A System to Transcribe Documents in European Languages with Human Help

Ravishankar Chityala[1] **and Sridevi Pudipeddi**[2]
[1] Minnesota Supercomputing Institute, University of Minnesota, Minneapolis, USA
[2] Mathematics Department, Waldorf College, Forest City, Iowa, USA

Abstract—*Existing text based CAPTCHAs can only transcribe words in an image that contains ASCII characters. In this paper, we are describing a system called UCAPTCHA, which is intended to transcribe words in an image that contains Unicode characters to text. Such a system will be useful in transcribing scanned documents from many European languages like Spanish, German, French etc.*

Keywords: CAPTCHA, OCR, Unicode, Image transcription.

1. Introduction

Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) present a problem that only humans can solve and computers may not solve. This system can tell the difference between computer and human. In this age of world wide web, CAPTCHAs can be used to prevent spamming.

There are many different types of CAPTCHAs. One type of CAPTCHA is the hand written CAPTCHA [2] (figure 1) that uses database formed of handwritten names of American cities which were selected from postal letters. An image of a randomly selected city is shown to user to gain entry to the service. But poor quality images can make it hard for the user. EZ-Gimpy and Gimpy CAPTCHA [4] - selects words from its dictionary of 860 words and displays them corrupted and distorted in an image to gain access to the service. Due to the small size of the dictionary, this CAPTCHA was broken. Audio Based CAPTCHA [9] - make use of simple words or numbers drawn from a random recordings and alter them with some noise or disturbance. These CAPTCHAs proved to be useful for visually impaired people. In general, these CAPTCHAs are hard to solve due to accent and language barriers. More CAPTCHAs can be found in these papers [6], [1], [5], [3]. Shah and Banday gave a nice summary of some CAPTCHAS with a short description on how each one works in their paper [7].

2. Motivation

Although there are many CAPTCHAs, in this paper we will be focusing on an image based CAPTCHA. The motivation for this CAPTCHA was the image based reCAPTCHA. reCAPTCHA [9] (figure 2) is a free web based CAPTCHA service that helps in digitizing books. In reCAPTCHA,

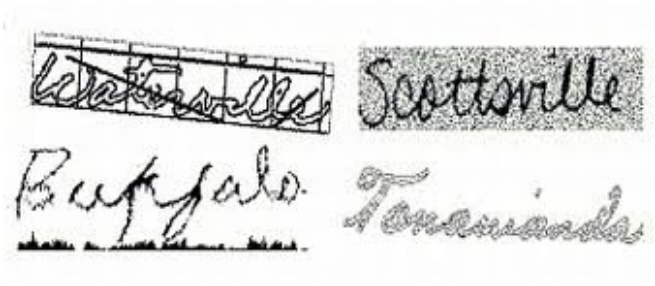


Figure 1: Hand Written CAPTCHA

the user is presented two words, obtained by scanning of books, the “control“ word and ”unknown“ word. The user is expected to solve the control word correctly and his response to the unknown word is recorded. By presenting the same unknown word to many different users, an accurate transcription of the unknown word can be obtained. reCAPTCHA presents words that are in ASCII character set and hence can help digitize and transcribe books, documents in English.

Figure 2: reCAPTCHA

Many of the European languages like Spanish, French, German, Italian etc. have many characters that are similar to ASCII but differ by a few accented characters. In this paper, we describe a CAPTCHA system, called UCAPTCHA that will present words having Unicode characters. The users can respond to most of the characters using ASCII characters from their keyboard and complete the Unicode characters using on-screen keyboard. This system will help transcribe books, documents etc. in European languages like Spanish, German, French etc. And hence increase the utility of

CAPTCHAs in transcription of valuable documents, maps, books etc.

In a typical setup, the user will be presented double words obtained from scanned documents either as a part of a HTML forms in some website or through our site http://ucaptcha.msi.umn.edu. One of the words in the setup has not been transcribed while the other word is transcribed. We will be referring to them as "unknown word" and "known word" respectively. When the user solves the CAPTCHA, we evaluate their response to the known word. If the response was correct, we assume that the unknown word was responded correctly and record both responses. If the response to the known word was incorrect, we present an error message and request the user to complete the UCAPTCHA again.

