A CAPTCHA in the Text Domain

Pablo Ximenes[1], André dos Santos[1], Marcial Fernandez[2], and Joaquim Celestino Jr.[2]

[1] UPRM – Mayagüez Campus (RUM) – ZIP: 00680 – Maygüez – PR – USA
[2] Av. Paranjana, 1700 – CEP: 60.740-000 – Fortaleza – CE – Brazil
pablo@ximenes.info, andre.moura@ece.uprm.edu,
{marcial, celestino}@larces.uece.br

Abstract. Research on CAPTCHA has led CAPTCHA design into adopting almost exclusively graphical implementations that deal mostly with character recognition. This has reached an exhaustion point, where new approaches are vital to the survival of the technique. This paper discusses the early stages of a research that intends to solve the open problem of a CAPTCHA in the text domain offering, this way, innovative research possibilities to the CAPTCHA paradigm. It is essentially an investigation on a CAPTCHA that draws its security from the cognitive and computational aspects behind phonetic punning riddles found on Knock-Knock Jokes. By the specification of a computational model, the implementation of a prototype and its experimentation with human individuals, it is shown that the proposal is indeed feasible and that studies in non conventional areas for Information Security are the key for developing the proposed goal.

Keywords: Security, CAPTCHA, Text Domain, Natural Language, Computational Humor.

1 Introduction

An important class of attacks within the internet requires the use of automated procedures to acquire access, privileges, or to exhaust resources of a system. For example, attacks that have as a goal to harvest free e-mail accounts require the use of automated tools to be able to register a large number of fabricated users.

One of the main strategies in defeating general automation based attacks is the CAPTCHA [1] paradigm. This technique implements a type of filter to computational systems that allows human access while denying service to computational robots, thus preventing automated attacks from taking place. In order to achieve that, CAPTCHAs explore several problems within the field of Artificial Intelligence (AI). One could argue that, because of their AI related characteristics, CAPTCHA systems would range within several interesting types of implementations, each one exploring different areas of AI. This does not happen, though. In fact, CAPTCHA implementations are mainly of graphical nature and deal mostly with pattern recognition, most specially with character recognition. This lack of innovative tracks has led CAPTCHA research into an exhaustion point, where new approaches are not only necessary but vital to the survival of the technique.

R. Meersman, Z. Tari, P. Herrero et al. (Eds.): OTM Workshops 2006, LNCS 4277, pp. 605 – 615, 2006.

One interesting and innovative approach, regarded as an important open problem by the very CAPTCHA research community, is the problem of a CAPTCHA in the text domain. A CAPTCHA of this sort would draw its inner workings from AI problems that could be found within text constructs and would require only plain text to be assembled. Besides granting new possibilities to the CAPTCHA paradigm, a CAPTCHA in the text domain would be specially suitable for devices with low accessibility capabilities and for visually impaired users, since it would not require any advanced graphical screens and no multimedia features.

This way, in order to alleviate the forementioned problems, we propose a novel type of CAPTCHA that functions within the text domain. This paper describes the early stages of our research on the development of such CAPTCHA. In essence, the proposed CAPTCHA draws its security from the cognitive and computational aspects related to phonetic punning riddles found on Knock-Knock Jokes and similar structures. By the specification of a computational model, the implementation of a prototype and its experimentation with human individuals, it is shown that the proposal is indeed feasible and that studies in non conventional areas for Information Security are the key for developing the proposed goal.

This paper is organized as follows: section 2 introduces some basic definitions concerning automation based attacks that will be used within this paper; section 3 talks about the history of automated human verification and presents the CAPTCHA paradigm; section 4 points out the problems of current CAPTCHA research regarding and presents the problem of a CAPTCHA in the text domain; section 5 outlines the core of our proposal and presents our prototype; section 6 explains the experiment performed with the prototype and discusses our findings; and finally in section 7 some conclusion are drawn.

2 Automated Attacks

Automation based attacks are those which do not actually violate a specific security rule; they simply use (or misuse) a legitimate system in a super-human way, performing requests to the system repeatedly and in a high rate, in order to achieve some objective that goes against the initial goals of the system's designer. Some computer systems are designed with the assumption that only humans will use them, factoring human limitations as a part of the system's security policy. This way, such computer systems do not worry about imposing restrictions on, for example, an excessive high rate of requests per second, because this is simply not humanly possible. The problem happens when super-humans, or computer software robots, enter in action. They can repeat the same request several times every second, disrupting the original intent of the vulnerable computer system. Thus, a simple electronic mailing server, while being under an automation based attack, may become a disgusting SPAM relayer.

