2014 IEEE International Conference on Systems, Man, and Cybernetics
October 5-8, 2014, San Diego, CA, USA

Vulnerability of the Conventional Accessible CAPTCHA used by the White House and an Alternative Approach for Visually Impaired People

Michitomo Yamaguchi
Department of Mathematical Modeling, Analysis and Simulation, Graduate School of Advanced Mathematical Sciences, Meiji University, Tokyo, #164-8525 Japan
Email: yama3san@meiji.ac.jp

Toru Nakata
Research Institute for Secure Systems, National Institute of Advanced Industrial Science and Technology, Ibaraki, #305-8568 Japan
Email: toru-nakata@aist.go.jp

Hajime Watanabe
Research Institute for Secure Systems, National Institute of Advanced Industrial Science and Technology, Ibaraki, #305-8568 Japan
Email: h-watanabe@aist.go.jp

Takeshi Okamoto
Graduate School of Technology and Sciences, Tsukuba University of Technology, Ibaraki, #305-8521 Japan
Email: ken@cs.k.tsukuba-tech.ac.jp

Hiroaki Kikuchi
Department of Mathematical Modeling, Analysis and Simulation, Graduate School of Advanced Mathematical Sciences, Meiji University, Tokyo, #164-8525 Japan
Email: kikn@meiji.ac.jp

***Abstract*—Many people with visual impairments complain about the poor accessibility of conventional CAPTCHA systems because the audio-style test is too difficult for humans. Even a U.S. governmental site, the "We the People" public website, was criticized for the same reason, and thus it implemented a more accessible quiz-based CAPTCHA system. However, this system is vulnerable to simple heuristics. In this study, we demonstrate the insecurity of this type of CAPTCHA system. We demonstrate that our solver program can beat the CAPTCHA with a success rate of over 99%. In addition, we propose a new verbal-style system to replace the quiz-based CAPTCHA. Our system synthesizes several sentences, which have different degrees of naturalness in terms of their contextual meaning, from a set of source documents using a flexible-order Markov chain. Only human users can perceive the difference in the semantics and select the most (or the least) meaningful option correctly. This test is implemented in a verbal style, which means that it is universally suitable for any type of perceptual channel. We implemented our proposed scheme and analyzed its security based on experiments.**

***Keywords*—accessibility, CAPTCHA, verbal style, vulnerability, We the People**

I. Introduction

A. Problems of Conventional CAPTCHA Systems

Variations on the completely automated public Turing test to tell computers and humans apart (CAPTCHA) system [1] are used widely to differentiate humans from malicious software agents. However, a major social problem is that visually impaired users cannot access the most commonly used systems, which challenge the user to read distorted letters. Several sites use "audio" CAPTCHAs to avoid the visual restriction, but some researchers [2], [3], [4] have shown that state-of-the-art audio CAPTCHAs are still too unclear and difficult for them. For example, the *We the People* [5] website of the White House was criticized by the National Federation of the Blind in the USA [6].

We claim that every CAPTCHA system should satisfy the following requirements.

- *Accessibility:* CAPTCHA systems should not employ tests that are restricted to specific perceptual channels.
- *Correctness:* The tests must distinguish humans from software agents.
- *Uniqueness:* The systems must generate brand new tests without limitations in terms of volume.

B. Our Contributions

First, we show that the quiz-based CAPTCHA system (Fig. 1) used by the *We the People* site (we refer to this as WtP-CAPTCHA), which addressed the criticisms such as [6], has severe vulnerability issues. The WtP-CAPTCHA algorithm only generates questions where the sentence structure is well defined, and thus software agents can solve them using routine procedures. Indeed, our attack program beat the system with a success rate of over 99%. This may have a major impact

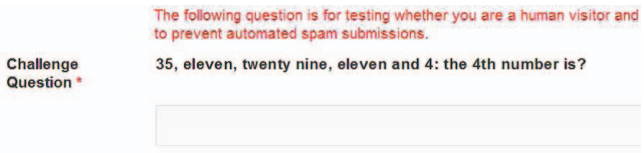

Fig. 1: Example of the CAPTCHA Used by the *We the People*.

978-1-4799-3840-7/14/$31.00 ©2014 IEEE

because this site is managed by governmental officials, and it provides an e-petition system.

Second, to overcome the vulnerability, we propose an alternative verbally based CAPTCHA system, and we analyze its security. The basic idea is similar to that described in our previous study [7]. Our previous system has the following features.

