AreCAPTCHA: Outsourcing Arabic Text Digitization to Native Speakers

Menna Bakry, Mohamed Khamis, Slim Abdennadher
Computer Science & Engineering Department
The German University in Cairo, Egypt
menna.bakry@student.guc.edu.eg, {mohammed.khamis, slim.abdennadher}@guc.edu.eg

***Abstract*—There has been a recent increasing demand to digitize Arabic books and documents, due to the fact that digital books do not lose quality over time, and can be easily sustained. Meanwhile, the number of Arabic-speaking Internet users is increasing. We propose AreCAPTCHA, a system that digitizes Arabic text by outsourcing it to native Arabic speakers, while offering protective measures to online web forms of Arabic websites. As users interact with AreCAPTCHA, we collect possible digitizations of words that were not recognized by OCR programs. We explain how the system works, the challenges we faced, and promising preliminary evaluation results.**

I. Introduction and Related work

In the recent events in Egypt, the country has lost many irreplaceable Arabic books and documents [1]. Similar Arabic text, that is only present in physical form, might be lost forever if not backed up. Digital copies have the advantages of being searchable, queryable and not losing quality over time. The motive behind this work is to propose an effective approach to digitizing Arabic books and documents.

The number of Arabic-speaking Internet users has been greatly increasing, reaching 86 million in 2011 [2]. Moreover, Arabic is the 7^{th} most used language on the Internet [3]. The increasing number of users who can read and write Arabic online is an opportunity that can be exploited to help digitizing Arabic documents.

Previous work by Luis von Ahn et al., reCAPTCHA [4], exploits the human's ability to read distorted text in order to digitize books. Like CAPTCHA [5], reCAPTCHA is used in online web forms to make sure that the entities filling the forms are humans. This is done by presenting the user with distorted characters that are difficult for OCR programs to read, but relatively easy for humans to read. This way, the online forms present a test that only humans can pass. Von Ahn improved this approach to further include two words instead of one; one of the words is known and is used as a test to the entity filling the online form, while the other word is unknown, and is taken from a book that was not digitized because OCR failed to fully recognize it. With the immense numbers of reCAPTCHAs filled daily, books can be digitized in a fraction of the time and costs needed if they were to be digitized by human workers.

The reCAPTCHA project was successful. It was later acquired by Google, who uses it to digitize one of its biggest products: Google Books. Currently, 200 million reCAPTCHAs are filled everyday [6]. Till recently, reCAPTCHA has only supported a few languages. Currently it allows customizing the displayed reCAPTCHA to even include unsupported languages, but then the reCAPTCHA in this case would not be helping in digitization.

A lot of websites adopted reCAPTCHA to protect their forms from being abusively filled by malicious programs. It is remarkable that many Arabic websites use reCAPTCHA in its English form, even though the majority of their visitors are Arabs. We expect this to be both: an inconvenience to users who might not speak but Arabic, and a waste of the valuable skill of reading Arabic that these users posses.

Additionally, existing OCR programs that support Arabic are limited and are relatively inaccurate [7]. This increases the demand for an equivalent reCAPTCHA system that can contribute in digitizing Arabic text. In this paper, we propose AreCAPTCHA, an Arabic reCAPTCHA system that is used to digitize Arabic books and documents. We pay attention to the Arabic-related challenges. We also provide an easy way for websites that already use von Ahn's reCAPTCHA to substitute it with our AreCAPTCHA. Additionally, we provide a safe fall-back to English reCAPTCHA option to assure webmasters that their websites will remain secured.

II. Arabic reCAPTCHA

In order for AreCAPTCHA to replace the current reCAPTCHAs on Arabic websites, it was required that the system would satisfy the initial purpose of any CAPTCHA, which is to validate that the entity filling the form is a human. Thus in every AreCAPTCHA, we display two words; the digitization of one of the words is *known* to us, and this is the one we use to validate that the user is a human. While the digitization of the other word is *unknown* to us; and this is the one we would like to digitize.

To achieve this, we first had to collect a Words Bank, in which words are classified as either *known* or *unknown*. Following that, a service should be made available for Arabic websites to use AreCAPTCHA. The service should handle displaying the AreCAPTCHA, as well as processing the results and determining the digitization most likely to be correct for every word.

