candidate relations will be verified by humans later in the generated CAPTCHAs. The learned relationships between concepts will be included into the commonsense knowledge base to create new CAPTCHAs.

[2]http://conceptnet5.media.mit.edu/

(a) Pessimal Print (b) GIMPY

(c) Google reCAPTCHA (d) BaffleText

Figure 1: Examples of visual CAPTCHA.

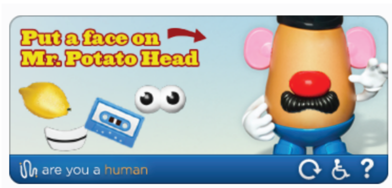

Figure 2: An example of Are You A Human CAPTCHA.

2. RELATED WORK

Figure 1 shows four examples of common CAPTCHAs: Pessimal Print [3], GIMPY [1], Google reCAPTCHA (which also includes an audio form of test), and BaffleText [2]. To pass the CAPTCHA tests, the users have to correctly decipher the letters in the images. As shown in Figure 1, these traditional CAPTCHAs are not user friendly, especially for non-English speakers [6].

A logic question CAPTCHA is a type of CAPTCHA that asks the users to solve simple mathematical problems that are automatically generated according to some templates[3]. The logic question CAPTCHA distinguishes bots using the commonsense knowledge of the humans. Compared to the traditional visual based CAPTCHAs, the logic based CAPTCHA is easier for humans. However, it is not as secure as traditional visual based CAPTCHAs due to the limited number of templates and the lack of pattern recognition tests [9].

Are You A Human (AYAH)[4] is a promising new type of game-based CAPTCHA for commercial use. Figure 2 shows a screenshot of one AYAH CAPTCHA. In Figure 2, the user needs to select the eyes and the mouth from the four images on the left and drag them into the Mr. Potato Head on the right. The AYAH CAPTCHAs are easy and fun for humans. But all the AYAH games are human-authored. Thus the authoring bottleneck limits the amount of the games. The AYAH CAPTCHA has been cracked by simple brute force attacks as follows: a specially designed bot is programmed to finish one particular AYAH game; if the bot encounters AYAH games other than the one it knows, it will refresh the CAPTCHA until that particular one shows up[5].

[3]http://www.smashingmagazine.com/2011/03/04/in-search-of-the-perfect-captcha/
[4]http://areyouahuman.com/
[5]http://spamtech.co.uk/software/bots/cracking-the-areyouhuman-captcha/

Our AGCG system is capable of not only automatically generating the game-based CAPTCHAs, but also learning new commonsense knowledge using human feedback. Thus it is a type of *human computation system*—an intelligent system that organizes humans to manually carry out computational processes that are too hard to solve, to collect commonsense knowledge typically not available to computational systems, or to label data [8]. The learned knowledge can contribute to ConceptNet, as well as create more secure CAPTCHAs in the future. Our game-based CAPTCHA can be viewed as a *Game With A Purpose* (GWAP), in which players generate useful data or solve problems as a by-product of play [10].

3. AUTOMATIC GAME-BASED CAPTCHA GENERATION

Our game-based CAPTCHAs verify humanity through asking the users to sort concepts according to their component membership relationship. Provided with two randomly selected test concepts and a list of component concepts, the user must sort the components according to which test concept they have membership with. In the component list, we intentionally include a few concepts that the CAPTCHA itself does not know which test concept the component concepts should be correlated to. The CAPTCHA will use the aggregate responses from the users to infer the relationships for those unknown concepts.

A basic requirement to pass the game-based CAPTCHAs is commonsense knowledge, which is the collection of facts and information that an ordinary human is expected to know. The game-based CAPTCHAs take advantage of the fact that bots lack the commonsense knowledge. Unfortunately, our AGCG itself is also a computer system. To automatically generate the game-based CAPTCHAs, we must build a database of commonsense knowledge on the component membership relationship for the AGCG system. In the remainder of this section, we will first describe how we acquire the commonsense knowledge on concept relationship. Then we will describe our proposed techniques to learn new knowledge that is missing in current knowledge datase and our preliminary implementation.

