9-29-2011

Report on Advances in the Field of Artificial Intelligence Attributed to CAPTCHA

Craig M. Schow
Utah State University

REPORT ON ADVANCES IN THE FIELD OF ARTIFICIAL INTELLIGENCE ATTRIBUTED TO CAPTCHA

By

Craig M. Schow

A report submitted in partial

fulfillment of the requirements for the degree

of

MASTERS OF SCIENCE

in

Computer Science

Approved:

_____	_____
Dr. Donald Cooley	Dr. Stephen Clyde
Major Professor	Committee Member

_____

Dr. Nicholas Flann

Committee Member

UTAH STATE UNIVERSITY

Logan, Utah

2011

ABSTRACT

Report on Advances in the Field of Artificial Intelligence

Attributed to CAPTCHA

by

Craig M. Schow, Master of Science

Utah State University, 2011

Major Professor: Dr. Donald Cooley
Department: Computer Science

A CAPTCHA is a specialized human interaction proof that exploits gaps between human and computer recognition abilities. By design, the hardness of a CAPTCHA is based on the difficulty of advancing the underlying artificial intelligence [AI] technology to a level that eliminates any exploitable gap. Due to this fact computer scientists have concluded that the widespread use of CAPTCHA would accelerate research in the underlying fields of AI eventually leading to near-human capabilities in certain AI systems. Despite these predictions no attempt has been made to identify advances in AI which can be attributed to the use of CAPTCHA.

The goal of this report is to explore the concept of CAPTCHA as a catalyst for advancement in AI. As part of this goal I examine the underlying basis for expected contributions, provide direct examples of documented advancements that have already been made, evaluate the strengths and weaknesses of the CAPTCHA model and based on the results identify specific areas of AI most likely to benefit from CAPTCHA in the future.

As a result of my research I have found that some advancement has been made as a result of CAPTCHA, but due to weaknesses in many CAPTCHA implementations these advancements have been limited and have often fallen short of expectations. As many of these weaknesses have been identified new methods of implementation have been introduced, but many of these have limitations as well. As part of the exploration of these challenges I have provided a basis that will allow for a more accurate understanding of the processes involved, and allow others to continue to build on the work which has already been done.

CONTENTS

CHAPTER 1

INTRODUCTION

A CAPTCHA (**C**ompletely **A**utomated **P**ublic **T**uring Test to Tell **C**omputers and **H**umans **A**part) is a type of challenge-response authentication that exploits gaps between human and computer recognition abilities in order to determine if the user is human. By design, the hardness of a CAPTCHA is based on the difficulty of advancing the underlying artificial intelligence [AI] technology to a level that eliminates any exploitable gap. This type of system is similar to some forms of public key cryptography where the underlying hardness is based on the difficulty of factoring large numbers. Thus, if an attacker can solve the underlying problem they will be able to defeat the protocol. Due to this fact, many computer scientists have concluded that the widespread use of CAPTCHA would accelerate research in the underlying field of AI eventually leading to near-human capabilities in certain AI systems much like cryptography has motivated research on algorithms and hardware used for factoring [1, 2, 3, 4, 5, 17].

Despite the common acceptance of CAPTCHA as a catalyst for advancing research in some fields of AI, very little work has been done to check the validity of this assumption or identify weaknesses that could be addressed. This report is intended to fill in the knowledge gap between the expectations that have been expressed and the actual results that have been produced. I have divided this report into the following chapters:

Overview of CAPTCHA

In this chapter I detail the main uses of CAPTCHA, describe how it works, and introduce any terms that are relevant to the report. This chapter contains enough detail to allow an individual who is unfamiliar with CAPTCHA to follow the ideas expressed in other sections of the report.

Basis of CAPTCHA as a Catalyst

In this chapter I explore the initial reasoning behind the idea that CAPTCHA would have a positive effect on the field of AI. I provide examples of related problems on which this reasoning is based and draw conclusions concerning expected outcomes. I also detail any underlying assumptions which have been made.

Direct Examples

In this chapter I identify specific examples of recent advancements in the field of AI that can be attributed in part to the use of CAPTCHA. Based on my findings, I identify strengths that have lead to success and areas in which advancements have fallen short of expectations.

Opposing Views

In this chapter I explore many of the weaknesses that have been identified in current implementations of CAPTCHA. As weaknesses are presented I identify how each limits the potential effectiveness of CAPTCHA in advancing AI.

Future Improvements

In this chapter I take what has been learned and project that knowledge forward in an attempt to outline future contributions that could be made. I identify specific areas for improvement and outline areas of AI most likely to be impacted. Finally, I identify a general framework that can be applied to future problems

CHAPTER 2

OVERVIEW OF CAPTCHA

CAPTCHA

A CAPTCHA is a type of challenge-response authentication schema that asks "Are you human?" [2] The term CAPTCHA stands for "**C**ompletely **A**utomated **P**ublic **T**uring Test to Tell **C**omputers and **H**umans **A**part" [7]. Many authors refer to CAPTCHAs as a type of Human Interactive Proof (HIP). In order to differentiate between a human and a computer the CAPTCHA program generates a test that is easily solved by humans, but difficult to solve using a computer [1]. Generally these tests take the form of distorted text, images or audio and are often found at the bottom of a web form [7]. The goal of each user is then to authenticate themselves as human by correctly recognizing or classifying the item. See Figure 1 below.