- Where the criteria are ill defined, contextual cognition tests are employed, which are difficult to solve using software agents.
- Large numbers of sentences are collected from the Internet for the uniqueness.
- The feature of natural languages called "consonant gradation" is imitated. This method works as if literal/hearing errors, which prevents adversaries from obtaining clues by searching open documents.

The improvement of our new method compared with [7] is in terms of its usability.

Our first approach is particularly useful for visually impaired people and Braille users. The results reported in [7] suggest that the accuracy rate is lower for visually impaired users compared with that among totally blind users because of the negative impact of consonant gradation imitation. We suggest that this will also apply to Braille users.

We focus our attention on perceptual tests of the contextual naturalness of natural and unnatural sentences, where we can even synthesize natural sentences using a flexible-order Markov chain. The use of a low-order Markov chain prevents the extraction of sentences directly from the sources. A high-order Markov chain would maintain the meaningful contexts, but this method does not retain consonant gradation, although it is tolerant of attacks via search engines to some extent.

The second approach is helpful for all users. In our test, users need to read several sentences and make their selections. It may be necessary to reduce the options to decrease the mental load on users, but this is a weakness when brute force attacks are implemented.

Therefore, we devised our own response mechanism. Our system presents several sentences, and users must select the odd sentences in terms of their naturalness as well as their features; i.e., natural or unnatural. This complex response style is tolerant of brute force attacks but without requiring additional sentences.

C. Related Work on Verbal CAPTCHAs

Several researchers have studied text-only CAPTCHA using contextual cognition. Ximenes et al. [8] utilized phonetic punning riddles based on "knock knock" jokes (KK jokes). Their system challenges a user to differentiate real KK jokes from fake KK jokes. Unfortunately, the security of this method against attacks is weak because random guesses have a success rate of over 11%.

Kamoshida et al. [9] and Yamamoto et al. [10] proposed methods based on the impression of differences in strangeness between human-produced sentences and machine-generated sentences, which often have unnatural meanings. These methods require the use of hidden sets of documents to prevent the identification of the sources of the tests. However, these methods fail to generate brand new tests because the number of hidden documents is finite.

The WtP-CAPTCHA system is one of the most accessible, and it employs simple quizzes that require no specific knowledge. This method is accepted widely, but we demonstrate its security weakness in Section II.

II. Vulnerability of the CAPTCHA System Used by the "We the People" Site

A. Vulnerability against Pattern Matching Tactics

We collected and analyzed 1,300 questions (quizzes) from WtP-CAPTCHA in 2014. We successfully categorized them into a few syntactic structure patterns (Table I) using the matching rules shown in Fig. 2, as follows.

- *Type A:* Choose the *i*-th element from a list based on a specific condition; e.g., select a word related to a certain keyword.
- *Type B:* Choose (the number of) elements from a list based on a certain condition.
- *Type C:* Choose the biggest/smallest number from a list.
- *Type D:* Translate a spelled number into digits.
- *Type E:* Simple calculation.
- *Type F:* Other patterns.

Types C, D, and E are numerical quizzes, and thus they are rather easy to solve using programs.

To solve Types A and B, we need to query the word meanings with a semantic database. We use FrameNet II [11], [12], which provides the semantic group for each word. We answer each question according to the following steps.

Semantic Tactics against Types A and B.

1) Parse the question sentence and extract a keyword that indicates the topic category.
2) Look up the semantic group of the keyword in the database.
3) Parse the question sentence and extract a word list that comprises candidates for the answer.
4) For each word in the list, look up its semantic group in the database and obtain an ordered set of words with the same meaning as the keyword.
5) Choose the *i*-th word from the Type A set and (the number of) words from the Type B set.

B. Limitations of Amending the Current WtP-CAPTCHA

In our opinion, WtP-CAPTCHA tends to employ numerical questions because this does not require specific knowledge, and this is very convenient for the uniqueness. However, it is very easy for humans and software agents to solve these questions.

Non-numerical questions have well-defined patterns, thus potential adversaries can identify them. This problem is caused by the need to autogenerate questions easily that only require general knowledge. The system employs new patterns of question sentences and new words, but this not an effective solution for avoiding pattern matching attacks using a semantic database based on machine learning.