3.1 Knowledge Acquisition

In practice, any knowledge database can be used as the knowledge source for the game based CAPTCHA generation. For CAPTCHAs that are for commercial use, a private knowledge database should be used for security purpose. In this research, our AGCG system uses the crowd-sourced database ConceptNet to bootstrap our knowledge database because ConceptNet is easily accessible to us for research purpose. ConceptNet is a large connected semantic graph in which nodes represent concepts and edges represent relationships, such as *part-of*, *contains*, etc. The concepts are in the form of words or short phrases of natural language. Figure 3 shows a subgraph extracted from ConceptNet. The *part-of* relation in ConceptNet is used as our source of knowledge for the component membership relationship. The AGCG randomly selects a subset of concepts that must have at least four inbound *part-of* relations from ConceptNet as the test concepts. Currently, we manually filter out inappropriate concepts and relations, such as offensive or adult contents, from the subset. The components

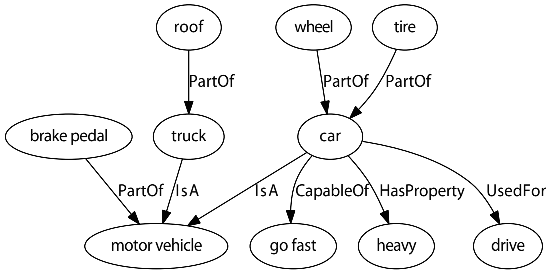

Figure 3: A subgraph extracted from the ConceptNet.

concepts are selected randomly from the concepts that have *part-of* relations with the test concepts.

3.2 Learning New Knowledge

Our game based CAPTCHAs are capable of learning new concept relationships that do not exist in the current knowledge database. We include a few concepts that do not have *part-of* relation with either of the two test concepts into the component list. The response from users who passed the test will be used to infer the relations between the unknown component concepts and the test concepts. We will use a two-tailed binomial statistical test to validate whether to accept the relationships as new knowledge or not.

In theory, any concept that is not related with the test concepts can be used as the unknown component concepts. But it is highly unlikely that a randomly selected concept has *part-of* relationship with the test concepts. To increase the knowledge learning speed for the game-based CAPTCHAs, we use a node similarity algorithm—SimRank [7]—to select component concepts that have a high probability of being related to the test concepts. The SimRank algorithm computes similarity between two nodes by comparing the similarity between the neighbors of the two nodes in a graph. Applying the algorithm to our AGCG system, we assume that the concept *car* in Figure 3 is the test concept. Using the SimRank algorithm, we find that *car* is similar to *truck* in the subgraph extracted from ConceptNet. Then the concept *roof* will be selected as our unknown component concept since it has part-of relation with *truck* but not with the original concept *car*.

The new commonsense knowledge learned by the CAPTCHAs can be used not only as a supplement to the existing knowledge database, but also to build new game-based CAPTCHAs that are more resistant to bots. As the knowledge database grows, the number of the CAPTCHAs that can be generated increases. Thus the generated game based CAPTCHAs will be more resistant to bot attack.

3.3 Preliminary Implementation

We have built a preliminary implementation for the AGCG system. At current stage, the AGCG is capable of retrieving the commonsense knowledge from ConceptNet, choosing unknown component concepts using SimRank, and creating the game-based CAPTCHAs with a template. Figure 4 shows a screen shot of one of the generated CAPTCHAs. In Figure 4, the CAPTCHA shows two test concepts: *bicycle* and *laptop*