Figure 1. Text-based CAPTCHA. Note the distortion in the text.

Although an ideal system would be able to correctly differentiate between humans and computers with perfect accuracy the CAPTCHA process does not need to be perfect in order to work in practice. If humans are rarely misclassified as computers the system will be acceptable to them. If computers are detected with a reasonable high probability, the threat of abuse will be significantly reduced [2]. Methods such as gap amplification (detailed below) are designed to allow for greater accuracy and effectiveness by increasing the probability that a computer will be detected while limiting the misclassification of humans. These methods allow the level of security and usability to be adjusted to meet the specific requirements of each application.

How CAPTCHA Works

A CAPTCHA takes advantage of the gap between the success rate of humans, and the current success rate of computers in solving difficult AI problems [1, 6]. Such a system works because although an attacker can duplicate his/her programs and data, he cannot duplicate his human pattern recognition skills without making at least some advance in AI technology [2]. For example even though a programmer may personally be able to identify objects within a distorted image creating a program with that same capability is considerably more difficult.

Thus, in order to mount an automated attack he/she would first need to solve the underlying recognition problem before the automated attack could be performed.

In general a CAPTCHA challenge works as follows:

1. A computer generates a challenge, and asks the user to solve it.
2. The user solves the test and sends the response.
3. The computer grades the response, and if correct, it is assumed that the user is human.

Each challenge test generated as part of a CAPTCHA must meet the following requirements:

Human Executable

A test is said to be human executable if the target population can complete the test with at least the minimum desired success rate [1, 2, 3]. It is important to note that the success rate for a given group may depend on factors such as language, education and the presence of disabilities [1, 3, 6]. For example individuals with a vision impediment, or lower resolution screen may have difficulty correctly identifying a text based CAPTCHA. In general usability issues related to text-based CAPTCHAs are addressed by providing the user with an audio-based option instead, but even this fails to allow for individuals with multiple disabilities or who speak another language.

Hard AI Problem

Due to the continuing advances being made in the field of AI the hardness of a problem is not based on any set standard, but rather on the current state-of-the-art. An AI problem is said to be hard if the AI community agrees that it is hard to find a solution that meets the required success rate [1, 2, 4]. The computational time required to find a solution may also be considered if a timeout method is provided [2].

Currently text, audio and image recognition are the most commonly used CAPTCHAs. Although there are a wide range of optical and voice recognition applications available, none approaches the recognition ability of humans, and for some, the large computational cost of achieving a high level of accuracy is prohibitive [2]. For example one difficulty with optical character recognition (OCR) is creating a recognizer that generalizes well enough to identify text of different fonts and distortions. Modest improvements to a recognizer could be accomplished by using extremely large data sets when training but the computational cost of doing so would be prohibitive. CAPTCHAS are designed to exploit the known limitations in current technology.

Computer Generated

In order for a CAPTCHA test to be useful there needs to be an automated way to generate tests and their solutions [1, 6]. In general this is done by taking a known object (text, image, audio, etc.) and applying a random set of transformations and adversarial clutter [1]. The

transformation of an object in this matter is sometimes referred to as Turing-resistant hashing [2].

Considerable work has been done on identifying random transformations that can be performed on text while retaining human-readability (See [6]). Some types of transformations that have been applied to text include the following:

Context-Free

Since OCR (Optical Character Recognition) and other types of object recognition typically use context-based correction as part of its recognition, the randomization of context-free or randomized objects renders this capability useless [2]. For example a recognizer might read the word “Human” as “Hvman” but then be able to correct for the error using a dictionary. A CAPTCHA that uses using random letters instead of complete words can prevent an automated recognizer from being able to correct using a dictionary.

Kerning

Kerning involves adjusting the amount of or removing the space between characters. This increases the difficulty of isolating individual characters using OCR [2, 6]. See Figure 2 below.

Figure 2: A word that has been kerned. Note the reduced character spacing for letters k-r-n and the increased spacing for n.

Stretch or Compression

In this method, a random level of stretch or compression is applied to the object to increase the difficulty of OCR recognition [2]. See Figure 3 below.

Figure 3: A word that has been stretched and compressed. Note the stretching of the S character and the compression of e-t-c-h.

Varying the Font, Style and Size

Many types of OCR systems are tuned for the specific type of document that will be scanned and thus will have difficulty identifying text of varying fonts, styles and sizes [2]. See Figure 4 below.

Figure 4: Two words composed of varying fonts and sizes. Note the variations in size and font.

Spatial Transformations

These types of transformations include rotation of the object to the left or right and randomizing its location within an image [2]. See Figure 5 below.

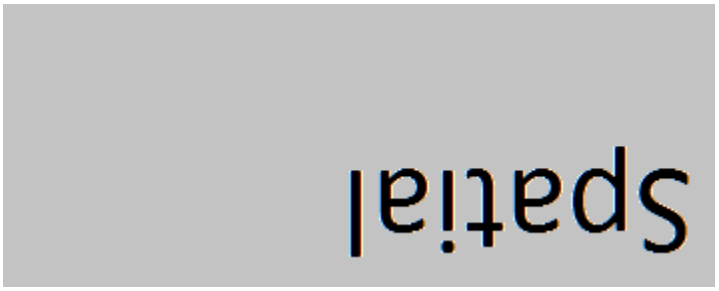

Figure 5: A word that has undergone spatial transformation. Note that the word has been rotated 180 degrees and is located in the bottom right of the image.