Type A	^([^:]+)\s*:\s+(?:the\s+)?(.+)(KEYWORD)\s+is\s*[\?\.:]?\s* ^(?:What\s+is|What's)\s+(?:the\s+)?(.+)\s+digit\s+in\s+(.+)\s*[\?\.:]?\s*$ Note that "KEYWORD" represents the string of "number|color|colour|body\s+part".
Type B	^The\s+list\s+(.+)\s+contains\s+how\s+many\s+(.+)\s*[\?\.:]?\s*$ ^How\s+many\s+(.+)\s+in\s+the\s+list\s+(.+)\s*[\?\.:]?\s*$
Type C	^Which\s+of\s(.+)is\s+the\s+(highest|largest|biggest|lowest|smallest)\s*[\?\.:]?s* ^([^:]+)\s*:\s+the\s+(highest|largest|biggest|lowest|smallest)\s+is\s*[\?\.:]?\s*
Type D	^Enter\s+the\s+number\s+(.+)\s+in\s+digits\s*[\?\.:]?\s*$ ^(?:What\s+is|What's)\s+(.+)\s(?:as\s+a\s+number|as\s+digits)\s*[\?\.:]?\s*$
Type E	^(?:What\s+is|What's)\s+(.+)\s+(OP)\s+(.+)\s*[\?\.:]?\s*$ ^(.+)\s+(OP)\s+(?:is\s+what|=|equals)\s*[\?\.:]?\s*$ Note that "OP" represents the string of "plus|add|minus|subtract|multiply|divide|\ + |\ − |\ ∗ |/".
Type F	(?:^The\s+colou?r\s+of|is\s+what\s+colou?r|what\s+colou?r\s+is\s+it) ^(.+)[,:]\s+which\s+.+\s+weekend\s*[\?\.:]?\s*$

Fig. 2: Samples of Regular Expressions Used by Our C# Attack Program (Partial).

TABLE I: Category of Queries in WtP-CAPTCHA during February/March, 2014.

Type of Questions	Examples	Appearance Rate [%]
Type A	*Ant, yellow and red: the 2nd colour is?* *In the number 9501081, what is the 3rd digit?*	29.6
Type B	*The list bank, snake, mosquito, restaurant, finger and cake contains how many body parts?* *From days Monday, Friday, Sunday or Wednesday, which is part of the weekend?*	26.5
Type C	*Of the numbers thirty one, 91, 30, seventy nine, forty five or sixty one, which is the largest?* *Enter the biggest number of eight, twenty one or 46:*	21.5
Type D	*Enter the number eighty seven thousand nine hundred and sixty nine in digits:* *What is thirty eight thousand seven hundred and seventy as digits?*	17.0
Type E	*What's 16 - three?* *3 plus ten is what?*	3.8
Type F	*If the cake is brown, what colour is it?* *What day is today, if tomorrow is Thursday?*	1.5
Fail to Categorize	*Which of hair, finger, heart, knee or toe is part of the head?*	0.1

III. Alternative Solution for Semantic CAPTCHA

We propose a safe and accessible CAPTCHA scheme, which is a challenge–response test where users are required to evaluate the naturalness of sentences. The ill-defined perception of naturalness is available only to humans, and its cognitive mechanism is difficult for software agents to mimic.

A. Notations

Let C be a multiassociative set. We use $c \leftarrow C(k)$ to indicate that c is sampled randomly from the values associated with the key k. Suppose that $C(k) = \{c_0, c_0, c_0, c_1\}$, c will be replaced by c_0 with a probability of 75% and by c_1 with a probability of 25%.

We denote the number of words in a string s by $|s|$. Let S be an array of strings. We use **concat**(S) to indicate a concatenation string of each element in S. We denote an integral element x from i to j by $x \in [i, j]$.

B. Basic Idea: Exploiting the Human Ability to Perceive the Naturalness of Sentences

We use a Markov chain model to synthesize sequences of words. A low-order Markov chain produces very awkward sentences. By contrast, a high-order Markov chain generates meaningful sentences to some extent, which may appear to be somewhat more natural. Thus, we obtain sentences with different degrees of naturalness and utilize them in our CAPTCHA system.

To implement this idea, we consider the following problems.

Uniqueness: We need to generate brand new tests at a high rate, but a high-order Markov chain often produces tests that are highly similar to those used previously.

Detection of Source Documents: We build the Markov chain models using unique source documents from the Internet. The high-order models may output sentences that are identical to those found in their sources. If we naively use these sentences as natural cases, potential adversaries may detect their naturalness simply by searching for these sentences on the Internet, rather than by applying semantic analysis.