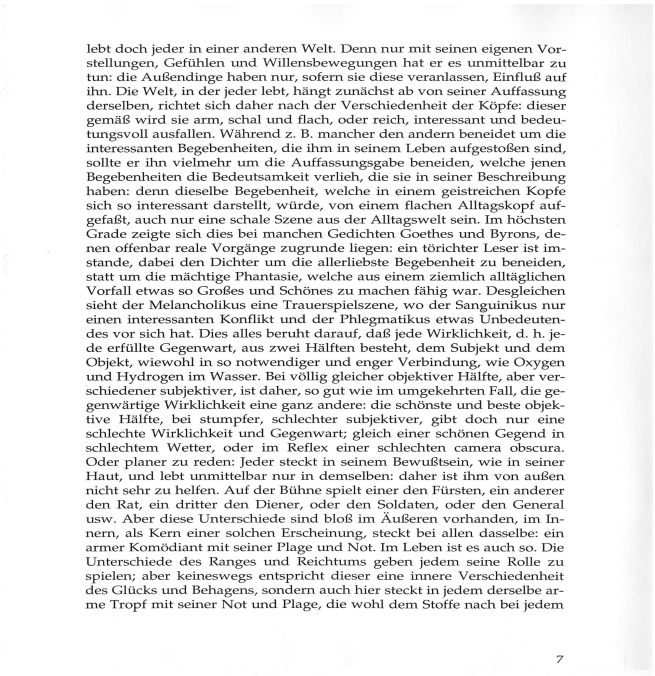
3. Methodology

In a CAPTCHA based system, the words shown to users would be obtained by scanning the documents that need to be transcribed. The scanned words that could be transcribed using an Optical Character Recognition (OCR) program will be converted to text. Some of the words may not be transcribed because they might be ill formed due to distortion and page curvature, artifact from the scanning process, poor quality of the original document etc. Such words that cannot be transcribed by computers will be transcribed by a human. The first step therefore is to convert scanned documents in to text using OCR. During this process, some of the words may not be transcribed. The details and the images of the words that could not be transcribed are uploaded to a database and served using a web application to users. We used two different OCR programs, GOCR [10] and Tesseract [11] to convert scanned documents to text. The text files generated by both the software were compared and the words that did not match were flagged as non-matching words and the rest were identified as matching words. We also recorded the location of the words. To obtain the image of the non-matching words and some of the matching words, we used the location obtained in the previous step. We then processed the scanned document based on the following steps, that were coded in Matlab, [12] and Python, [13].

- Project the document after adaptive thresholding in the horizontal direction and obtain a profile
- Segment the projected profile and separate individual lines based on their connectivity
- On the lines which have the non-matching word
  - Perform vertical projection after performing dilation
  - Segment the projected profile of the words
  - Separate individual words based on their connectivity
  - Store the individual words as non-matching words and a few matching words from that line. Even though we are interested in the non-matching words we still require matching words and the importance of these words will be explained in the section, "Embedding mode".

Let us consider a document as a prototype to explain the above-mentioned steps in detail. The document in figure 3 is passed through GOCR and Tesseract. The figure 4 is a image of a line which has one non-matching word and another matching word from figure 3.

Note: From now on we call matching words in the both GOCR and Tesseract as known words and the non-matching words as unknown words.



lebt doch jeder in einer anderen Welt. Denn nur mit seinen eigenen Vorstellungen, Gefühlen und Willensbewegungen hat er es unmittelbar zu tun: die Außendinge haben nur, sofern sie diese veranlassen, Einfluß auf ihn. Die Welt, in der jeder lebt, hängt zunächst ab von seiner Auffassung derselben, richtet sich daher nach der Verschiedenheit der Köpfe: dieser gemäß wird sie arm, schal und flach, oder reich, interessant und bedeutungsvoll ausfallen. Während z. B. mancher den andern beneidet um die interessanten Begebenheiten, die ihm in seinem Leben aufgestoßen sind, sollte er ihn vielmehr um die Auffassungsgabe beneiden, welche jenen Begebenheiten die Bedeutsamkeit verlieh, die sie in seiner Beschreibung haben: denn dieselbe Begebenheit, welche in einem geistreichen Kopfe sich so interessant darstellt, würde, von einem flachen Alltagskopf aufgefaßt, auch nur eine schale Szene aus der Alltagswelt sein. Im höchsten Grade zeigte sich dies bei manchen Gedichten Goethes und Byrons, denen offenbar reale Vorgänge zugrunde liegen: ein törichter Leser ist imstande, dabei den Dichter um die allerliebste Begebenheit zu beneiden, statt um die mächtige Phantasie, welche aus einem ziemlich alltäglichen Vorfall etwas so Großes und Schönes zu machen fähig war. Desgleichen sieht der Melancholikus eine Trauerspielszene, wo der Sanguinikus nur einen interessanten Konflikt und der Phlegmatikus etwas Unbedeutendes vor sich hat. Dies alles beruht darauf, daß jede Wirklichkeit, d. h. jede erfüllte Gegenwart, aus zwei Hälften besteht, dem Subjekt und dem Objekt, wiewohl in so notwendiger und enger Verbindung, wie Oxygen und Hydrogen im Wasser. Bei völlig gleicher objektiver Hälfte, aber verschiedener subjektiver, ist daher, so gut wie im umgekehrten Fall, die gegenwärtige Wirklichkeit eine ganz andere: die schönste und beste objektive Hälfte, bei stumpfer, schlechter subjektiver, gibt doch nur eine schlechte Wirklichkeit und Gegenwart; gleich einer schönen Gegend in schlechtem Wetter, oder im Reflex einer schlechten camera obscura. Oder planer zu reden: Jeder steckt in seinem Bewußtsein, wie in seiner Haut, und lebt unmittelbar nur in demselben: daher ist ihm von außen nicht sehr zu helfen. Auf der Bühne spielt einer den Fürsten, ein anderer den Rat, ein dritter den Diener, oder den Soldaten, oder den General usw. Aber diese Unterschiede sind bloß im Äußeren vorhanden, im Innern, als Kern einer solchen Erscheinung, steckt bei allen dasselbe: ein armer Komödiant mit seiner Plage und Not. Im Leben ist es auch so. Die Unterschiede des Ranges und Reichtums geben jedem seine Rolle zu spielen; aber keineswegs entspricht dieser eine innere Verschiedenheit des Glücks und Behagens, sondern auch hier steckt in jedem derselbe arme Tropf mit seiner Not und Plage, die wohl dem Stoffe nach bei jedem

