Transcribing Handwritten Text Images with a Word Soup Game

Vicent Alabau and **Luis A. Leiva**
ITI/DSIC, Universitat Politècnica de València
Camí de Vera, s/n – 46022 Valencia (Spain)
{valabau,luileito}@{iti,dsic}.upv.es

CHI'12, May 5–10, 2012, Austin, Texas, USA.
ACM 978-1-4503-1016-1/12/05.

Abstract
The major contribution presented here is the transformation of the tedious process of transcribing text images into an enjoyable game. A web-based application uses a word soup interface and a game engine on top of an automatic handwriting transcription system as input to play the game. This paper describes the rationale and design principles for the game, envisioning evaluation strategies, and deriving insights for future developments.

Author Keywords
Distributed knowledge acquisition; Web-based games; Crowdsourcing; Handwriting Transcription

ACM Classification Keywords
I.2.6 [**Learning**]: Knowledge acquisition; H.5.3 [**Group and Organization Interfaces**]: Web-based interaction

Introduction
Transcribing handwritten text is an important research topic, especially because of the increasing number of publishers creating large quantities of digitized documents [5]. Lately, procedures for scanning books at high speed and relatively low cost have improved considerably; so much that at present it is possible to plan the digitization of millions of books per year. This fact has fostered the creation of digital libraries by public

institutions[1,2,3] not only to preserve the cultural heritage, but also to make it possible to index, copy, edit, or translate the texts, search for words, etc. Although many manuscripts have been transcribed as of today, most of the digital libraries host only images of the pages from the original books, which are mainly legacy documents that still need to be converted to a textual format[4]. Handwriting transcription can be a very laborious and expensive work, especially when the writing style of the text is too variable and the original document is particularly degraded.

HTR is not OCR

Handwritten Text Recognition (HTR) aims at alleviating the task of transcribing manuscripts. Note that this problem is much more complicated than that of Optical Character Recognition (OCR). State-of-the-art OCRs achieve very good performance (see, e.g., [1]) although there is no OCR engine that supports handwriting recognition as of today (Figure 1).

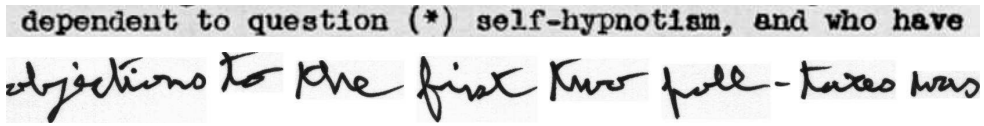


Figure 1: Typically OCR deals with documents presenting a predictable inter-word and inter-character space, consistent typesetting, etc. (1a) On the contrary, HTR is suited to transcribe cursive, continuous handwriting, presenting skewed/slanted words, irregular calligraphy, and so on (1b).

HTR is close to the Automatic Speech Recognition (ASR) problem. In fact, both are modeled in a very similar way. In HTR, the models must be trained for each particular book, making recognition even more complicated. What is more, HTR still needs human resources to improve. Concretely, three popular approaches to HTR are widely adopted at present: *post-edition* [2], where a human amends the automatic transcription of a system, *interactive-predictive* [5], where the human and the system iteratively collaborate to derive the best transcription, and *active learning* [4], where the system asks the user to transcribe the words with lower confidence measure.

Crowdsourcing Text Transcription

A technique which is having considerable success in recognizing difficult words is to submit them automatically to humans in a reCAPTCHA system [7]. Typically, users that solve a reCAPTCHA are rewarded in the form of a subscription to a service or an access to a restricted area. However, reCAPTCHA cannot be so directly applicable to HTR. Firstly, because line and word segmentation are not trivial. Thus, extracting isolated words to present to the user may be misleading. Secondly, cursive handwriting reading may be difficult even for human experts. Hence, a minimum of word context can be of great help (e.g., showing the n words surrounding the word to transcribe, or a full line sentence).

Crowdsourcing Tasks through Games

The reCAPTCHA concept led von Ahn *et al.* [6] to create what is now known as *games with a purpose* (GWAPs). In a GWAP, the players perform a useful computation as a side effect of an enjoyable game play. In this case, users are rewarded in terms of 'amusement'. GWAPs are designed for 2-player fast-paced gaming, so that the task is performed as soon as possible, usually with great success. This makes GWAPs social, since players have to validate each other's computation. However, 2-player GWAPs present logistical challenges, and in fact their

[1] http://www.bl.uk
[2] http://www.wdl.org
[3] http://www.cervantesvirtual.com
[4] http://www.cic.net/Home/Projects/Library/BookSearch/