We overcome these problems by varying the order N of the Markov chains. Suppose that the intervals $[N_{NS,L}, N_{NS,H}]$ and $[N_{WS,L}, N_{WS,H}]$ represent a range of flexible orders for generating a natural sentence (NS) and an unnatural sentence (commonly known as a "word salad," WS), where $N_{WS,L} < N_{NS,L}$, $N_{WS,L} < N_{WS,H}$, and $N_{NS,L} < N_{LS,H}$. The model order changes within the range when synthesizing sentences, and thus the uniqueness and naturalness of the sentences depend on the range. The order $N_{*,L}$ ensures the uniqueness of the sentence, which prevents the generation of the same sentences found in their sources, although humans are more sensitive to the difference between low orders in terms of naturalness. The high-order $N_{NS,H}$ helps humans to appreciate the naturalness of a sentence. This feature overcomes the shortcomings discussed above.

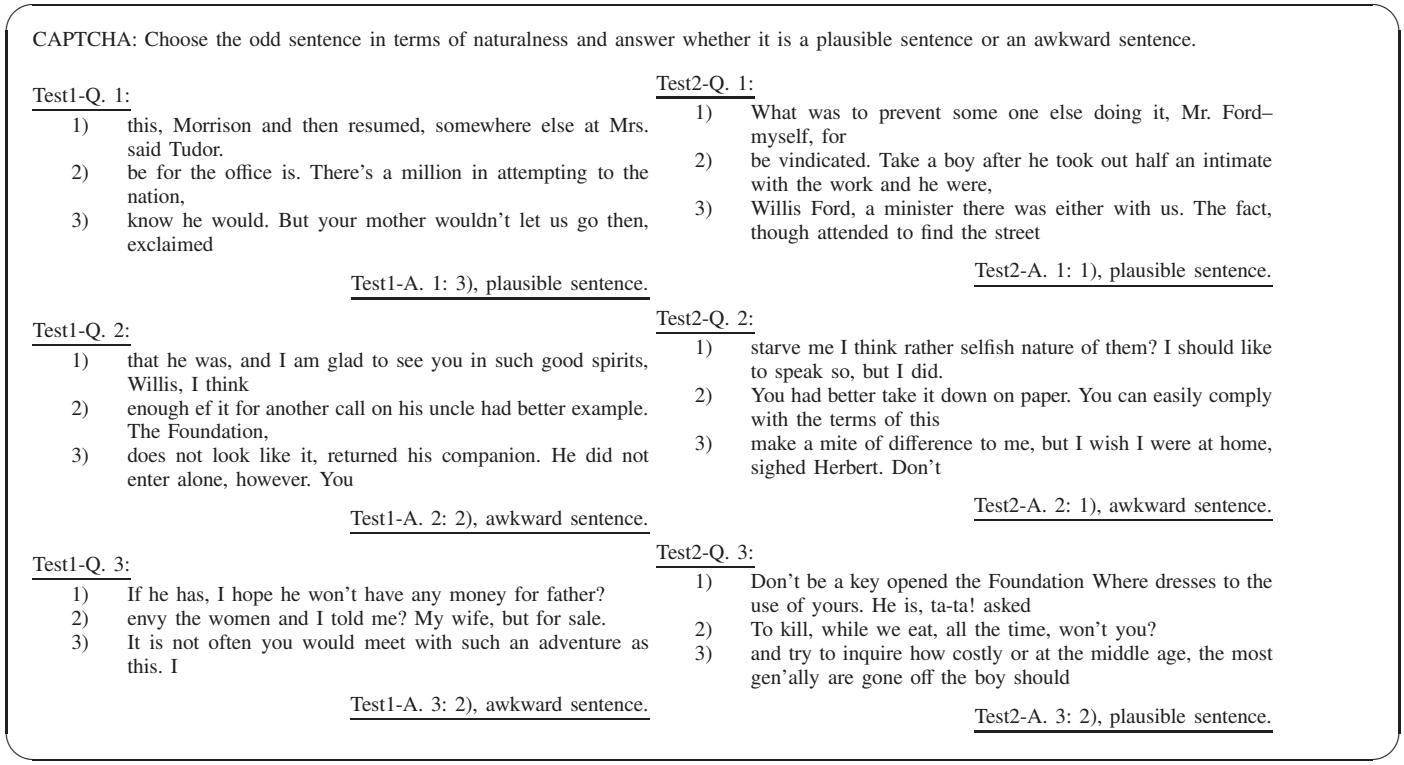

Fig. 3: Samples of Tests Generated Using Our System ($[N_{NS,L}, N_{NS,H}] = [3, 4], [N_{WS,L}, N_{WS,H}] = [1, 2]$).