7

Figure 3: A German document used as a prototype

derselben, richtet sich daher nach der Verschiedenheit der Köpfe: dieser

Figure 4: The fifth line from the German document which contains one unknown word and one known word

4. The website

Once the words were obtained, we upload the information like the name of the document, the location of the word

Figure 5: known word from figure 4

Figure 6: unknown word from figure 4

in the document and the location of the word in the file system to a MySQL database. During this process, we also distorted the word by a randomly distributed distortion. A website available at http://ucaptcha.msi.umn.edu was coded in Django and serves two words. Users can then use an on screen keyboard to input Unicode characters, while the ASCII characters can be entered using the regular keyboard. In the double words, one word is known whereas the other word is unknown. As the double words are presented, the properly transcribed unknown word will then be moved to the known words list. The unknown word will be transfered to the known words list provided 95% of the transcription are the same.

The words are then presented in two modes: Demonstration mode and Embedding mode.

4.1 Demonstration Mode

In the demonstration mode, we present the two words to the users (Figure 7) in our website. By default the virtual keyboard is not on. The user can turn it on using the keyboard symbol at the bottom of the text box. Figure 8 is the view with the virtual keyboard on the web page. Completing the CAPTCHA would take the user to a second page that indicates whether the user was successful or not.

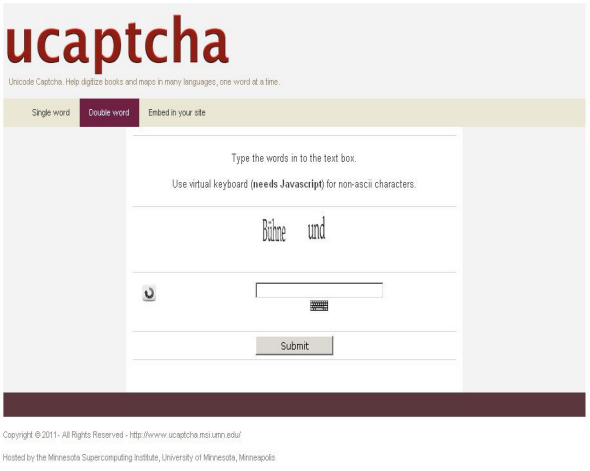


Figure 7: View of the webpage in the demonstration mode without the virtual keyboard on.

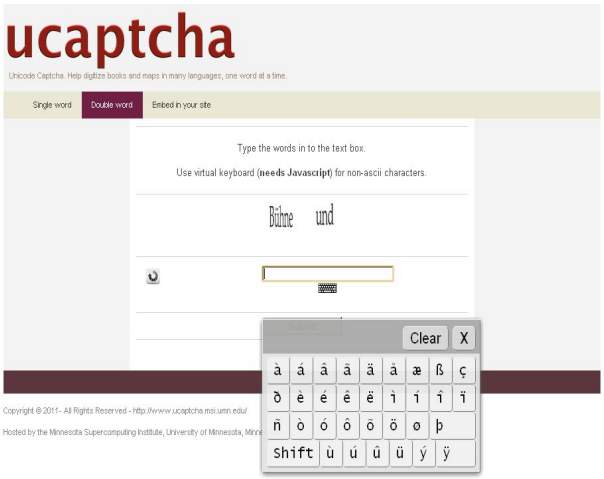


Figure 8: View of the webpage in the demonstration mode with the virtual keyboard on.