Stopping automation based attacks within the Internet is a growing trend. This effort has been done basically within two fronts. One approach tries to defeat the mechanisms and techniques that originate the attack, in attempt to stop its very source. An example to that would be a system that, as soon as a super-human use (or automated attack) is identified, it would deny service to the source of the use. The problem with that approach is that Internet access techniques such as NAT and proxy servers are becoming quite common. Thus, legitimate human users can suffer service denial, as they can easily be misrepresented as attackers. This approach also fails for distributed attacks, which make use of several sources, real and/or fake, such as Distributed Denial of Service (DDoS). It is currently hard, if not unfeasible, to determine the source of such attacks, and service denial to human users would be a problem again. Another problem regarding this approach exists in SPAM filtering, where legitimate emails are sometimes labeled as SPAM. The other approach to counter automated attack works by changing the design of vulnerable systems in a way that potential misuse by automation based attackers is acknowledged as part of the design. This approach is fulfilled with proposals that have as main idea to identify whether or not the entity that is using the system is an authorized party, which means being human. Since systems that are vulnerable to automation based attacks are so due to the fact that they rely on humanity as a characteristic of their users, a protection technique of this sort would have to concentrate on pointing out non-human users, or computational robots, and deny service to them. If a system can be certain that it is being used by human users, high usage, which in other situations would be considered an attack, would not constitute a problem; and if the system knows it is being used by a computer, even during low usage, service denial would take place. Thus, the key to stop automation based attacks is the proper identification of humanity.

3 Turing Tests

Identifying humanity is a complex and long-lasting task. It dates back from the beginnings of modern computer science when Alan Turing presented his theories on the possibility of thinking machines in his famous article “Computing Machinery and Intelligence” [8]. There, Turing proposes his so-called “Imitation Game” (later known as Turing Test), where a human individual would have to interrogate two hidden entities and try to discover their nature concerning humanity. One of the entities would be a human being and the other would be a computer program. Through a series of indirect questions using a computer interface, the human interrogator would have to determine which one was each. This way, the Turing Test was the first test intended to identify humanity within a computational environment.

3.1 Human in the Loop

In 1996, based on Turing’s ideas, Moni Naor proposed a theoretical framework that would serve as the first approach in testing humanity by automated means [11]. In Naor’s humanity test, the human interrogator from the original Turing Test was substituted by a computer program. The original goal of his proposal was to present a scheme that would discourage computer software robots from misusing services originally intended to be used by humans only, much in the same sense of stopping an automation based attack though human identification. Basically, he proposed an adaptation of the way identification is handled in cryptographic settings to deal with this situation. There, when one party A wants to prove its identity to another party B, the process is a proof that A can effectively compute a (keyed) function that a different user (not having the key) cannot compute. The identification process consists of a challenge selected by B and the response computed by A. What would replace the keyed cryptographic function in the proposed setting would be a task where humans excel, but machines have a hard-time competing with the performance of a three-years-old. By successfully performing such task the user proves that he or she is human.

3.2 CAPTCHAS

Later, Naor's ideas were the basis for a more complete and thorough work on the subject of automated Turing Tests, called then the CAPTCHA paradigm, which was a successful formalization and substantiation of Naor's conceptual model, done by Luis von Ahn et al [1], known as CAPCTHA.

CAPTCHA stands for Completely Automated and Public Turing Test to Tell Computers and Humans Apart. Even though the name itself is self explanatory, some remarks are yet necessary.

Besides formalizing Naor's ideas, von Ahn's work discriminated the important characteristics of an automated Turing Test, leaving aside some of the original concepts that were unnecessary.

As Hard AI problems used by CAPTCHA systems must also be easy for humans to solve, they are generally related to aspects of human cognition. Examples of such problems are optical character recognition (OCR), audio recognition, natural language processing, and image recognition. The same problems for a human being would be, respectively, reading text in images, listening to text in audio samples, understanding the meaning of a text excerpt, and understanding and/or identifying an image sample. It is evident that a human being would have no problems solving those problems, as for a computer program this would not be a trivial task.

