FluTCHA: Using Fluency to Distinguish Humans from Computers

Kotaro Hara
Human-Computer Interaction Lab
Computer Science Department
University of Maryland
College Park, MD 20742
kotaro@cs.umd.edu

Mohammad Taghi Hajiaghayi
Computer Science Department
University of Maryland
College Park, MD 20742
hajiagha@cs.umd.edu

Benjamin B. Bederson
Human-Computer Interaction Lab
Computer Science Department
University of Maryland
College Park, MD 20742
bederson@cs.umd.edu

CHI 2012, May 5–10, 2012, Austin, TX, USA.
ACM xxx-x-xxxx-xxxx-x/xx/xx.

Abstract
Improvements in image understanding technologies are making it possible for computers to pass traditional CAPTCHA tests with high probability. This suggests the need for new kinds of tasks that are easy to accomplish for humans but remain difficult for computers. In this paper, we introduce Fluency CAPTCHA (FluTCHA), a novel method to distinguish humans from computers using the fact that humans are better than machines at improving the fluency of sentences. We propose a way to let users work on FluTCHA tests and simultaneously complete useful linguistic tasks. Evaluation studies demonstrate the feasibility of using FluTCHA to distinguish humans from computers.

ACM Classification Keywords
H5.0. General

General Terms
Experimentation, Human Factors, Languages

Introduction
Even though there have been significant improvements in computational power, there are some tasks that programmers have as yet been unable to create effective algorithms to solve. People, however, can often solve such tasks with only a few seconds of effort.

For example, reCAPTCHA [1] asks people to read and enter characters from an image. It is a CAPTCHA [2] that distinguishes humans from computers, and it is often used on web registration sites to filter out spammers. ReCAPTCHA also harnesses human power to work on an image-understanding task that is hard for computers. ReCAPTCHA assigns two words for each task. The system knows the solution to one of the words while the other is unknown (i.e., it has not yet been transcribed.) To pass the test, people must type in (i.e., transcribe) both words. As a result, unknown words get transcribed – but individual answers cannot be relied upon until multiple matching transcriptions for the words are given.

The idea of asking users to complete tasks that simultaneously benefit them and the system is very powerful. However, recent improvements in image understanding technologies can break CAPTCHA tests with high probability. According to Yan et al., [8] it is possible to break image recognition task with nearly 100% accuracy (although in practice, they continue to have a much lower break-in rate.) This suggests that we need to consider new CAPTCHA tasks that are more difficult for computers to solve, while remaining easy for people.

We introduce FluTCHA as a new kind of CAPTCHA. Instead of distorted text image recognition tasks, FluTCHA asks users to perform textual linguistic tasks – editing non-fluent texts to make them fluent. This is something that is relatively easy for native language speakers, but remains very difficult for computers [5].

The challenge with FluTCHA is that the resulting edited sentences need to be evaluated for fluency – which we already know computers are not good at doing. So the full FluTCHA activity combines the human linguistic activity with a second human activity to grade the modified texts. Thus FluTCHA is both a CAPTCHA and an example of human computation. Other people grade the edited sentences over the Internet. FluTCHA uses human graders from other FluTCHA users or can hire workers from Amazon Mechanical Turk (AMT).

In order to generate an unlimited number of the non-fluent sentences, FluTCHA collects original sentences from Japanese news websites and sends them through a machine translation system to generate non-fluent English sentences. Many sentences processed in this way are understandable even though they are not fluent. News articles are generated everyday, so we have a continuous source of new content.

FluTCHA distinguishes humans from computers while simultaneously using that same human effort to generate a corpus of human edits of machine-translated sentences that could be used in future machine translation systems.

Related Work

One other CAPTCHA system has previously considered sentence fluency. SS-CAPTCHA [10] provides users with a number of sentences. Out of those sentences, P sentences are natural sentences that are generated by humans or collected from human work. Q of them are language from a target language. Then users are asked to select P natural sentences from P+Q sentences. If users successfully choose P natural sentences, they are classified as humans. Otherwise, they are classified as computer programs.