A. Words Bank

The first step was to collect a words bank to be used for the AreCAPTCHA. As shown in Figure 1, the first stage is where the system reads a scanned document (in image format),

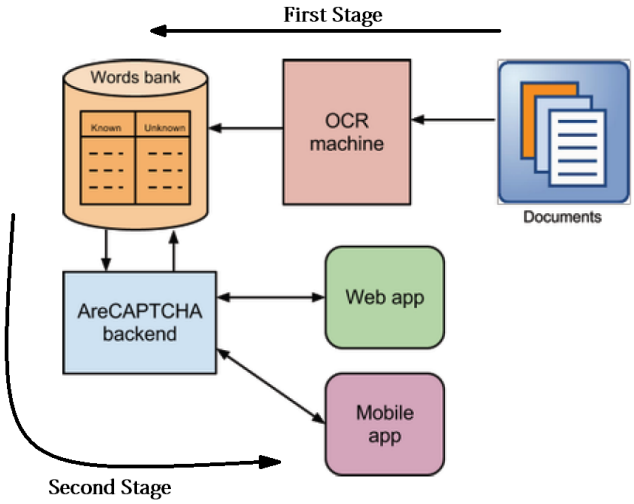

Fig. 1. Stages of creating the AreCAPTCHA application

extracts the words, and decides which of these words can be used for the human test (the known words), and which need to be digitized (the unknown words). The list of known and unknown words form the words bank.

We scanned sample pages and attempted to digitize them using two Arabic OCR platforms: The first one is Tesseract, it is an open source engine that was developed at HP labs then improved by Google, and released under the apache license 2.0.[8]. The second platform is ABBYY, it is an OCR software that is not free, but it allows users to upload images and get back a response with the OCRed data. Tesseract seemed more suitable for the task as it provides more information per digitized word. Additionally, unlike Abbyy, it had the advantage of being able to digitize multiple columned papers without misplacing the words.

Using the OCR engine, and with the help of the OpenCV library, the scanned image is cropped around the boundaries of each word into multiple smaller images for each word in `.jpg` format. Moreover, the system distorts the word images using random lines and dots. Using a library called ImageMagick, some wavy distortions are added to all images before saving them. This makes it harder for malicious bots to pass the test, while still not conveying to the user which word is used for which purpose (see Figure 2).

For each word image, an identifier for each one is stored in a database, accompanied by the following:

- An identifier
- The digitization suggested by the OCR engine
- A certainty rate, which is a number generated by the OCR engine (Tesseract) that refers to how sure the engine is about the correctness of the digitized word.
- The classification of this word: *known* or *unknown*.
- The system also keeps track of the "most repeated digitization" across all users for each word-image.
- Finally, the "digitization failures", which is the number of times users failed to read this image.

Fig. 2. Adding random waves, dots and lines as extra degradations to cropped images

It seemed plausible at first to depend on the certainty rate provided by Tesseract as an indicator to classify the words to *known* and *unknown*. However, the value was found to be inaccurate, as in many cases high certainty values were given for incorrectly digitized words, and vice versa.

At this point, the database was ready with the bank of known and unknown words. However, the number of known words was limited due to the inaccurate OCR digitization. Therefore, we manually selected some words to be used as the *known* words.

B. The Web Service

The second stage in the project was the backend of the Arabic reCAPTCHA web service. The backend was created and exposed as a web service that is ready to respond to two kinds of HTTP requests:

The first request retrieves a new AreCAPTCHA with two words: a known word and an unknown word. Both of the words are distorted and placed randomly so that the user would not know which one is used for which purpose.

The second request validates the AreCAPTCHA where the user's answer to the known word is compared to its corresponding digitization on the server. If the digitization input of the user for the known word is incorrect, then he fails the test. In this case, failure count for the known word is incremented in the database, and the user receives two new words. If the failure count exceeded a certain number of failures (10 failures), we assume that the digitization that we already have for this known word on the server might be incorrect. Therefore, this image is prevented from being displayed in the AreCAPTCHA among the known category, and it is considered to be among the unknown category that still needs to be digitized by users instead.

On the other hand, if the digitization input of the user for the known word is correct, then he passes the test, and his answer for the unknown word is trusted and inserted in the database as a possible digitization. Afterwards, the most repeated digitization which is the most common digitization from users for the unknown word is updated in the database. Hence, if the number of users agreeing on a certain digitization for this unknown word exceeded a certain number of votes, then this image is successfully digitized and is treated as an image from the known category that is used as a human test for users. Consequently, every time a user fills an AreCAPTCHA correctly, it helps us to digitize a new word and increase our number of known words in the database and this is how we solved the problem of having a limited number of known words from the OCR.