C. Flow of the Challenge–Response Test

Our program generates CAPTCHA tests as follows.

Overview of the Process.

1) Collect source documents from the Internet and generate/update a corpus. Model the corpus with a multiorder Markov chain. The program may skip this step provided that it can maintain the uniqueness.
2) Generate a question and its answer k times based on the corpus.
   - Synthesize natural and unnatural sentences using a Markov chain where its order is in the range $[N_{NS,L}, N_{NS,H}]$ and $[N_{WS,L}, N_{WS,H}]$, respectively.
   - Produce a trio of sentences as the question. The trio comprises: 1) two natural sentences and one unnatural sentence, or 2) one natural sentence and two unnatural sentences. The selection is decided randomly.

   The answer to the question is a single sentence; i.e., the unnatural sentence for 1), and the natural sentence for 2).
3) Present the set of questions as a single trial to the user; i.e., a prover.
4) Receive the answers from the user. The system checks the correctness. When the number of correct answers is greater than or equal to a certain threshold t, the system admits the user as a human.

Fig. 3 shows several examples of our tests.

D. Details of the Synthesis of Sentences

We build a multiassociative set for each order $N \in [N_L, N_H]$ and combine them as a corpus C for the Markov chain model. When a sequence of N words (i.e., N-gram) is input as the key, the corpus outputs a certain word that tends to appear with words that belong to the N-gram in the source document. The procedure used to build the corpus is the same as that described in [7]. Let L_L and L_H be the minimum and maximum number of words in a synthesized sentence, respectively. Let $W_L, W_M, W_H \in [0, 100]$ be control parameters. The program produces sentences synthesized by a Markov chain where the order is in the range of $[N_L, N_H]$, as follows.

Sentence Generation.

(Input: L_L, L_H, W_L, W_M, W_H, N_L, N_H and C.)

1) Pick L uniform randomly from $[L_L, L_H]$.
2) Pick an initial order N uniform randomly from $[N_L, N_H]$ and generate the beginning of a synthesized sentence.
   - Collect N-gram words, which are registered in C as associative keys, and pick a *key* uniform randomly from among them.
   - Pick a $val \leftarrow C(key)$.
   - Assign $S \leftarrow (key, val)$.
3) If $L \leq |\mathbf{concat}(S)| \leq L_H$, then output $\mathbf{concat}(S)$ as a synthesized sentence and finish. Otherwise, go to Step 4.
4) If $N_L = N_H$, then set the model order $N' \leftarrow N$. Otherwise, change N' temporarily, as follows.

$N \leq N_L$:

$$N' = \begin{cases} N_H & \text{(with probability } W_L/100) \\ N+1 & \text{(otherwise)} \end{cases}$$

$N \geq N_H$:

$$N' = \begin{cases} N_L & \text{(with probability } W_H/100) \\ N-1 & \text{(otherwise)} \end{cases}$$

Else:

$$N' = \begin{cases} N-1 & \text{(with probability } W_M/100) \\ N+1 & \text{(otherwise)} \end{cases}$$

5) Choose a word for the next chain and save the current order N.
   - Assign $key \leftarrow (S[|S| - N'], \ldots, S[|S| - 1])$. If key is registered in C, then pick a $val \leftarrow C(key)$. Otherwise, compute $N' \leftarrow N' - 1$ and reexecute this process. Note that reexecution may only occur when $N' > N + 1$.
   - Assign $S \leftarrow (key, val)$.
   - Set $N \leftarrow N'$.
6) Go back to Step 3.

IV. Evaluations of the Accessible CAPTCHAs

The performance levels of the three methods are compared in Table II, where the symbols represent the following: '+++' represents a strength, '++' represents a strength to some extent, '+' represents a weakness, and '()' represents likelihood. We discuss the items described in Table II in order.

A. Condition

We built an attack program and collected questions for use as targets, as described in Section II.