4.2 Embedding Mode

In the embedding mode, the mode that would be the most useful way to disseminate the UCAPTCHA to more users is shown in figure 9. The website that needs to embed the UCAPTCHA, can request the HTML code for embedding by filling the "Embed in your site" link. After processing the form, the web master of the embedding site would be provided a HTML code that needs to be embedded in their HTML POST form. When a user visiting the site fills this form and the UCAPTCHA, the response to UCAPTCHA is sent to our site while the other information is passed to the embedder website. If the user fails in solving the known word correctly, a string of value zero is returned, otherwise a value of one is returned.

A web-master can also request words from a specific language when they request the embedding code. Thus, a German website will serve German words in its UCAPTCHA. This also has an advantage, that the German user solving the Unicode can do so using their physical keyboard instead of the virtual on-screen keyboard. This allows the user to solve the UCAPTCHA faster and with less hassle. In case, where the user does not have a physical German keyboard, they can use the virtual on-screen keyboard.

5. Discussion

Since there are only two words being presented, it is possible that users can guess the location of the unknown word. Once located, they could spam the response to the unknown word while filling the known word correctly. This would pass the test we perform to check if the response is correct. Such spamming can skew the transcription results. In order to prevent users from guessing the location of known and unknown word, we locate them at random at run time.

Figure 9: View of the webpage in the Embedding mode.

In some cases, we also present two known words to the user. Also, to prevent the users from seeing the same word, we maintain a large list of words.

6. Conclusion

In this paper, we present a CAPTCHA system useful for transcribing words from text containing Unicode characters. Such a system will be useful in transcribing scanned documents from many European languages like Spanish, German, French etc.

The CAPTCHA can be in demonstration mode, in which a known and unknown word are presented in our site. It can also be embedded in other sites. In both cases, when a user responds to the CAPTCHA, as long as the response to known word is correct, we assume that the unknown word was typed correctly. The same unknown word is presented to many different users and if more than 95% responses are same, we assume that the unknown word was transcribed correctly. We then extract the unknown word from the unknown list and place it in the known word list.

7. Future work

We have assumed that the scanned document is free of any distortions and artifacts but in a real life scenario, these artifacts do exist. We will investigate various computational and algorithmic approaches in solving this problem.

Also, we will work with European libraries to obtain documents that need to be transcribed. This will help the libraries to transcribe digitize valuable documents.

8. Acknowledgment

We would like to acknowledge The Supercomputing Institute for Advanced Computational Research, University of Minnesota for hosting the web application.

References

[1] R. Ferzli, R. Bazzi, L. J. Karam, A captcha based on the human visual system masking characterists, ACME, 2006.

[2] A. Rusu, V. Govindaraju, Handwritten CAPTCHA: using the difference in the abilities of humans and machines in reading handwritten words, proc. of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR- 9 2004), 2004.

[3] D. Misra and K. Gaj, Face Recognition CAPTCHAs, Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW 2006), 2006.

[4] L. von Ahn, M. Blum, and J. Langford, Telling Humans and Computers Apart Automatically, Communications of the ACM, vol. 47, no. 2, pp. 57-60, February 2004.

[5] M.H. Shirali-Shahreza and M. Shirali-Shahreza, Persian/Arabic Baffletext CAPTCHA, Journal of Universal Computer Science (J.UCS), vol. 12, no. 12, pp. 1783-1796, December 2006.

[6] M.H. Shirali-Shahreza and M. Shirali-Shahreza, Question-Based CAPTCHA, proc. of Int'l conference on computational Intelligence and Multimedia Applications, 2007.

[7] Shah N.A and Banday M.T, Drag and Drop Image CAPTCHA, Sprouts: Working Papers on Information Systems, 8(46), 2008; http://sprouts.aisnet.org/8-46

[8] L. von Ahn, B. Maurer , C. McMillen, D. Abraham, and M. Blum, Science, Vol. 321, pp. 1465-1468, September 2008

[9] C. Nancy, Sound oriented captcha, in: Proceedings of the First Workshop on Human Interactive Proofs, Xerox Palo Alto Research Center, CA, 2002.

[10] http://jocr.sourceforge.net/

[11] http://code.google.com/p/tesseract-ocr/

[12] http://www.mathworks.com/

[13] http://www.python.org