Even posing as a difficult task, attackers and security analysts are always trying to find new forms of breaking CAPTCHAs. Be it for self protection or malicious reasons, CAPTCHA systems are constantly subject of attacks and studies that aim to disrupt their efficacy [5,14]. As all the strength of CAPTCHA systems is dependent only on Hard AI problems, breaking a CAPTCHA, in a final analysis, would mean pushing the AI community solving capabilities further ahead. Much in the same sense Naor foretold, the attacker-protector model which is very common in the Information Security field works for CAPTCHAs as a win-win situation, where breaking the system does not only imply a system weakness, but contributes with computer science as a whole improving techniques from other fields.

Therefore, because of its strong formal foundations, the CAPTCHA scheme is the leading research paradigm on automated Turing Tests.

4 Trouble in Paradise

Even though the CAPTCHA framework has a strong formalization and several empirical evidences, CAPTCHA implementations are being extensively broken [5,13,14,16]. Some part of this phenomena falls into von Ahn's objective of improving AI, but a considerable part is simply a direct result of the exhaustion of CAPTCHA's graphical based model .

Instead of searching for alternate means to explore human cognition, CAPTCHA researchers focus only on basic human senses, such as hearing and specially vision, mainly because they are simple and well known. This tends to push CAPTCHA research onto proposals that mostly try to explore graphical tests, while other possibilities remain open problems, such as the problem of a CAPTCHAs in the text domain.

A CAPTCHA in the text domain (or text based CAPTCHA) would mainly explore linguistic cognition aspects of humans, or the ability humans have to understand linguistic constructs. The construction of a CAPTCHA in the text domain is often cited as an important open problem [1,4,6]. To our knowledge, the only formal attempts to construct a CAPTCHA of this sort are [4] and [15], but they all fail to address the issue.

In [4], a word from a piece of text taken from a data source of human-written text is randomly selected and substituted by another word selected at random, in the hope that it would be easy for humans to pick that word (because it didn't fit in the context), but difficult for computers. However, it is demonstrated also in [4] that it was possible to write a program that had considerable success-rates in "cheating"" the test by taking into account statistical characteristics of natural language.

In [15], it is proposed the use of lexical semantics to construct an HIP that draws its security from the problem of word-sense ambiguity, i.e., the phenomenon that a single word can have different meanings and that different words can have the same meaning, depending on the context in which a word is used. Despite the fact that indeed this HIP proposes a task difficult for computers and easy for humans, it violates Kerckhoff's principle [12] that is present in the CAPTCHA paradigm, as the efficacy of the test is based on the secrecy of the database that holds the "secret annotations", which are mappings necessary for the disambiguation process, which is all it is necessary to solve the test. Furthermore, it is not very clear how this database would be constructed and the author only indicates that it is necessarily constructed with human intervention possibly creating a barrier for automating the test.

5 A CAPTCHA in the Text Domain

5.1 Proposal

Through a general overview of some possibilities for the deployment of our CAPTCHA, we decided to concentrate on a particular work by Julian Taylor [9]. She has studied automated (computational) generation and recognition of humorous constructs on the focused domain of Knock-Knock (KK) jokes. A KK joke is basically a type of humorous punning (wordplay) riddle. A regular KK joke is a dialog between two people that uses wordplay in the punch line. A KK joke can be summarized using the following structure:

Line1: "Knock, Knock"
Line2: "Who is there?"

Line3: any phrase
Line4: Line3 followed by "who?"
Line5: One or several sentences containing one of the following:
Type1: Line3
Type2: a wordplay on Line3
Type3: a meaningful response to Line3.