Fill and Noise

Fill and noise includes the addition of textures, squiggles, circles, arcs or other nonsensical writings, images or symbols [2, 6]. The presence of random objects makes recognition of underlying characters more difficult. Collectively fill and noise are often referred to as adversarial clutter. See Figure 6 below.

Figure 6: Words inside of a noisy image. Note the lines, shapes and other markings included in the image.

Lossy Image Compression

Lossy compression also makes the recognition process more difficult, while making little difference to humans [2, 6]. See Figure 7 below.

Figure 7: A text image that has undergone lossy compression. Note the loss of clear character outlines and the distortion of colors.

The security or difficulty of the CAPTCHA is based the difficulty of solving the AI problem involved, not on the complexity of the algorithm used in creating the CAPTCHA. As is the case with most security protocols, it should always be assumed that the adversary has full knowledge of the algorithm used. In the case of a CAPTCHA, the only information hidden from the adversary is the randomness used in the transformation processes used to generate the CAPTCHA challenge [1, 2, 7].

Gap Amplification

In many cases, the difference in the rate of success between humans and computers is smaller than desired. This problem can be solved by repeating the test multiple times in a process called Gap Amplification [1]. For example if the success rate for humans and computers is as follows:

Human: 98%
Computer: 15%

In this situation the gap between computers and humans would be 83% (.98 – .15 = .83 or 83%). Given the ability to make repeated attempts, a computer could easily defeat the CAPTCHA. If you required that two separate CAPTCHA challenges be completed successfully in sequence then the success rate for humans and computers would be as follows:

Human: .98 * .98 = .9604 or 96.04%
Computer: .15 * .15 = .0225 or 2.25%

The gap in success rates for humans and computers is now 93.79% (.9604 - .0225 = 93.79 or 93.79%). In this case it is now six times more difficult for a computer to defeat the CAPTCHA, but only marginally more difficult for a human. It is important to note that although you could use this process to further amplify the gap between humans and computers, it is unlikely that users would tolerate the increased hassle of having to solve large numbers of CAPTCHAs.

The use of gap amplification is most prominent in systems where the need for increased security overrides the desire for usability. For example Google's Gmail system only requires a user to solve a single CAPTCHA in order to create an e-mail account, but if a user forgets their password they are required to solve a CAPTCHA each time they attempt to login. In this case the need to prevent an automated system from being able to access a user's e-mail account was deemed sufficient to require an added level of security at the cost of usability.

Uses of CAPTCHA

The primary uses of CAPTCHA involve situations where it is desirable for a computer to distinguish between another computer and a human. This would include the following:

Online Polls

In November 1999 Slashdot.com released an online poll asking which graduate school was best for studying computer science. At the time Slashdot.com had mechanisms in place to restrict users to one vote per IP address in an attempt to prevent the same user from submitting multiple votes. Students from both Carnegie Mellon and MIT were able to bypass these restrictions by creating automated voting programs capable of forging votes to use the IP addresses of other computers on the local network [1, 7]. Although the Slashdot.com online polls are still susceptible to this kind of attack, many other online polls now require voters to complete CAPTCHAs before accepting the vote in order to prevent automated voting.

Free Email Services

Several companies offering free e-mail services have suffered attacks from programs designed to automate the creation of large numbers of e-mail addresses which are then used to generate spam [1, 4, 7]. Many companies including Yahoo, Google and Microsoft now require users to complete a CAPTCHA as part of the process to create an e-mail account using their service. By using CAPTCHA challenges as part of the registration process it becomes much more difficult to automate the registration process.

Spam

Spamarrest.com has implemented an e-mail filtering system that relies on CAPTCHA to block unwanted messages from non-human senders. The system works by maintaining a list of all e-mail addresses that are recognized by the recipient as being owned by a human. If an e-mail is received from an e-mail address that is not on the list of approved senders then the Spamarrest.com system will send an e-mail response asking the sender to complete a CAPTCHA challenge. If the CAPTCHA challenge is successfully completed then the e-mail is delivered to the recipient and the senders address is added to the list of approved senders, otherwise it is marked as spam. This helps insure that only e-mails of human origin will be received by Spamarrest.com users.

Preventing Dictionary Attacks

A computer program can attempt to gain unauthorized access to a system or account by repeatedly trying to login using different passwords in what is known as a dictionary attack. The traditional countermeasure for this type of attack is to lock or disable the account after a given number of failed attempts [3]. Given that many types of accounts are sequential (bank accounts) or otherwise predictable in nature, this approach has the drawback of allowing the attacker to

perform a denial of service attack by freezing the accounts of valid users [2, 3]. For a more detailed treatise on preventing dictionary attacks see [9].

In order to increase the difficulty of performing dictionary attacks while preventing valid users from being locked out of the system, many services such as Hotmail and Gmail now require the user to complete a CAPTCHA challenge as part of every subsequent login attempt after three unsuccessful login attempts have been made. This type of system effectively prevents repeated automated dictionary attacks on an individual account. This method does not prevent another type of attack where a common password is tried once for each account in the system.

CHAPTER 3

BASIS OF CAPTCHA AS A CATALYST

CAPTCHA as a Catalyst

Due to the fact that the underlying hardness of a CAPTCHA is based on an unsolved AI problem, solving a CAPTCHA implies solving the underlying AI problem [1]. This is often described as a win-win situation, i.e. either the CAPTCHA will remain unsolved, providing a way to differentiate between humans and computers, or the underlying AI problem will be solved [1].