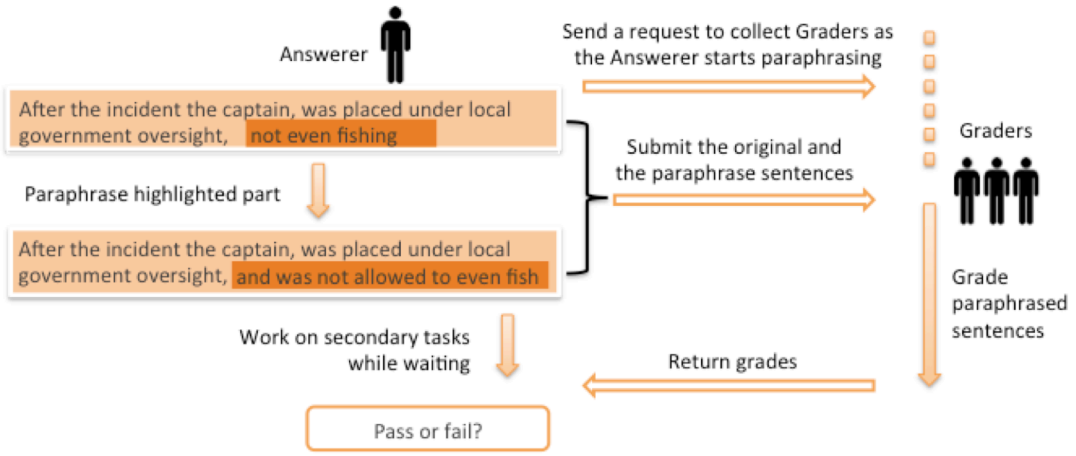

figure 1: FluTCHA workflow. The answerer on the left is the entitity being tested for being human. The graders on the right help determine whether the entity on the left is human.

Original Japanese	大統領が署名を事実上拒否し、交渉が行き詰まっていることが理由という (The negotiations stalled because the president refused to sign.)
Machine Translation	Effectively refused to sign the President, that is why negotiations stalled.

table 1: A non-fluent machine translated sentence is generated from a sentence in a Japanese newspaper article by Google Translate. The author translated the original Japanese sentence into the English sentence in parentheses.

Both FluTCHA and SS-CAPTCHA focus on fluency of sentences. However, SS-CAPTCHA uses source sentences collected from public sources on the web. This implies once spammers know where the source sentences are taken from, they will know the correct answers for SS-CAPTCHA tasks, i.e. it is vulnerable against dictionary attacks. FluTCHA, on the other hand, generates the non-fluent sentences automatically, and is not vulnerable, even if the source of original sentences is discovered. Moreover, the FluTCHA process results in linguistic data that could be useful to improving machine translation systems. This could, of course, result in FluTCHA being made obsolete by high quality machine translations sytems, but that would be a positive outcome in its own right.

FluTCHA was motivated in part by MonoTrans, a system that combines machine translation and monolingual humans to collaboratively translate text than machine translation could not do alone [7]. As with MonoTrans, FluTCHA uses computer and human processing where each can provide value in a way that the other cannot.

FluTCHA System

Machine translation services such as Google Translate often provide translation sufficient for humans to understand the meaning of original texts. However, translated texts tend to not be fluent. Table 1 shows an English translation of a sentence taken from a Japanese news article. To make such sentences more fluent, we ask humans to modify the machine translated sentence to improve the fluency of the sentence.

Figure 1 shows the workflow of FluTCHA, which involves two groups, *answerers* (an entity tested by FluTCHA) and *graders*. As answerers come to a web site registration page, FluTCHA gives them a task that shows a non-fluent text with an area of consecutive words highlighted. FluTCHA also provides the same sentence where a text field substitutes the highlighted area with a textbox that the answerer must fill in with a paraphrased version of the highlighted text (i.e., text that has the same meaning, but is fluent.)