KK jokes are more common in the English speaking world, though its structure provided us with important characteristics that we believe may assist in our goals. These are:

1. A KK joke is a linguistic construct that by its humorous nature becomes easily recognizable and sometimes enjoyable.

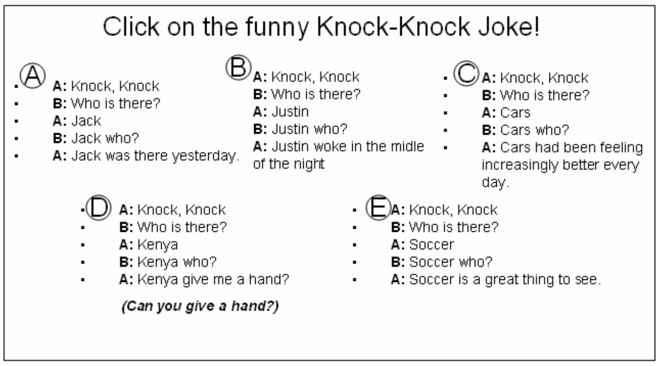
2. Despite the fact KK jokes are not cultural available worldwide, one may argue that phonetic punning riddles are.

3. A regular KK joke is a simple and stable structure, with a formation rule.

4. KK jokes are based on phonetic punning riddles as they explore cognitive aspects not only related to linguistics, but also to sound interpretation. This way, we empower ourselves with more tools in order to build the proposed CAPTCHA.

5. There is evidence of an incongruity between computation KK joke generation and computational KK joke understanding. This conclusion was drawn by some of the remarks found on Taylor's work on KK Jokes. She was able to build a successful KK joke generator, but was not capable of building a KK joke recognizer that could in fact do its job. Despite her efforts, Taylor's KK Joke recognizer could only find wordplays, but was unable to determine if the joke made sense or not. She justifies that by stating that the creation of KK jokes requires restrict knowledge, whereas their understanding requires "world" knowledge. We believe this gap is sufficient enough to generate humorous text excerpts based on the KK joke structure that computers will not "get".

Our proposed scheme consists basically of a challenge that would present a set of KK Joke like structures to the user. Despite all of the presented structures would be built upon the same general linguistic structure, only one of them would make sense as a real KK joke. The user would have to indentify the correct joke within the set in order to prove his human condition. Therefore, our proposed CAPTCHA concentrates



Click on the funny Knock-Knock Joke!

Ⓐ
- **A:** Knock, Knock
- **B:** Who is there?
- **A:** Jack
- **B:** Jack who?
- **A:** Jack was there yesterday.

Ⓑ
A: Knock, Knock
B: Who is there?
A: Justin
B: Justin who?
A: Justin woke in the midle 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?

(Can you give a hand?)

Ⓔ
- **A:** Knock, Knock
- **B:** Who is there?
- **A:** Soccer
- **B:** Soccer who?
- **A:** Soccer is a great thing to see.

Fig. 1. Example of the proposed scheme

on a test that would generate fake jokes following the same structure of a regular KK Joke (or a variation of it), together with a real KK joke. By correctly identifying the real KK Joke, a human individual would be set apart from computational robots. Figure 1 shows an example of the proposed CAPTCHA, for the English language.

5.2 The Prototype

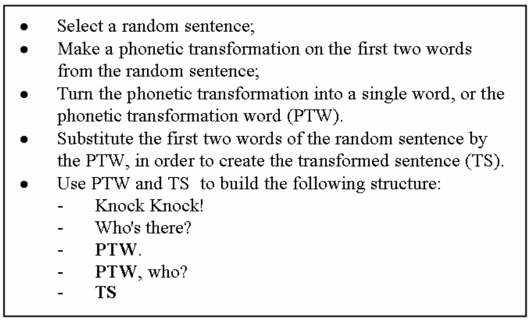
In order to understand human cognitive impacts of an automatically generated KK Joke and to create the foundations for and at the same time improve our model, we have developed a basic prototype for our CAPTCHA system.

The prototype consisted basically of a double challenge test, where a user was required to pin point the non-bogus KK joke among a given set. Each challenge presents one real KK joke and two fake KK jokes. By successfully identifying the non-fake KK joke on each one of the two challenges, the user proves that he or she is human.