One example of security protocols becoming ineffective due to advances in technology is that of the Data Encryption Standard (DES) adopted in 1979. The hardness of the DES system was based on the difficulty of breaking a 56-bit key. Despite enormous efforts by both governments and individual researchers to find weaknesses in the DES protocol their efforts proved largely unsuccessful. Notwithstanding the apparent lack of weaknesses in the DES protocol, the DES standard is now considered insecure because advances in computer processing power have allowed for simple brute-force attacks to be performed in a matter of a few days.

Precisely Stating the Problem

Part of the difficulty involved with AI is that it is rare for the problem to be precisely stated [1]. “We believe that precisely stating unsolved AI problems can accelerate the development of Artificial Intelligence: most AI problems that have been precisely stated and publicized have eventually been solved” [1]. The most frequent example used is that of Chess.

Chess

The literature on the subject of chess-playing machines dates back to the late 19th century [21]. Chess has been described as an ideal problem for computing because:

> "(1) The problem is sharply defined both in allowed operations (the moves) and in the ultimate goal (checkmate); (2) It is neither so simple as to be trivial nor too difficult for satisfactory solution; (3) Chess is generally considered to require 'thinking' for skilful play; a solution of this problem will force us either to admit the possibility of a mechanized thinking or to further restrict our concept of 'thinking'; (4) The discrete structure of chess fits well into the digital nature of modern computers" [21].

The ability to precisely state the problem when creating algorithms to solve chess allows for the construction of specialized "Knowledge Spaces" which aide in the search space pruning process by allowing the algorithm to eliminate all illegal moves [19]. For example many early estimates of the complexity of Chess did not account for illegal positions such as pawns in the first row, multiple pieces on a single square and both kings being in check. By eliminating illegal positions from the search space there is an exponential reduction in the size of the search space. Considerable work has also been done to further reduce the potential search space by attempting to understand the motives of the game (e.g. A player's motivation is to win the game, thus all moves made by the player will have that goal in mind.) Pruning by motivation and the elimination of illegal moves are only effective if the problem is clearly stated and the desired outcome is clearly understood.

Although various techniques involving pruning combined with brute force are theoretically capable of solving chess, the computational cost of running these algorithms to completion is well beyond the abilities of current computing technology. Consequently projects such as IBM's Deep Blue combine elements of the following:

- Extended Book: Deep Blue uses an extended book algorithm to evaluate the effectiveness of the first 30 or so moves from its database of over 700,000 Grandmaster chess games [20].
- Endgame Database: Some of Deep Blue's algorithms use a pre-calculated table of ending moves when the number of pieces reaches a certain population size. The resulting reduction in complexity resulting in fewer remaining pieces allows for the search to run to completion [20]

It is important to note that although there are similarities between Chess and the AI problems typically used in CAPTCHA there are important differences as well. These differences include the following:

- In Chess the pieces are always the same, and there are a finite number of location combinations. AI problems such as OCR may have to compensate for an infinite variety of positions and transformations.

- In Chess there are heuristics which can be used to evaluate the strength of a resulting position after each move given the context of all pieces on the board. In many AI problems there are no known methods to evaluate the effectiveness of a partial solution until it has been run to completion.

Inducing Research

Another difficulty involved with finding better methods for solving AI problems involves the lack resources being dedicated to solving a particular problem. The inclusion of AI problems in CAPTCHA provides incentive to researchers as well as malicious programmers to advance the field of AI [1]. The most common example given is that CAPTCHA will help to induce AI research, much like research in factoring has benefited from its inclusion in modern cryptography [1, 7].

Factoring

The problem of factoring integers has been around for a long time. In 1967 two mathematicians stated that "in general nothing but frustration can be expected to come from an attack on a number of 25 or more digits, even with the speeds available in modern computers" [11]. Another mathematician stated that "In those days [1970's], integer factorization was not fashionable, and there was not much interest in going after records" [10].

With the advent of public-key cryptography in the mid 1970's there has been an increased focus in the field resulting in many advances in both the algorithms and computational power dedicated to factoring large integers. Notable advances in algorithms include the elliptical curve factoring algorithm in 1985 and the number field sieve in 1990 (See [10, 12] for more information.) Currently, the largest number to be factored is the RSA-768 [14].

The evidence that factoring has benefited from its use in public-key cryptography is overwhelming [10, 11, 12, 14]. Over the past 20 years nine of twelve new factoring records came as the direct result of an RSA (a public-key cryptography algorithm) challenge at the cost of thousands of CPU years [13].

CHAPTER 4

DIRECT EXAMPLES

Mori and Malik

In 2003 vision researchers Greg Mori and Jitendra Malik of the University of California, Berkeley decided to take up the challenge of defeating two types of word-based CAPTCHA (EZ-Gimpy and Gimpy), both of which were being used by Yahoo to prevent bots from creating free email accounts [5]. As a result of this research they were able to construct a program to defeat both CAPTCHAs with a high success rate (92% and 33% respectively) [5].