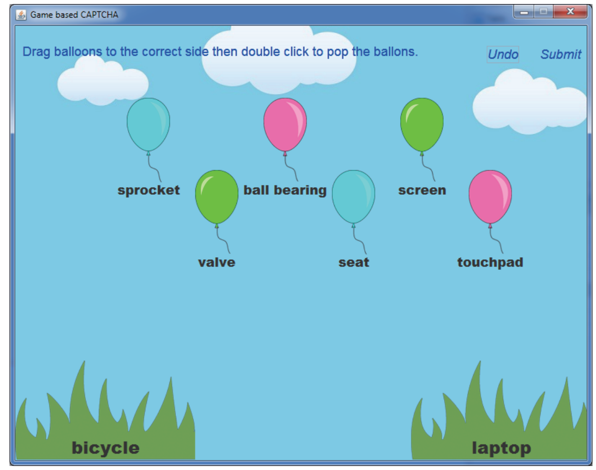

Figure 4: A screenshot of a preliminary game-based CAPTCHA.

(bottom left and bottom right), and a list of component concepts: *screen*, *touchpad*, *sprocket*, *ball bearing*, *valve*, *seat* (hanging on six balloons). The CAPTCHA knows *screen* and *touchpad* have *part-of* relationship with *laptop*, and *seat* and *sprocket* have *part-of* relationship with *bicycle*. Using SimRank, our AGCG algorithm finds that the concept *machine* is similar to the test concept *bicycle*. Thus two of the concepts that have *part-of* relationship with machine, i.e. *ball bearing* and *valve*, are selected as our unknown component concepts. In the game, a user needs to drag each balloon to the left or the right side then pop the balloon (through double-clicking) to drop the component concept onto the test concept using gravity. The component concepts that do not correlate with either of the test concepts should be left floating in the air. The user has to correctly indicate all or a percentage of (e.g. three out of four) the relations for the concepts *screen*, *touchpad*, *seat*, and *sprocket* to pass the test. A binomial test will be used to analyze the users' response for the two unknown concepts, *ball bearing* and *valve*. New *part-of* relationship will be learned if we get statistically significant results from the binomial test.

It is possible that a selected component concept that has *part-of* relationship with one test concept also has *part-of* relationship with the other test concept in a single CAPTCHA, even when the relationship between the component concept and the other test concept is missing in current ConceptNet. To minimize the chance of this coincidence, the AGCG system selects two concepts that are at least k steps away in ConceptNet as the test concepts, where k is a pre-defined constant. We can also work around the problem through requiring the users to correctly answer a percentage of the component concepts.

3.4 Security Enhancements

To make our game based CAPTCHAs more secure and more fun, we propose three further enhancements: converting concept labels into images, procedural content generation, and using user's in-game player features.

Converting concept labels into images makes the game-based CAPTCHAs more resistant to bots which must use computer vision to recognize objects. We plan to use a similar approach as in Spritely [4], which converts words into sprites downloaded from Internet. Google clipart image search can be used to convert the concept labels into clipart images. The noun project[6] can also be used as the source of image conversion. It can be difficult to find images for concepts referring to abstract objects. For example, the clipart image for the concept *heavy* in Figure 3 might be misleading for humans. Thus we only use the *part-of* relationship in ConceptNet to build the game-based CAPTCHAs so that the connected concepts are more likely to refer to physical objects. In the future, we will use machine learning to classify each concept into physical category and abstract category based on the concept's relations with other concepts.

To make the CAPTCHAs more fun for humans, we plan to use procedural content generation (PCG) to generate game content for the CAPTCHA. PCG has been widely used in computer game area to automatically generate game terrain, levels, characters, and game mechanics [11, 12]. In our CAPTCHA, instead of asking the users to directly drag the related concepts together, we will procedurally generate various types of games using templates, such as shooting a component concept to its related test concept similar as in Angry Birds, dropping the component concepts using gravity into buckets labeled with the test concepts, and flinging the component concepts with inertia,etc. The PCG not only makes the CAPTCHAs more fun for humans, but also increases the difficulty for bots, because it will require the ability to play many types of games and PCG makes it harder to recognize the templates and to recognize the concepts, either in the form of texts or images, in the templates.