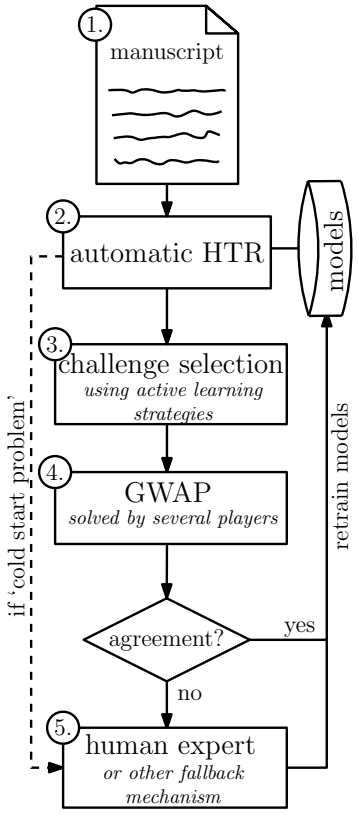


Figure 2: Flowchart for an active learning HTR system including GWAP as a low cost tool.

utility has been called into question [3]. Fortunately, there are two important features of handwritten documents that play to our advantage: *1)* except in very rare cases, there is no urge in having the transcriptions; and *2)* the number of manuscripts is finite — especially the legacy ones, since most of the documents generated nowadays are already in digital form. Consequently, in this paper we propose a transcription game that is inspired by reCAPTCHA and GWAPs. To our knowledge, the system presented here differs from previous approaches in the following items:

- We tackle the problem of HTR, a much harder problem than OCR even for a human.
- Our game is not based on input-agreement scoring; instead, it is a 1-player slow-paced game play based on a classical word search game.
- Instead of exclusively relying on human computation, our systems combines HCI and machine learning tools.

Game Rationale

We follow the active learning scenario proposed in [4], in which the system decides the parts (i.e., the words) of an automatic output (i.e., the transcription) to be revised by the human. Seamlessly combined with the role of GWAPs, our system provides an economic and alternative way of performing transcriptions. Thus, we propose to extend the active learning workflow to take into account the human computation through gaming, as depicted in Figure 2:

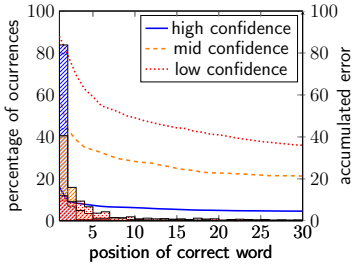
1. As we will focus on the transcription part, the manuscript is expected to be preprocessed and segmented into lines. At present, this process is quite standard and book-independent.
2. Then, the manuscript is passed to an automatic state-of-the-art HTR system [5]. It is worth noting that no HTR system is (and will be) completely error-free, hence human revision is eventually required.
3. As the system is unaware of where the errors are located, an active strategy is used to select a set of words in the transcription with high probability of being incorrect (see [4]). This set of words along with their confidence estimations are used to design a challenge (with the appropriate level of difficulty).
4. The challenge is presented as a game and should be played by different users. At this point, two outcomes are expected:

   On agreement. If the crowd agree on the correct transcription, then the HTR models are updated accordingly.

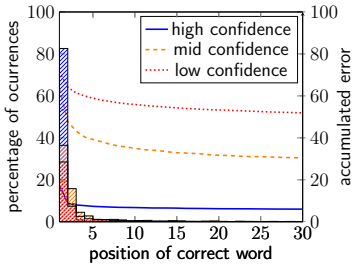
   On irresolution. If a particular challenge has not a clear agreement or it has not been solved by any user, then a contingency plan is applied.

5. When the challenge cannot be solved, the system resorts to a fallback mechanism. If resources are available, the contingency plan may be just to ask for transcription by a human expert. Otherwise, the transcription could be submitted to another GWAP or crowdsourcing tool, or be discarded from future challenges. Finally, if the correct transcription is obtained, then the HTR models are updated.

Note that both the validated and the discarded transcriptions (if any) are to be removed from the challenge selection in step 3. Eventually, the pool of possibly wrong transcriptions will be empty and the transcription of the manuscript will be finished. Additionally, the model update will make automatic HTR in step 2 less error prone [4], therefore boosting the convergence to better transcriptions.



(a) Cristo-Salvador corpus



(b) IAMDB corpus

Figure 3: Position statistics of the correct hypothesis in the n-best lists for two corpora. The bars represent the percentage of times that the correct word was at the given position. The lines display the accumulated error at a given size of n-best list.

The Cold Start Problem

In the case that the system cannot be provided with reasonable quality models initially, an expert (or exclusively crowdsourcing-based tools) should be used to create a statistically significant batch of transcriptions.

Game Design

A careful design of the GWAP in Figure 2 is crucial for a successful result. Traditional word games, such as Scrabble and the Hangman seem very suitable to this purpose. Both examples can be designed as a 2-player game, where the players validate each other's responses to questions regarding the image to be transcribed. However, to provide a 1-player game experience we can rely on machine learning (ML) methods instead.