Although a full description of their work is beyond the scope of this report, some key areas of their research involved the following:

Gimpy and EZ-Gimpy

The two types of CAPTCHA challenges defeated by Mori and Malik were Gimpy and EZ-Gimpy. Gimpy and EZ-Gimpy are freely available scripts which can be used to generate CAPTCHA challenges. Gimpy works by selecting six or seven words from a dictionary and then rendering all words inside of a distorted image (See Figure 8 below). The user must then correctly identify three of the seven words in order to complete the challenge. EZ-Gimpy is a simplified version written in Perl which uses a single word from the dictionary for the challenge (See Figure 9 below).

Figure 8: An example image created using Gimpy. Notice that words may be repeated inside of the Gimpy image.

Figure 9: An example image created using EZ-Gimpy.

Locating Text in a Cluttered Environment

One of the challenges faced by Mori and Malik was to locate the text within the image itself. As part of their solution, they pre-computed a large number of generalized shape contexts and then compared them to a random sample of points from the image. This technique results in the creation of clusters of matches to form at the location of the text allowing them to quickly prune away other parts of the image [5]. This process determines where to look for text.

Finding Letter Hypotheses

Another challenge was to identify possible letters in the text. They decided to use Canny edge detection, focusing on shape contexts near high values of the texture gradient operator (i.e. areas that do not match the background texture.) By employing a voting scheme based on the pre-computed shape contexts used earlier, they were able to create a set of tuples containing the letter, location and score [5]. This process outlines individual characters and attempts to identify which characters could be represented.

Extracting Candidate Words

In order to find candidate words they took clusters of letter hypotheses and looked for matches that would enforce spatial continuity within words (i.e. letters had to be in order). Using this technique they were able to prune away all but a few words which would then be scored based on the matching cost of individual letters. The score for a word would be the average score for matching each of its letters. Their answer was the word with the best matching score [5]. This process selects the most likely word being represented in the image.

Results

Although, none of the specific techniques used by Mori and Malik were new, their work did provide a general framework which could be applied to a wider range of AI problems. Their

work represents advancement in technology sufficient to render the current implementations of Gimpy and EZ-Gimpy ineffective as screening tools.

Despite the relative success of Mori and Malik in defeating two forms of CAPTCHA, their work has been criticized by some in the AI community. Many express the view that the work of Mori and Malik did not represent advancement in the field of character recognition, but rather the creation of specialized routines for defeating CAPTCHAs [16]. The creation of specialized routines for defeating a CAPTCHA is similar to overtraining a recognizer for a specific test data set. It gives excellent results on the test data but is unable to generalize in a useful way when given real-world data, i.e. data not in the training set.

Contributions of CAPTCHA

In their research Mori and Malik noted that since the source code for generating CAPTCHAs is publically available, they had access to a practically infinite set of test images with which to work [5]. The availability of a large dataset is particularly useful when considering that in most object recognition problems the dataset is limited and it is often difficult to generate many reasonable test images [5].

Another unique aspect of their work involving CAPTCHAs is the adversarial nature of CAPTCHAs themselves. Both CAPTCHAs used in their research were designed to be difficult for computer programs to handle. The CAPTCHA itself takes advantage of known weaknesses in computer recognition, thus in order to defeat the CAPTCHAs they were required to focus their efforts on improving vulnerable aspects of current recognition technology.

Similar work has also been done using audio CAPTCHAs [18].

CHAPTER 5

OPPOSING VIEWS

Although there is evidence that some advancements have been made due to the use of CAPTCHAs, there is no strong consensus among researchers that the expected gains in the field of AI have, or will ever be realized. Some of the criticisms include the following:

Human Tolerance and Accessibility

Since the effort of reading and typing nonsensical words often requires a significant conscious effort by the person answering the CAPTCHA, many users view them as an irrelevant intrusion that is both irritating and threatening [15]. The necessity of creating challenges that will be tolerated by users severely restricts how hard the test can be [16]. Another example from a W3C working group is stated as follows:

> "[CAPTCHA] comes at a huge price to users who are blind, visually impaired or dyslexic. Naturally, this image has no text equivalent accompanying it, as that would make it a giveaway to computerized systems. In many cases, these systems make it impossible for users with certain disabilities to create accounts, write comments, or make purchases on these sites, that is, CAPTCHAs fail to properly recognize users with disabilities as human" [23]

With the necessity of compensating for advances in AI technology by increasing the difficulty of CAPTCHA challenges it is unlikely that users will continue to accept ever more intrusive tests in the future.

Specialization of Recognizers

The underlying hardness of a CAPTCHA is based on the fact that years of research have failed to provide solutions to many AI recognition problems. For many audio, text and image recognition problems there still exists a significant gap between AI and human capabilities. Although it is assumed that a CAPTCHA can only be defeated by making some advancement in AI recognition technology, many individuals have been able to meet the specific challenge posed by CAPTCHA in ways that are not particularly relevant to solving the underlying AI problem [16]. Instead of solving the underlying AI problem, they have created specialized routines that are only useful in breaking CAPTCHAs [16]. This is due largely to the fact that the challenges posed by CAPTCHAs are artificial, and have little basis on the real-world problem [16]. Examples of this include the following:

- Mori & Malik: The work of Mori and Malik applied a generalized framework to defeat Gimpy and EZ-Gimpy CAPTCHA images. Their methods included the exclusive use of Gimpy and EZ-Gimpy images in their training and testing data [5]. The resulting recognizer was not tested on real data or other CAPTCHA systems.
- Breaking PayPal HIP: In 2008 a researcher at the Rochester Institute of Technology created a specialized three-step process (preprocess, segment, classify) for defeating the CAPTCHA used by PayPal. His classifier was trained and tested using only PayPal CAPTCHA images. Details of his work (including Matlab code) are included in his research paper [24].
- PWNTCHA: PWNTCHA is a project dedicated to detect and decode a wide range of text-based CAPTCHA images through the use of specialized recognizers. The project has resulted in the creation of specialized recognizers for PayPal, Xanga, Slashdot and many other sites. An extensive list of defeated systems along with source code can be found at caca.zoy.org.