To further increase the security for the game-based CAPTCHA, the CAPTCHA can utilize a user's in-game player features, such as the speed of cursor movement, the pattern of cursor movement and the pattern of mouse click, etc., to help decide whether the user is a bot or not. Humans' in-game feature data are needed to train a classifier that identifies the human behavior.

4. DISCUSSION AND CONCLUSIONS

In the preliminary implementation, the game based CAPTCHA uses text based concept labels. Thus a bot equipped with computer vision skills can easily recognize the text in the game. But to crack the CAPTCHA, the bot also needs to reason about the relationship between the concepts, either through searching online or cracking into the knowledge database. Although we extract the initial knowledge database from ConceptNet which is publicly accessible, the AGCG system can be easily deployed with a private knowledge database which is more secure for commercial use. Ideally, private knowledge database has relations that do not overlap entirely with public commonsense knowledge databases due to the sparseness of knowledge database and the large amount of possible commonsense relations.

It may take a bit longer for players to finish our game based CAPTCHA than a traditional visual based CAPTCHA on a desktop computer. But the games may bring more enjoyment to a user than an OCR task, to be determined by evaluation. Our game-based CAPTCHA could be more appropriate for mobile environments where it is easier for the users to swipe and drag than to type in words.

The CAPTCHA is an important mechanism to prevent bots from accessing web services. A growing research community is studying how to build new CAPTCHAs that are resistant to bots while easy for humans. Our automatically generated game-based CAPTCHAs combine the security of the traditional visual based CAPTCHAs, the human friendliness of the logic based CAPTCHAs, and the fun of computer games. It can automatically generate large enough number of game based CAPTCHAs to prevent simple brute-force attacks. Thus we believe that our game-based CAPTCHAs are capable of creating a more secure environment on Internet and providing a better web service to users.

[6]https://thenounproject.com/

5. REFERENCES

[1] L. V. Ahn, M. Blum, and J. Langford. Telling humans and computers apart automatically. *Communications of the ACM - Information Cities*, 47(2):56–60, 2004.

[2] M. Chew and H. S. Baird. Baffletext: a human interactive proof. Proceedings of the 10th IS&T/SPIE Document Recognition & Retrieval Conference, 2003.

[3] A. L. Coates, H. S. Baird, and R. J. Fateman. Pessimal print: A reverse turing test. pages 1154–1158. Proceedings of the 6th International Conference on Document Analysis and Recognition, 2001.

[4] M. Cook and S. Colton. A rogue dream: Automatically generating meaningful content for games. AIIDE 2014 Workshop on Experimental AI In Games, 2014.

[5] C. Cruz-Perez, O. Starostenko, F. Uceda-Ponga, V. Alarcon-Aquino, and L. Reyes-Cabrera. Breaking recaptchas with unpredictable collapse: Heuristic character segmentation and recognition. *Pattern Recognition Lecture Notes in Computer Science*, 7329:155–165, 2012.

[6] C. A. Fidas, N. M. Avouris, and A. G. Voyiatzis. On the necessity of user-friendly captcha. pages 2623–2626. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2011.

[7] G. Jeh and J. Widom. Simrank: A measure of structural-context similarity. pages 538–543. Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining, 2002.

[8] E. Law and L. V. Ahn. Human Computation. *Morgan and Claypool*, 2011.

[9] K. R. Soumya, R. M. Abraham, and S. K. V. A survey on different CAPTCHA techniques. *International Journal of Advances in Computer Science and Technology*, 3(2):117–122, 2014.

[10] L. von Ahn and L. Dabbish. Designing games with a purpose. *Communications of the ACM*, 51(8):58–67, 2008.

[11] G. N. Yannakakis and J. Togelius. Experience-driven procedural content generation. *IEEE Transactions on Affective Computing*, 2(3):147—161, 2011.

[12] A. Zook and M. O. Riedl. Automatic game design via mechanic generation. Proceedings of the 28th AAAI Conference on Artificial Intelligence, 2014.