We also built a program to implement the method described in Section III. We utilized *Project Gutenberg* [13] as source documents. We specified the parameter set as $(N_{NS,L}, N_{NS,H}) = (3, 4)$, $(N_{WS,L}, N_{WS,H}) = (1, 2)$, and $\{(L_L, L_H), (W_L, W_M, W_H)\} = \{(15, 25), (100, 50, 100)\}$. We collected sentences, which comprised 63,481 words and 6,430 types of words.

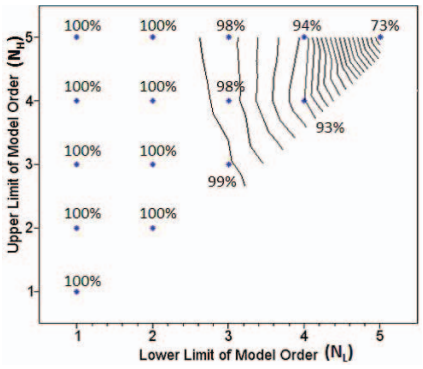

Fig. 4: Distribution of Unique Sentences.

B. Security Against a Random Guessing Attack

We set an upper limit of 1% for the success rate of a random guessing attack [14]. Our previous scheme satisfied this criterion. Under the conditions described in Section III-C, the probability of success for a brute force attack based on random guessing is $\sum_{i=t}^{k} \binom{k}{i} P^i (1-P)^{k-i}$, where $P = 1/6$. We asked the users to solve at least 3 questions without mistakes, and thus the success probability was 0.47%, and our new scheme also satisfied the criterion.

It appeared that WtP-CAPTCHA performed better than our methods when subjected to this type of attack because it uses numerical questions. This was the case provided that a brute force attack was applied. However, it was possible to guess the answer in an effective manner based on the information in Table I; i.e., several Type B questions often required the user to answer with a single number. Indeed, we successfully beat the system by entering '1' in over 10% of the cases based on 1,000 replicates. This shows that WtP-CAPTCHA is vulnerable to random guess attacks.

C. Security Against Known Walk-through Tactics

We considered two features: uniqueness and rate of detecting source documents. The uniqueness represents the strength against an attack based on collecting old questions and matching them using a database. The detection rate represents the strength against an attack based on the extraction of clues from open documents using search engines.

Uniqueness: We ran each program in Table II 1,000 times. Our new and previous programs successfully generated unique questions in 100% of cases.

The WtP-CAPTCHA system generated different questions in over 99% of cases, but the questions produced were similar in terms of sentence constitution. We attacked the WtP-CAPTCHA system using a program that implemented the algorithm described in Section II-A. Based on 1,300 trials, our program could categorize the WtP-CAPTCHA questions in over 99% of cases. We tested 300 of the answers and confirmed that our program broke the WtP-CAPTCHA code in over 99% of cases. These results show that WtP-CAPTCHA was severely vulnerable to our pattern-matching attack.

We also studied the uniqueness distribution with various order ranges using our new scheme. We generated 10,000

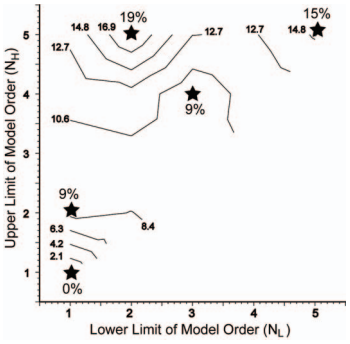

Fig. 5: Distribution of the Source Detection Rate.

TABLE II: Comparison of Three Methods.

Method	Security		Usability		
	Random Guess	Walk-throughs	Braille Use	Operability	Time Consumption
WtP-CAPTCHA [5]	+	+	+++	+	+++
Our Previous Study [7]	++	++	(+)	+++	+
Our Proposal	++	++	(+++)	+++	++

sentences for each case. The results are shown in Fig. 4, where lines with a number mean border of uniqueness rate. These results demonstrate that potential adversaries could beat our system, which was based simply on high order ranges such as $(N_L, N_H) = (5, 5)$, by matching with old sentences. Therefore, it is difficult for a Markov chain with a single order to utilize a sufficiently large order.

Rate of Detecting Source Documents: Adversaries may try to find the sources of sentences using search engines. We assess this risk as follows: We generate 100 synthesized sentences as described in Section III-D. For each sentence, we query it with the *Bing* search engine and check whether the result includes a source or book title that contains the corresponding sentence in the top 10 results. If the sentence is detected, then the adversary is considered to have found its source.