Useless Answers

Despite the enormous collective cognitive effort provided by millions of individuals solving CAPTCHAs each day, their work is almost immediately discarded once the test is complete [16]. Since the response provided by the user does not provide any direct information on how the user arrived at their answer, the problem is artificial in nature, and the answer has already been determined. Consequently these types of responses are of little use to the AI community [16].

CHAPTER 6

ONGOING AND FUTURE IMPROVEMENTS

The Strengths of CAPTCHA

Regardless of the weaknesses that exist in the current implementation of CAPTCHA there are significant positive aspects upon which future CAPTCHA implementations can build. These aspects include the following:

- Wide Usage: According to the reCAPTCHA website there are over 100,000 websites currently using the reCAPTCHA system including Facebook, Ticketmaster and Craigslist. It is estimated that over 100 million CAPTCHA challenges are solved every 24 hours [22]. CAPTCHAs are widely recognized by internet users and are generally tolerated by them.
- Increased Focus on Research: Ever since the introduction of CAPTCHA it has been a frequent topic in the AI and computer security community. Papers have been written on its use (See [2] and [3]), attempts to analyze its weaknesses (See [5] and [18]) and ideas for improvements and expansion (See [8], [15], [16] and [17]). Much of this work has focused on improving techniques for dealing with distortions in the data.
- Broad Range of Available AI Problems: Up to this point the vast majority of CAPTCHA challenges have been based on character recognition problems. Despite this fact, some CAPTCHA challenges have already started to leverage a wider variety of AI problems including audio recognition, image recognition and cognition. Some CAPTCHA systems such as one used by Facebook have the ability to personalize the challenge to a specific user by having them recognize images of friends.

Real Pattern Recognition

Over the past few years there has been a push to integrate real pattern recognition problems in CAPTCHA challenges as a replacement for the computer generated problems that have been used up to this point. By taking this approach, the security of the CAPTCHA is more tightly coupled with the underlying AI problem itself, and the substantial human resources that go into answering CAPTCHA challenges can be used to train and test new classifiers [16].

One example of this approach includes the following project:

reCAPTCHA

The reCAPTCHA project is currently being used to harness the collective cognitive ability of human users around the world. It is estimated that over 100 million CAPTCHAs are solved each day with an average time for completion of 13.5 seconds [22]. That would amount to over 42

years of human effort every being spent each day. The reCAPTCHA project uses novel techniques to harness this collective human cognitive power in the attempt to digitize old printed material. The project also attempts to improve upon the failures of the original CAPTCHA.

The process used by reCAPTCHA is as follows [22]:

1. A collection of scanned images of old print material are passed through two separate OCR algorithms. The output from both algorithms is compared to one another and to the contents of an English dictionary. Any words that are output differently by both OCR algorithms, or which do not match a word for the dictionary are marked as suspicious and are passed on to the next step in the algorithm. All other text is considered correct.
2. Each suspicious word is placed in an image along with another word for which the correct answer is already known. The two words are then distorted to make computer-based OCR more difficult. The resulting image is then used as a CAPTCHA.
3. The user is asked to identify both words. If the user is able to identify the control word, then it is assumed that the user has also entered the suspicious word correctly and the response is marked as a human vote for that word. If the first three human responses match, but the word was not recognized by either computer OCR algorithm it is marked as correct and becomes a control word.
4. In case of discrepancies between human responses reCAPTCHA enters a process where each OCR response gets half of a vote, and each human response gets a full vote. The suspicious word is then sent out to multiple human readers until one response collects 2.5 votes. For example, if two human responses and one OCR response match it is considered correct, or if three human responses but no OCR responses match it is also considered correct.
5. In order to account for words that are beyond human recognition, a button allows users to request a new pair of words. If six users reject a word before any correct response is accepted the word is rejected as unreadable.
6. The winning human response to each suspicious word is combined with the rest of the original text and becomes part of the finished digitized material.

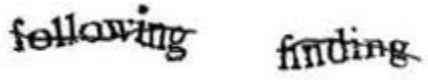

Figure 11: An example image from a reCAPTCHA challenge. Notice the addition of clutter and spatial distortion added to increase the difficulty of automated OCR.

In the first year of its use reCAPTCHA used over 1.2 billion CAPTCHAs to correctly decipher over 440 million suspicious words. As of September 2008 the rate of transcription exceeded 4 million words per day [22]. Currently reCAPTCHA has been used to digitize over 20 years of archives from The New York Times. Popular sites using reCAPTCHA include Craigslist, CNN.com, Facebook, TicketMaster and Twitter.