After answerers finish paraphrasing, FluTCHA takes the original translated sentence (*prefluent* sentence) and the paraphrased sentence (*postfluent* sentence) and sends them to graders. FluTCHA then provides graders with a pair of prefluent sentences and their corresponding postfluent sentences in random order. Graders are asked to grade the fluency of each sentence on a nine-point scale. Graders are also asked to grade similarity in meaning between the two sentences. This allows FluTCHA to notice if an answerer entered a random phrase such that the phrase changes the meaning between sentences.

While answerers are waiting for their work to be graded, FluTCHA asks them to work on a secondary task. FluTCHA, for example, could provide answerers with tasks to grade other answerers' works, so we would not have to pay workers on AMT.

We intend FluTCHA to be used in scenarios such as web service registration, thus postfluent sentences need to be graded in nearly real-time and results must be sent back to answerers quickly. In order to achieve that, we can collect workers as soon as an answerer starts working on paraphrasing. If we constantly have enough users using FluTCHA, we can ask answerers to work on grading other answerers' as a secondary task described in the previous paragraph. Alternatively, we could use the quikTurkit approach introduced by Bernstein et al. [4] so that there are graders available when needed.

Evaluation

To show that FluTCHA has the potential for being effective, we need to show that FluTCHA is capable of distinguishing humans from computers with high accuracy. We collected 478 sentences from 44 news articles by crawling a Japanese news website. We then translated them into English by Google Translate.

We asked five native English speakers to volunteer for a study. First, participants highlighted non-fluent parts of machine-translated sentences to create prefluent sentences (*highlighting* task). Then we asked them to

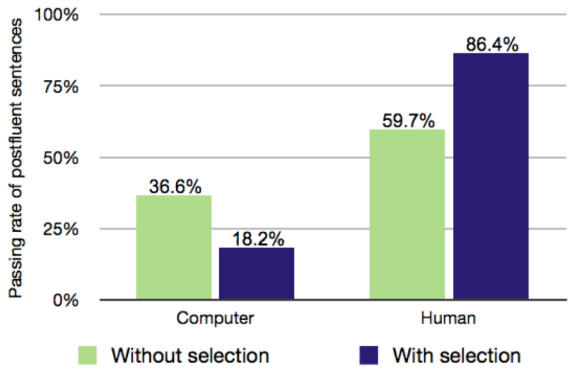

figure 2: Passing rate of computer and human generated postfluent sentences created without and with sentence selection process.

paraphrase the highlighted region of the sentences that are highlighted by other participants to generate postfluent sentences (*paraphrasing* task). The paraphrasing task took 39.2 sec to complete on average.

After each highlighting task was completed, the highlighted parts of prefluent sentences were sent to Google Translate to translate them from English to Japanese, then from Japanese back to English (*pivot*). We substituted the highlighted part of the corresponding prefluent sentences with pivoted phrases to generate pivoted sentences. We use these sentences to evaluate computers' capability to generate fluent paraphrases for prefluent sentences.

Having created prefluent, postfluent, and pivoted sentences, we posted tasks on Amazon Mechanical Turk (AMT), an inexpensive online labor market where task requesters can post microtasks, to rate the fluency of those prefluent, postfluent, and pivoted sentences along with similarity in meaning between those sentences with a nine point scale (*grading* task). We asked workers on AMT to work on two grading tasks per assignment for 5 cents. We collected 898 grades from 83 distinct workers. Each set of sentences was graded from 6 to 17 times.

We considered tasks as passing if any given sentence showed at least 1 point improvement compared to it's matching prefluent sentence.

Postfluent sentences generated by humans passed single tests at a rate of 59.7% (536/898), while pivoted sentences generated by computers passed at a rate of 36.6% (329/898).