To strengthen our contend that internationalization would not be an issue to the development of our proposal and for the obvious reason of being mostly a Brazilian research team, we have developed our prototype in Portuguese. Therefore, we have coined a variation of a regular KK joke, since there is no such structure in the Portuguese language. It is basically a simpler KK joke based structure in Portuguese, as follows:

Line1: "Você conhece <wordplay>?"
Line2: "<wordplay> quem?
Line3: a meaningful response starting by <wordplay>
The translation for that would be:
Line1: "Do you know <wordplay>?"
Line2: "<wordplay> who?
Line3: a meaningful response starting by <wordplay>

The reason we did not performed a literal translation of the same structure found on a regular KK Joje is that we believe the KK joke structure is too long. This doe not constitute a problem for native speakers of English since cultural background helps in



- Select a random sentence;
- Make a phonetic transformation on the first two words from the random sentence;
- Turn the phonetic transformation into a single word, or the phonetic transformation word (PTW).
- Substitute the first two words of the random sentence by the PTW, in order to create the transformed sentence (TS).
- Use PTW and TS to build the following structure:
  - Knock Knock!
  - Who's there?
  - **PTW**.
  - **PTW**, who?
  - **TS**

Fig. 2. Real Joke Generation Algorithm

attenuating this problem. Since our translation could not count on any cultural background from the users it had to be simpler.

The KK joke generation process explored phonetic puns (wordplays) as a means of creating meaningful jokes. The generation algorithm of a real KK joke is summarized by the steps in figure 2.

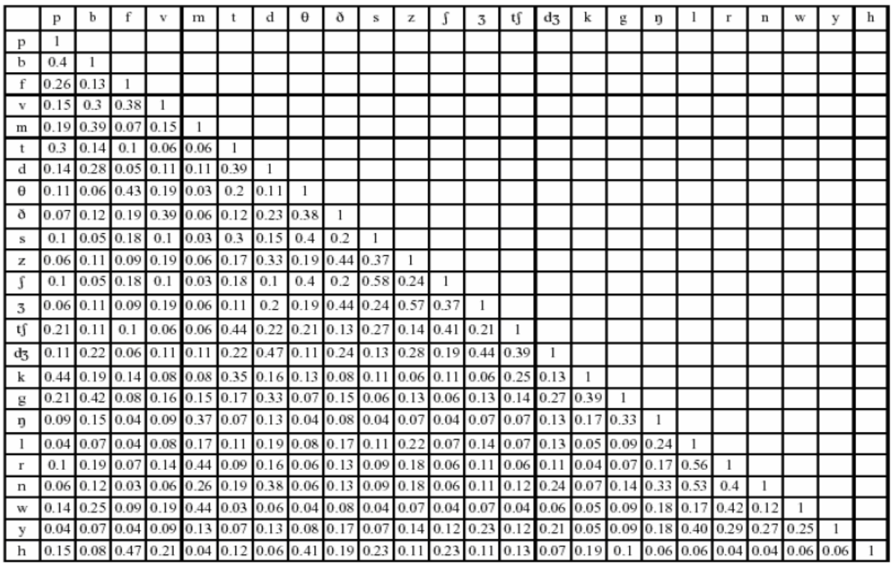
On the other hand, the generation algorithm for a fake KK joke is summarized in figure 3.

- Generate two non-fake KK jokes;
- Exchange their **PTWs;**
- Randomly pick one of changed jokes;
- Exclude the other;

Fig. 3. Fake Joke Generation Algorithm

The phonetic transformation process used in both algorithms followed some of the basic ideas proposed by Taylor. Basically it consists of a process that changes consonantal phonemes by similar ones, based on the phoneme similarity table proposed by [10], that can be found in figure 4.

The resulting structure is a sentence that resembles its original version, mainly because sound similarities. As our prototype was designed for Portuguese language, we have adapted it to use the phoneme similarity table in such a way that only phonemes found on Portuguese language would be considered.