The results are shown in Fig. 5, which demonstrates that the adversaries could detect the source of about 20% of the sentences in the worst case.

However, even if adversaries can detect the source, they cannot beat our system immediately because our system modifies the natural sentences. Moreover, in the conditions described IV-A, adversaries cannot decide whether a sentence is natural based only on the results of a search. Thus, it seems that walk-throughs of our methods are difficult to achieve because this would require the emulation of complex human cognitive processes.

D. Usability

Braille use: WtP-CAPTCHA and our proposed method will satisfy the requirements of Braille users, whereas our previous method will not because it requires the imitation of consonant gradation.

Operability: WtP-CAPTCHA requires that visually impaired users solve various quizzes, and thus the ability to perform calculations is required in some cases. Our methods only require the ability to recognize the naturalness of sentences.

Time Consumption: Our proposed method requires that users read more text than WtP-CAPTCHA, although we have improved this in our latest version by developing a novel response mechanism.

Suppose that we employ a simple response mechanism such as selecting a natural (or unnatural) sentence. Because P is $1/3$, the probability of success for a brute force attack is 3.7% ($= P^3$). Thus, at least 5 questions need to be submitted to achieve the same security level as our proposed method. Therefore, our response method has an advantage in terms of its time requirements compared with the simple method.

V. Conclusion

We demonstrated the weakness of the CAPTCHA system employed by the White House site. We also constructed an alternative CAPTCHA system that differentiates humans from software agents based on the ability to appreciate the naturalness of a sentence. In future research, we will test whether visually impaired users can solve our tests in actual experiments.

References

[1] L. von Ahn, M. Blum, N. J. Hopper, and J. Langford, "CAPTCHA: Using Hard AI Problems for Security," in *Proceedings of EUROCRYPT*, vol. 2656. Springer-Verlag, 2003, pp. 294–311.

[2] J. P. Bigham and A. C. Cavender, "Evaluating Existing Audio CAPTCHAs and an Interface Optimized for Non-visual Use," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2009, pp. 1829–1838.

[3] E. Bursztein, S. Bethard, C. Fabry, J. C. Mitchell, and D. Jurafsky, "How Good are Humans at Solving CAPTCHAs? A Large Scale Evaluation," in *Proceedings of the 2010 IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2010, pp. 399–413.

[4] S. Shirali-Shahreza and M. H. Shirali-Shahreza, "Accessibility of CAPTCHA Methods," in *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*. ACM, 2011, pp. 109–110.

[5] The White House. (2014) We the People: Your Voice in Our Government. [Online]. Available: https://petitions.whitehouse.gov/

[6] BBC News. (2013) Blind Federation Criticises CAPTCHA Security Test. [Online]. Available: http://www.bbc.com/news/technology-22754006

[7] M. Yamaguchi, T. Nakata, T. Okamoto, and H. Kikuchi, "An Accessible CAPTCHA System for People with Visual Disability—Generation of Human/Computer Distinguish Test with Documents on the Net" in *Proceedings of Human-Computer Interaction International 2014*. Springer-Verlag, 2014 (to be published).

[8] P. Ximenes, A. Santos, M. Fernandez, and J. Celestino, "A CAPTCHA in the Text Domain," in *On the Move to Meaningful Internet Systems: OTM 2006 Workshops*. Springer-Verlag, 2006, vol. 4277, pp. 605–615.

[9] Y. Kamoshida and H. Kikuchi, "Word Salad CAPTCHA—Application and Evaluation of Synthesized Sentences," *15th International Conference on Network-Based Information Systems*, pp. 799–804, 2012.

[10] T. Yamamoto, J. Tygar, and M. Nishigaki, "CAPTCHA Using Strangeness in Machine Translation," *2013 IEEE 27th International Conference on Advanced Information Networking and Applications*, pp. 430–437, 2010.

[11] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The Berkeley Framenet Project," in *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*. Association for Computational Linguistics, 1998, pp. 86–90.

[12] F. Homepage. (2014) Welcome to Framenet. [Online]. Available: https://framenet.icsi.berkeley.edu/fndrupal/

[13] P. Gutenberg. (2014) Project Gutenberg: Main Page. [Online]. Available: http://www.gutenberg.org/wiki/Main_Page

[14] E. Bursztein, M. Martin, and J. Mitchell, "Text-based CAPTCHA Strengths and Weaknesses," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*. ACM, 2011, pp. 125–138.