In von Ahn's implementation of reCAPTCHA, an unknown word has to obtain at least 2.5 votes before it could be considered as a correctly digitized word [4]. Each human answer is counted as one vote and each OCR digitization is counted as one half of a vote (recall that these words all have been previously processed by OCR). Hence, if the first two human guesses match each other and one of the OCRs, they are considered a correct answer; if the first three guesses match each other but do not match either of the OCRs, they are considered a correct answer, and the word becomes a known word. However, in AreCAPTCHA, we assume a word is digitized only if 3 or more digitizations were collected for the word, and the number of similar digitizations is more than half the total number of digitizations collected for this word. We discuss details about choosing this metric in the evaluation section.

Furthermore, the AreCAPTCHA has a refresh button تحديث that allows users to request a new pair of words (see Figure 3). When six users refresh before any correct spelling is entered for a word, the word is flagged as *unreadable* since users cannot distinguish it well. The system later reduces the distortions applied to this word and pushes it again to the *unknown* category.

Fig. 3. Arabic reCAPTCHA example

C. Integration to existing websites

Integrating our Arabic reCAPTCHA with any website is simple since we provide a single library file that can be downloaded by webmasters. It enables them to link their websites to the already implemented functions on the server that shows and validates the AreCAPTCHA. This process simulates the English reCAPTCHA in the way it is presented to developers to enable them to migrate to AreCAPTCHA easily. Moreover, we added a feature that gives them a safe fall back to switch to the English reCAPTCHA if AreCAPTCHA was not desired, as well as giving them the freedom to choose any styling theme they want.

III. Evaluation

In order to anticipate the success of our Arabic reCAPTCHA, it was essential to evaluate the correctness of the collected data. It was also important to investigate whether or not Arab users are ready to fill it instead of the English one.

A. Collected Data

AreCAPTCHA was hosted on http://recaptchadomain.webege.com/AreCaptcha/AreCaptcha.php. We distributed the link during the testing phase through Emails and social media (Facebook, Twitter).

1) Data Quantity: We collected a total of 288 digitizations for 168 different scanned words (see Table I).

Number of words	Collected digitizations per word
88	digitized once
44	digitized twice
33	digitized thrice
2	digitized four times
1	digitized five times

TABLE I. Number of words, and number of times they were digitized by our users

After filtering out the words that were digitized two or more times (80 words), we found that all of the different digitizations for 65 of them matched across different user inputs (81.25%). While in the 15 other cases, there were 2 different digitizations for 13 of words, and 3 different digitizations for the remaining 2 words. It is interesting that in some cases, all 5 and 4 digitizations of the same words were matched across the different users.

B. Data Quality

Taking a closer look at the words that were digitized 2 or more times, we concluded the values in Table II. Generally, the tendency to get at least one correct digitization is very high (90%) if we consider words that were digitized 2 or more times, and even higher (94.44%) if we only consider those digitized 3 or more times.

Considering the words that were digitized 2 times only did not yield good results. Moreover, the number of words that were digitized 4 and 5 times is too small to be significant.

1) When is a digitization correct?: In von Ahn's implementation of reCAPTCHA, digitizations are given points depending on their source: digitizations by users are given 1 point per user, and those by OCR are given 0.5 points. As mentioned earlier, the threshold used in von Ahn's reCAPTCHA is 2.5.

In our case however, relying on the OCR results is not plausible with the relatively low quality of the Arabic OCR technologies. Moreover, the results from Table II show that 2 digitizations are not sufficient for Arabic words. We attribute this to the complex nature of the Arabic alphabet and symbols. Thus, we consider a digitization to be correct only if it was matched across more than half the number digitizations for this word, with a minimum of 3 digitizations per word.

It is unfortunate that we only reached an 83.33% correct-rate using that metric. While von Ahn's reCAPTCHA reached a 99% correct-rate [4]. However, it is still much higher than

	Words with N digitizations					
	N=2	N=3	N=4	N=5	2≤N≤5	3≤N≤5
At least one of the digitizations is correct	84.09%	93.94%	100%	100%	90%	94.44%
At least half of the digitizations are correct	84.09%	84.85%	100%	100%	85%	86.11%
More than half of the digitizations are correct	11.36%	84.85%	50%	100%	43.75%	83.33%
Number of words with N digitizations	44	33	2	1	88	36

TABLE II. The number of correct digitizations

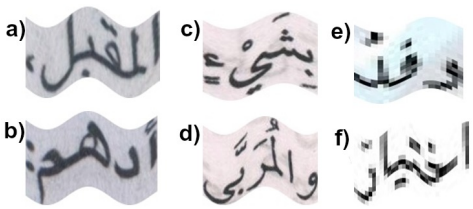

Fig. 4. Some words that were incorrectly digitized

most of the correct-rate of existing Arabic OCR technologies. Additionally, it can be noticed that a great increase exists once we consider 3 or more digitizations instead of 2 or more. This suggests that collecting more digitizations per word will yield even better results.