	p	b	f	v	m	t	d	θ	ð	s	z	ʃ	ʒ	tʃ	dʒ	k	g	ŋ	l	r	n	w	y	h
p	1																							
b	0.4	1																						
f	0.26	0.13	1																					
v	0.15	0.3	0.38	1																				
m	0.19	0.39	0.07	0.15	1																			
t	0.3	0.14	0.1	0.06	0.06	1																		
d	0.14	0.28	0.05	0.11	0.11	0.39	1																	
θ	0.11	0.06	0.43	0.19	0.03	0.2	0.11	1																
ð	0.07	0.12	0.19	0.39	0.06	0.12	0.23	0.38	1															
s	0.1	0.05	0.18	0.1	0.03	0.3	0.15	0.4	0.2	1														
z	0.06	0.11	0.09	0.19	0.06	0.17	0.33	0.19	0.44	0.37	1													
ʃ	0.1	0.05	0.18	0.1	0.03	0.18	0.1	0.4	0.2	0.58	0.24	1												
ʒ	0.06	0.11	0.09	0.19	0.06	0.11	0.2	0.19	0.44	0.24	0.57	0.37	1											
tʃ	0.21	0.11	0.1	0.06	0.06	0.44	0.22	0.21	0.13	0.27	0.14	0.41	0.21	1										
dʒ	0.11	0.22	0.06	0.11	0.11	0.22	0.47	0.11	0.24	0.13	0.28	0.19	0.44	0.39	1									
k	0.44	0.19	0.14	0.08	0.08	0.35	0.16	0.13	0.08	0.11	0.06	0.11	0.06	0.25	0.13	1								
g	0.21	0.42	0.08	0.16	0.15	0.17	0.33	0.07	0.15	0.06	0.13	0.06	0.13	0.14	0.27	0.39	1							
ŋ	0.09	0.15	0.04	0.09	0.37	0.07	0.13	0.04	0.08	0.04	0.07	0.04	0.07	0.07	0.13	0.17	0.33	1						
l	0.04	0.07	0.04	0.08	0.17	0.11	0.19	0.08	0.17	0.11	0.22	0.07	0.14	0.07	0.13	0.05	0.09	0.24	1					
r	0.1	0.19	0.07	0.14	0.44	0.09	0.16	0.06	0.13	0.09	0.18	0.06	0.11	0.06	0.11	0.04	0.07	0.17	0.56	1				
n	0.06	0.12	0.03	0.06	0.26	0.19	0.38	0.06	0.13	0.09	0.18	0.06	0.11	0.12	0.24	0.07	0.14	0.33	0.53	0.4	1			
w	0.14	0.25	0.09	0.19	0.44	0.03	0.06	0.04	0.08	0.04	0.07	0.04	0.07	0.04	0.06	0.05	0.09	0.18	0.17	0.42	0.12	1		
y	0.04	0.07	0.04	0.09	0.13	0.07	0.13	0.08	0.17	0.07	0.14	0.12	0.23	0.12	0.21	0.05	0.09	0.18	0.40	0.29	0.27	0.25	1	
h	0.15	0.08	0.47	0.21	0.04	0.12	0.06	0.41	0.19	0.23	0.11	0.23	0.11	0.13	0.07	0.19	0.1	0.06	0.06	0.04	0.04	0.06	0.06	1

Fig. 4. Phoneme Similarity Table

6 Experiment

6.1 Trial

In order to experiment with our CAPTCHA prototype, we have developed a free SMS messaging WEB site that used our prototype. The site offered free SMS messaging to the main Brazilian cell phone operators, including one that normally charges for this service, even via web. We believe this was just enough incentive in a way that would not compel users to forcedly use our system.

We have run the prototype for two days. During this period, a total of 584 users came to have contact with the system, but only 455 of them actually tried to use it at least once. During the experiment a total of 894 tests were performed. Taking into consideration that our prototype presented a double challenge test, another form of analysis would be to consider each challenge alone. The total of single challenges was 1893. A total of 221 tests were answered correctly, which equals 24.72% of the total of answered tests. Analyzing by the challenges point of view, a total of 887 were answered correctly (46.86% of the amount of answered challenges). At first this seems a little discouraging, but after further analysis we noticed that some challenges were just answered too fast, some even in approximately 0 seconds. If one considers that each challenge is presented with three text excerpts, it is possible to argue that a fast glimpse at each one of them would require at least 2 or 3 seconds and the whole challenge would require at least 6 seconds to be answered. We believe this anomaly happened due to some attempts in using a computational robot to break our system (which we further confirmed to be true).

This way, we decided to filter the results by answering time. This led to new and less discouraging results which we summarized in table 1 for complete tests and table 2 for challenges alone.