To improve the success rate of humans and decrease the success rate for computers, we can select prefluent sentences used for tests that have been proven to be effective for distinguishing humans from computers. To automatically select those sentences, we used an F-measure to balance precision and recall [3]. It is defined as follows where P represents the precision of FluTCHA separating human work from computer work and R represents the success rate of human work:

$$F = \frac{2PR}{P+R}$$

$$P = \frac{\#\ passing\ human\ works}{\#\ passing\ human\ works + \#\ passing\ computer\ works}$$

$$R = \frac{\#\ passing\ human\ works}{\#\ passing\ human\ works + \#\ failing\ human\ works}$$

We selected sentences that had an F-measure larger than 0.8. The result is shown in Figure 2. The success rate for human work increased from 59.7% to 86.4% while the success rate for computer work decreased from 36.6% to 18.2%. By calculating cumulative probabilities, we estimate that people can pass the test 98.1% of the time after 2 trials. The expected number of trials for humans to pass the test is 1.16 (1/86.4%) and the expected number of trials for computers to pass the test is 5.49 (1/18.2%).

Discussion

Computers can generate paraphrases by pivoting one language with another, but it does not guarantee the paraphrase improves the fluency of the original. If spammers can write programs that improve fluency, they could break FluTCHA. There are studies of generating paraphrases to improve quality of sentences

[6,9]. However, these require sufficient amount of human support or domain specific supervised learning. Thus, we believe that these technologies are not currently applicable to break FluTCHA.

The fact that it takes 39.2 seconds for a single task is an important limitation, but not necessarily a showstopper. Many registration systems require a multi-step authentication which validating an email address – which also takes some time. We could do the same thing; send an email when the task is validated.

We used Japanese as the source and English as the target of translation. Future work will be to test various target languages so non-English speakers can use the system. Moreover, this would be an advantage of our design. Spammers sometimes hire people in low-income countries to break the visual CAPTCHA. However, if we use FluTCHA instead, people who are not fluent in the target language cannot pass the tests.

Conclusion

In this paper, we describe a novel way to distinguish humans from computers using tasks to make machine-translated sentences fluent. The work done using FluTCHA could also be used to improve the fluency of machine-translated sentences. In our experiment, we have shown it is possible to distinguish humans from computers by FluTCHA tests. We showed that we can distinguish between humans and computers with humans succeeding 86.4% of the time while computers succeed only 18.2% of the time.

Acknowledgements

We are grateful to everyone in HCIL for helpful discussions and feedback.

References

[1] Ahn, L.V., Maurer, B., McMillen, C., Abraham, D., and Blum, M. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science*, September 12, 2008, 1465-1468.

[2] Ahn, L.V., Hopper, N., Blum, M., and Langford, J. CAPTCHA: Using Hard AI Problems for Security. *Proc. Eurocrypt*, 2003, 294–311

[3] Baeza-Yates, R.A. and Ribeiro-Neto, B. Modern Information Retrieval, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1999

[4] Bernstein, M.S., Brandt, J., Miller, R.C., and Karger D.R. Crowds in Two Seconds: Enabling Realtime Crowd-Powered Interfaces. *Proc. UIST 2011* 33-42

[5] Callison-Burch, C., Bannard, C. and Schroeder, J. 2004. Improving statistical translation through editing. *Workshop of the EAMT*. 2004 26-32

[6] Chen, D.L. and Dolan, W.B. Collecting Highly Parallel Data for Paraphrase Evaluation. *Proc. ACL,* 2011, 190-200

[7] Hu, C., Bederson, B.B., and Resnik, P. MonoTrans2: A New Human Computation System to Support Monolingual Translation, *Proc. CHI 2011* 1133-1136

[8] Yan, J and Salah, A.E.A. Breaking Visual CAPTCHAs with Naïve Pattern Recognition Algorithms, in *Proc. ACSAC* 2007. IEEE computer society, 279-291.

[9] Liu, C., Dahlmeier, D., and Ng, H.T., PEM: A paraphrase evaluation metric exploiting parallel texts. *Proc. EMNLP 2010* 923-932

[10] Yamamoto, T., Tygar, J.D., Nishigaki, M. CAPTCHA Using Strangeness in Machine Translation. *24th IEEE AINA* 430-437