2) Reasons behind Incorrect Digitizations: After reviewing the incorrect digitizations for words that were digitized 2 or more times, it was found the incorrect digitizations were due to the following:

1) Disregarding punctuation characters: some of the displayed words had punctuation characters such as Figures 4a and 4b. This was responsible for many of the incorrect digitizations as many users ignored the punctuation characters either intentionally or unintentionally.
2) Confusing letters: Arab users seem very likely to confuse the letters ي and ى when placed near the end of the statement (Figures 4c and 4d respectively). A possible solution is to increase the minimum number of required digitizations when such characters exist.
3) Low quality of the scanned word: some users seem to have confused Arabic diacritics with dots due to bad quality. In Figures 4e and 4f, some users interpreted the diacritic symbol above the last letter as a dot, thus perceiving the letter as ز while it was actually ر. This is also due to not having the word in its context, which would normally help in identifying the word. This can be improved by using higher dpi when scanning the documents.
4) Some users wrote Arabic diacritics: despite being not requested, which led to false negatives when matching the digitizations. A solution is to strip the diacritics off the digitized words when matching them, since digitizing the diacritics is out of the scope of this work.

A sample of the output by AreCAPTCHA for one of the scanned paragraph can be seen in Figure 5.

C. Perceiving AreCAPTCHA

To investigate how Arabic Internet users perceive the Arabic reCAPTCHA, a questionnaire was administered to 64 participants.

It is interesting that in Egypt, as well as many countries in the Arab world, English is the dominating language in most of the universities, many schools, and many of the positions occupied by middle and upper class citizens. Thus, a concern was whether or not these users would prefer to fill an Arabic reCAPTCHA on the Internet instead of an English one. Therefore, the questionnaire was mainly targeting users who were expected to type English more than Arabic.

1) Demographics: The participants were of different age groups: around 55% were between 21 and 25 years old, around 16% were 26 to 30 years old, approximately 11% were 31 to 40 years old, and about 19% were above 40. They were all Egyptians except for a single Iraqi participant, thus they were all native Arabic speakers. The participants were of different professions and fields, including students (undergrad, master and PhD students), teachers, academics, pharmacists, lawyers, officers, managers, surgeons, engineers and auditors.

2) Questionnaire structure: The questionnaire was made available online. The participants were first asked to fill an online AreCAPTCHA. The participants were asked some questions to estimate how *fast* they type Arabic compared to English, which of the two languages they type with *more often*, and which they *prefer* to type with. The participants were asked whether they found Arabic reCAPTCHA a good idea, bad idea, or felt neutral towards it, in addition to their reasons behind that. They were asked as well which reCAPTCHA would they fill if they were to choose between Arabic and English reCAPTCHA. Finally, the participants were asked if they have any related comments.

3) Results and discussion: Our results show that out of the 64 participants, 18 (28%) type Arabic faster than English, while 46 (72%) type English faster. It can also be observed that 52 (81.25%) type more English than Arabic in their daily life, while 12 (18.75%) type more Arabic than English. Finally, 48 (75%) prefer typing English over Arabic, while 16 (25%) prefer Arabic over English.

Out of the 64 participants, 42 (66%) said they would prefer an English reCAPTCHA. While 22 (34%) said they would prefer an Arabic one. It is interesting out of those who preferred the Arabic reCAPTCHA, 11 (50%) reported that they type English *more often* than Arabic, 7 (32%) *prefer* typing in English, and 7 (32%) type English *faster* than Arabic. This shows that there is a reasonable probability that Arabic-speakers who are infrequent users of Arabic would still fill an Arabic reCAPTCHA.

Fig. 5. Scanned section of a book (top). The results by Tesseract (middle), and the results from AreCAPTCHA (bottom). Red are words that were not digitized. Yellow words are those correctly digitized by Tesseract. Green words are those correctly digitized by AreCAPTCHA.