Table 1. Success percentages for complete tests

Minimun Time Spent	Total of Answered Tests	Total of Correct Answers	Success Percentage
0 seconds	894	221	24,72%
5 seconds	550	181	32,9%
15 seconds	443	147	33,18 %

Table 2. Success percentages for single challenges

Minimun Time Spent	Total of Answered Tests	Total of Correct Answers	Success Percentage
0 seconds	1893	887	46,86%
5 seconds	1258	677	53,81%
15 seconds	1112	602	54,11 %

7 Conclusions

This paper has shown an innovative approach in CAPTCHA research by presenting a strong proposal for a CAPTCHA in the text domain. Though yet inconclusive, our results indicate that some generated KK jokes share some particular characteristics that permit humans to point them out as real jokes. A random guess in our experiment would generate the probability of 11,11% of success for a complete test and 33,33% of success for challenges alone. Taking into consideration that the best results for challenges were 54,11% of success, there is an advantage 20,78% of success over random chance. If we analyze complete test this gap is even bigger, equaling 22,07% (33,18% - 11,11%). This is specially encouraging, mainly because our prototype is yet in its first version where several issues may still be perfected and new concepts are yet to be incorporated. Taking all these factors into consideration, our findings indicate a real feasibility of building a CAPCTHA of the proposed sort.

Acknowledgments. This research was sponsored by UOL (www.uol.com.br), through its UOL Bolsa Pesquisa program, process number 200503311809.

References

1. Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. CAPTCHA: using hard ai problems for security. In Advances in Cryptology, Eurocrypt 2003, volume 2656 of Springer Lecture Notes in Computer Science, pages 294–311, May 2003.
2. Tsz-Yan Chan. Using a text-to-speech synthesizer to generate a reverse turing test. In Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence, page 226. IEEE Computer Society, 2003.
3. Allison L. Coates and Richard J. Fateman. Pessimal print: A reverse turing test. In Sixth International Conference on Document Analysis and Recognition (ICDAR'01), 2001.
4. Philip Brighten Godfrey. Text-based CAPTCHA algorithms. In First Workshop on Human Interactive Proofs, 2002. Unpublished Manuscript. Available electronically: http://www.aladdin.cs.cmu.edu/hips/events/abs/godfreyb_abstract.pdf.
5. Greg Mori and Jitendra Malik. Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA. In Conference on Computer Vision and Pattern Recognition (CVPR '03), volume I, 2003.
6. Bartosz Przydatek. On the (im)possibility of a text-only CAPCHA. In First Workshop on Human Interactive Proofs, 2002. Unpublished Abstract. Available electronically: http://www.aladdin.cs.cmu.edu/hips/events/abs/bartosz_abstract.pdf.
7. Graeme Ritchie "Prospects for Computational Humour," Proceedings of 7th IEEE International Workshop on Robot and Human Communication, Takamatsu, Japan, pp. 283-291, 1998
8. Alan M. Turing. Computing machinery and intelligence. Mind, 49:433–460, 1950. 19. Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. Hips. http://www.aladdin.cs.cmu.edu/hips/.
9. Julia Taylor, "Computational Recognition of Humor in a Focused Domain", Master Thesis, University of Cincinnati, 2004.
10. Stefan Frisch, "Similarity And Frequency In Phonology", Doctoral dissertation, Northwestern University, 1996

11. Moni Naor. Veri_cation of a human in the loop or Identi_cation via the Turing Test. Unpublished Manuscript, 1997. Available electronically: http://www.wisdom.weizmann.ac.il/~naor/PAPERS/human.ps.
12. Auguste Kerckhoffs, La cryptographie militaire, Journal des sciences militaires, vol. IX, pp. 5–83, Jan. 1883, pp. 161–191, Feb. 1883.
13. Chellapilla K., and Simard P., "Using Machine Learning to Break Visual Human Interaction Proofs (HIPs)," Advances in Neural Information Processing Systems 17, Neural Information Processing Systems, MIT Press, 2004
14. Gabriel Moy, Nathan Jones, Curt Harkless, and Randall Potter Distortion Estimation Techniques in Solving Visual CAPTCHAs Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2004
15. Richard Bergmair and Stefan Katzenbeisser, "Towards Human Interactive Proofs in the Text-Domain Using the Problem of Sense-Ambiguity for Security" 7th International Conference, ISC 2004, Palo Alto, CA, USA, September 27-29, 2004
16. The OCR Reaearch Team, "Weak CAPTCHAs", 2006 available online in: http://ocr-research.org.ua/list.html