One of the major benefits involved with reCAPTCHA is that it addresses many of the criticisms involved with the original CAPTCHA including the following:

- Human Tolerance: As part of the study performed in [22] it was found that websites that switched to reCAPTCHA challenges from the original CAPTCHA experienced fewer complaints from their users. The drop in complaints is partially attributed to the fact that users are more willing to accept a system that contributes to the digitization of human knowledge. The use of complete words over random strings may also be a factor.
- Specialization of Recognizers: In reCAPTCHA only words from scanned material upon which both methods of OCR have already failed are used as control words. This method of selection chooses the “hardest” words for computers to decipher [22] and prevents specialized recognizers from being able to take advantage of weaknesses in algorithms that have been used in the past to generate CAPTCHAs.
- Useless Answers: In previous CAPTCHA implementations human responses were almost immediately discarded once submitted by the user. One of the major benefits of reCAPTCHA is its ability to combine millions of responses into the effort of digitizing printed material. This process is helping to digitize over 160 books every day [22].

Despite the major advances that are being made in the reCAPTCHA project there are still some limitations. First, although human acceptance for the new system has risen, users are often presented with scanned images which are beyond human recognition, forcing many to attempt multiple CAPTCHAs. Second, although it is more difficult to create specialized recognizers to defeat a reCAPTCHA, the new requirement of a centralized database opens up new security considerations. For example, an attacker who gains access to the reCAPTCHA database would be able to automate responses to reCAPTCHA challenges by comparing the challenge to images stored in the database. Once reCAPTCHA has been defeated an attacker would then be free to mount automated attacks on the underlying system. Third, despite the fact that responses are being put towards some useful effort, there is still little, if any, information contained in responses that can assist in the direct advancement of AI technology.

Human Cognitive Sciences

Although CAPTCHA does not provide direct information on the human Cognitive process, with close to 200 million CAPTCHAs being solved by humans each day [8] there is potential for

observations to be made. Examples of studies which could be performed with the aid of CAPTCHAs include the following:

- Character VS. Word Recognition: There is currently no consensus on whether human character recognition occurs letter-by-letter or by a word-template model [6]. An empirical study comparing the human success rates when recognizing complete words, compared to random characters could provide additional evidence for cognitive scientists to consider.
- Color: A CAPTCHA challenge could be used to study the effects of color on human recognition. One example would be creating a series of text-based CAPTCHA challenges using colors that are often associated with specific words, such as “Danger” and the color red or “Hazard” and the color yellow. Potentially useful Information could be obtained by comparing the relative success rates of recognition when an associated color is used. Another variation could include the use of color in image recognition. An example of this could include comparing human success rates identifying a pink apple as opposed to one in more natural color.
- Spatial Distortions: Does the relative position, size or orientation of an object have an impact on human recognition? A CAPTCHA challenge could be used to explore the impact of spatial distortions on human recognition.
- Font Usability: By comparing the success rates of CAPTCHA challenges created using text of differing character fonts researchers could gain additional information on the usability of fonts.

To the best of my knowledge no research has yet been done using CAPTCHA challenges to advance research in human cognition.

Linguistic Cognition

Another area that is likely to benefit from CAPTCHA in the future is that of computer linguistic cognition. Linguistic cognition involves the ability to recognize and understand language. Linguistic cognition is a prime candidate for use in CAPTCHA systems due to the large gap between the capabilities of humans and computers and the ability to make systems available to individuals with disabilities by allowing challenges to be presented as text, spoken audio or brail. The fact that language consists of simple constructs allows for automatic creation of false samples [17]. Examples of CAPTCHA type challenges involving linguistic cognition include the following:

- Knock-Knock Jokes: Researchers in Brazil have presented some very interesting work involving human and computer understanding of Knock-Knock jokes [17]. In their work they found that although it is very simple to identify the necessary parts of a Knock-Knock joke, it is very difficult to simulate the complex wordplay required for the joke to make sense. The challenge works by presenting users with a known Knock-Knock joke

and one or more generated by a computer program, the ability of humans to identify which of the jokes actually makes sense allows them to verify that users are human. One example used in their study is as follows:

Click on the funny Knock-Knock Joke!

A: Knock, Knock
B: Who is there?
A: Justin
B: Justin who?
A: Justin woke in the middle of the night

A: Knock, Knock
B: Who is there?
A: Cars
B: Cars who?
A: Cars had been feeling increasingly better every day

A: Knock, Knock
B: Who is there?
A: Kenya
B: Kenya who?
A: Kenya give me a hand **Solution: Can you give me a hand?**

- Task Completion: Another system presented in [16] provides users with a text-based set of instructions ranging from “Click on the point of the left mountain peak” to “Type the second to last word in this sentence into the provided text box.” This system is based on the ability of humans to understand written instructions and perform appropriate tasks.

CHAPTER 7

CONCLUSION

In conclusion, I have found that the basis upon which CAPTCHA has been presented as a catalyst for advancements in the field of AI is correct. In presenting this conclusion, I do not wish to overstate its impact, or gloss over the many problems that have and do exist. I have based my conclusion in part on the following:

- The vast amount or research that has already been done: While conducting my research for this project I was able to tap into a vast collection of scholarly work relating to CAPTCHA. The quantity, quality and availability of information on this topic represents a considerable amount of time and effort dedicated to the creation of a better CAPTCHA, and exploration of the underlying AI problems.
- Specialized recognizers: The existence and sophistication of specialized recognizers designed to defeat CAPTCHA demonstrates a concerted effort by researchers to devise AI solutions to CAPTCHA challenges. As projects such as reCAPTCHA move towards the use of real world data for which current technology has already failed, the creation of specialized recognizers will become more relevant to producing advances in the AI community.
- Widespread use: Advances such as reCAPTCHA are starting to make better use of the millions of responses generated each day. In its current form, the reCAPTCHA project uses the cognitive efforts of people around the world to not only digitize books, but also to create a database of problems for which our current AI technology fails. In my experience, it is the solutions that fail which provide the greatest insight into what needs to be done to improve the system. Although the methods used may be the same, having this type of training data could be very useful.