Our results also show that (12.5%) think that Arabic reCAPTCHA is a bad idea, and 13 (20.3)% feel neutral towards it. While 43 (67.2%) find it a good idea. It should be noted that none of the participants were briefed before filling the questionnaire about the idea of digitizing books through AreCAPTCHA. Only 3 showed that they have prior knowledge of the earlier idea by von Ahn. Not knowing that this is the purpose of the project, the 3 participants suggested that we would digitize Arabic books using AreCAPTCHA. Many of the participants noted that they find it plausible to have Arabic reCAPTCHA in Arabic websites. Some suggested that it is culturally more convenient for Arabs. Others found it a good idea to encourage Arabic usage online. As for the negative reviews, some addressed the issue that many Arabs do not type Arabic frequently. Others expressed their fear that in the case of using AreCAPTCHA, the websites would be only usable by Arabic speakers.

In summary, the results show that although some Arabic Internet users might prefer English over Arabic in typing preference and speed, a reasonable portion of them are ready to participate in filling an Arabic reCAPTCHA.

IV. Conclusion and Future work

The preliminary results showed that AreCAPTCHA collected reasonable quality results. The achieved correctness-rate of collected digitizations is 83.33%, we strive to achieve better than that by collecting higher number of digitizations per word.

Additionally, when discussing the reasons behind incorrect digitizations, we suggested some improvements that can improve the results. These include defining more constrained metrics for words that contain letters that were found to be confusing to our users, using higher quality scanners with higher dpi and stripping the diacritics when matching the digitizations.

Previous work anticipated that Arabic recognition could be challenging for Tesseract [9]. Recent research has been carried out to provide OCR systems that are better suited for Arabic and Urdo [10]. Therefore, an area of improvement is by trying out new OCR systems that are optimized for Arabic, as well as trying commercial OCR softwares that could digitize Arabic documents with high accuracy and with reliable output information about the certainty rate numbers.

In the future, we intend to perform more experiments using more words with a more diverse set of sources. This will help us identify areas of improvement, and reach a better representation of the input of the system.

The punctuation and diacritics could be digitized using methods of Human Computation that were proven to be successful. For example, a Game With A Purpose [11] or an Arabic Grammar teaching software (similar to Duolingo [12]) can be implemented to collect such data from Arab users.

Eventually, we hope that the technique that we are proposing can help in redirecting the human processing power of a noticeable large number of users who can write and read a language that most of the other world's population cannot read. And we hope that the AreCAPTCHA can contribute in solving problems that most computers could not solve yet, and help to digitize human knowledge that is given in the precious Arabic documents and deserves to be protected.

Acknowledgment

The authors would like to thank the testers, who volunteered to fill AreCAPTCHAs, as well as the participants who gave us their time filling the questionnaire.

References

[1] "Library fire in Egypt clashes destroys 'irreplaceable' 200-year-old documents," http://www.cnn.com/2011/12/17/world/africa/egypt-unrest/index.html, 2011.

[2] "Arabic Speaking Internet Users Statistics," http://www.internetworldstats.com/stats19.htm, accessed: 2013-09-27.

[3] "Internet World Users by Language," http://www.internetworldstats.com/stats7.htm, 2013, accessed: 2013-09-27.

[4] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, "recaptcha: Human-based character recognition via web security measures," *Science*, vol. 321, no. 5895, pp. 1465–1468, 2008.

[5] L. Von Ahn, M. Blum, N. J. Hopper, and J. Langford, "Captcha: Using hard ai problems for security," in *Advances in CryptologyEUROCRYPT 2003*. Springer, 2003, pp. 294–311.

[6] "reCAPTCHA: Stop Spam. Read Books." http://www.google.com/recaptcha/aboutus, 2013, accessed: 2013-10-06.

[7] V. Märgner and H. El Abed, *Guide to OCR for arabic scripts*. Springer, 2012.

[8] R. Smith, "An overview of the tesseract ocr engine," in *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, vol. 2. IEEE, 2007, pp. 629–633.

[9] R. Smith, D. Antonova, and D.-S. Lee, "Adapting the tesseract open source ocr engine for multilingual ocr," in *Workshop on Multilingual OCR Cc*, 2009.

[10] N. Sabbour and F. Shafait, "A segmentation-free approach to arabic and urdu ocr," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2013, pp. 86 580N–86 580N.

[11] L. Von Ahn, "Games with a purpose," *Computer*, vol. 39, no. 6, pp. 92–94, 2006.

[12] L. von Ahn, "Duolingo: learn a language for free while helping to translate the web," in *Proceedings of the 2013 international conference on Intelligent user interfaces*. ACM, 2013, pp. 1–2.