Based on the work that has already been done I would expect the greatest future advancements to be made on AI problems that meet the following criteria:

- Wide performance gap: AI problems such as identifying objects in images, sounds and cognition stand to benefit from the large performance gap between humans and current technology.
- Large collection of existing data: AI problems such as character and image recognition which have large quantities of existing test data are most likely to benefit from recent advances in CAPTCHA.
- Problems for which there is a single consensus solution: It is unlikely that problems that do not have consensus solutions will benefit from advances in CAPTCHA. These types of problem involve deciding issues of morality, politics or preference.

As I have explored the possibilities surrounding the use of CAPTCHA I have come to see it as more than a simple human-interactive proof. Projects such as reCAPTCHA have already demonstrated that CAPTCHA challenges can be designed to ask specific questions and obtain accurate answers. The beauty of CAPTCHA is that with a little creativity researchers can create challenges that test existing theories on human nature and probe the way in which we think. Perhaps the greatest scientific benefits of CAPTCHA will come not from technological advancements made in the effort to defeat it, but it's unique potential to ask individuals around the world more than simply "Are you human?"

BIBLIOGRAPHY

[1] Ahn, L., Blum, M., Hooper, N.J., and Langford, J. CAPTCHA: Telling Humans and Computers Apart. 2003. http://www.captcha.net/captcha_crypt.pdf. October 2010.

[2] Xu, J., Lipton, R., and Essa, I. Hello, Are You Human? Technical Report, GIT-CC-00028, Georgia Institute of Technology, 2000.

[3] Pinkas, B., and Sander, T. Securing Passwords Against Dictionary Attacks*. In Proceedings of the ACM Computer and Security Conference*, 2002.

[4] Naor, M. Verification of a Human in the Loop or Identification via the Turning Test. 1996. http://www.wisdom.weizmann.ac.il/\~naor/PAPERS/human.ps. October 2010.

[5] Mori, G., and Malik, J. Recognizing Objects in Adversarial Clutter – Breaking a Visual CAPTCHA. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2003.

[6] Coates, A., Baird, H., and Fateman, R. Pessimal Print: A Reverse Turing Test. *In Proceedings of the International Conference on Document Analysis and Recognition*. 2001

[7] Ahn, L., Blum, M., and Langford, J. Telling Humans and Computers Apart Automatically. *In Communications of the ACM*, February 2004.

[8] What is ReCAPTCHA? 2010. http://www.google.com/recaptcha/learnmore. December 2010.

[9] Pinkas, B., and Sander T. Securing Passwords Against Dictionary Attacks*. In Proceedings of the 9th Conference on Computer and Communications Security*, November 2002.

[10] Odlyzko, A. The Future of Integer Factorization, 1995.

[11] Brillhart, J., and Selfridge, J. Some Factorizations of $2^{\square} \pm 1$ and Related Results. *In Mathematics of Computation*, 1967.

[12] Lenstra, A. Jr., Manasse, M., and Pollard, J. The Number Field Sieve. *In Proceedings of the 22nd Annual ACM Symposium on Theory of Computing.* 1990.

[13] General Purpose Factoring Records. 2010. http//www.crypto-world.com/FactorRecords.html. December 2010.

[14] Kleinjung, T., Aoki, K., Franke, J., Lenstra, A. Thom'e, E., Bos, W., Gaudry, P., Kruppa, A., Montgomery, L., Osvik, A., Riele, H., Timofeev, A., and Zimmerman, P. Factorization of a 768-bit RSA modulus. *In Eprint archive no. 2010/006.* 2010.

[15] Baird, H., and Bently, J. Implicit CAPTCHAs. *In Proceedings of Document Recognition and Retrieval XII.* January 2005.

[16] Lopresti, D. Leveraging the CAPTCHA problem. *In Proceedings of the Second International Workshop on Human Interactive Proofs*. May 2005.

[17] Ximenes, P., Santos, A., Fernandez, M., and Celsti, J. A CAPTCHA in the Text Domain. *In On the Move to Meaningful Internet Systems.* 2006.

[18] Tam, J., Simsa, J., Hyde, S., and Ahn, L. Breaking Audio CAPTCHAs. *In Advances in Neural Information Processing.* 2008.

[19] Albert, D., Schrepp, M., and Held, T. Construction of Knowledge Spaces for Problem Solving in Chess.

[20] Campbell, M. Knowledge Discovery in Deep Blue. *In Communications of the ACM*. November 1999.

[21] Shannon, C. Programming a Computer for Playing Chess. *In Philosophical Magazine*. March 1950.

[22] Ahn, L., Maurer, B., McMillen, C., Abraham, B., and Blum, M. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *In Science Magazine*. September 2008.

[23] Inaccessibility of CAPTCHA, Alternatives to Visual Turning Tests on the Web. W3C Working Group Note 23 November 2005.

[24] Kluever, K. Breaking the PayPal HIP: A Comparison of Classifiers. Rochester Institute of Technology Document and Pattern Recognition Lab. May 2008.