The key idea of mixing HCI and ML is to allow the player resolve the uncertainty of the ML system toward a particular word transcription. A common way to express this uncertainty is by means of n-best lists. These lists provide a set of the most likely candidates along with a score of the goodness for the candidate. We define the confidence of the word w being correct as $c(w)$ [4]. Then, three categories raised naturally from the data sets: high confidence words ($c(w) = 1$), medium confidence words ($1 < c(w) \leq 0.5$), and low confidence words ($c(w) < 0.5$). Figure 3 shows statistics of the n-best list of two different corpora. The bars represent the percentage of times that the correct word was at the given position. As it can be seen, most of the time the correct transcription occurs in the first 10 positions. On the other hand, the lines display the accumulated error for a given size of n-best list. Note that it is possible that the correct transcription cannot be recognized, and thus, it could be not present in an n-best list regarding its size. This results in a residual error, where the curve stabilizes. For high confidence words, this happens at position 5, i.e. a 5-best list achieves almost the best residual error ($\sim 7\%$). Medium confidence words stabilize around 20 with a residual error of $20\% \sim 30\%$, while low confidence words do it around 30 with $35\% \sim 50\%$ of error. At this point, four questions arise: *1)* How will the player resolve the uncertainty? *2)* What will happen if the word is not in the n-best list? *3)* How to set different levels of difficulty? *4)* How to score the game plays?

The Word Soup Game

The word soup game (WSG) is a game that shows letters in a grid. The goal is to find all the words 'hidden' given a word list or a challenge (e.g. "find proper names"). Typically WSGs contain almost correct but erroneous words to confuse the player. This feature can be exploited to our advantage. Then, the mapping between the WSG and our transcription problem can be performed as follows:

- The challenge of the WSG is to find the transcriptions for the words selected in the step 3 of Figure 2. Examples of challenges would be 'find the second and fourth word from the image' or 'find the highlighted words in the image'.
- The soup is populated using n-best lists. As the player should know the correct transcription, the erroneous words will be ignored and the correct solution should be found (if any). This way, the user will resolve the ML uncertainty unknowingly.

Contingency Plan

Particularly for low confidence words, there is a risk that the correct solution is not found in the n-best list. To this end, a contingency plan must be devised since it can be frustrating to know that the correct solution is not in the WSG. Two mechanisms have been created so far: *1)* a timer is set depending on the difficulty of the task, and,

after expiring, the game suggests the player move to the next challenge; *2)* a link to report the current game (see also Evaluation Strategies).

Design of Difficulty Levels
The level of difficulty is established by the size of the grid: easy (10×10), standard (15×15), and hard (20×20). Each game level is assigned a maximum of n-bests, e.g. 20, 45 and 80 respectively. Then, the challenge is built by adding the least confident word as long as the maximum n-best size for the current game level is not reached. From Figure 3, high confidence adds the 5-best words, while mid and low add 20 and 30, respectively. For instance, a standard game would typically have 2 mid confidence words plus 1 high confidence word.

Game Play Scoring
Scoring can be obtained by the following expression: $s = \sum_w \delta(w)\left[1 - c(w) + b\right]$, where $\delta(w)$ equals 1 if the player found the word w and 0 otherwise, and b is a 'bias' score to be assigned to high confidence words.

Evaluation Strategies

Two features are key to assessing the utility of this game. First, outliers must be effectively discarded in order to decide the correctness of gathered users' data. To do so, we aggregate soup results that multiple players have (either partially or completely) solved. This would also help to make our game harder to cheat. Second, the system must avoid presenting "problematic cases" to the users, mainly as a consequence of the automatic image processing steps: image lines that are too hard to read, misleading challenges, or word soups that cannot be solved. To partially overcome these issues, a report link is included in both the game instructions and in the challenge description. When a user clicks on the report link, the image, on the one hand, will not be presented to the user again and, on the other hand, will be added to a 'black list' for later review. Moreover, if there is consensus from many users, the image will be definitely removed from the corpus used in the game — instead, it should be submitted to other HTR system, possibly one of the three popular approaches at present, see HTR is not OCR.

Discussion

We estimate that, if our game was deployed at popular online gaming sites, an important quantity of scanned books could be transcribed in a matter of days or weeks with low error rates. For this to succeed, the game must be enjoyable and addictive. However, clearly insolvable challenges present a big handicap, since users get frustrated when the solution is not in the soup.

Although a report system has been taken into consideration, lowering potential user frustration could be tackled by converting the soup into a fast-paced game, so that players cannot not realize whether the solution was in the soup or not. Another option would be to slightly change the rules of the soup game to facilitate the resolution of "problematic cases", e.g., by allowing the player to change some letters of the selection. Hopefully, these novelties in the game play will attract more casual players and will keep them motivated. Additionally, a refinement of reCAPTCHA could be used as a fallback mechanism, e.g., asking for the transcription of the wrong word and its known context, or drawing lines to separate the words in a segment. Finally, we will report on a formal human evaluation. A set of different challenges and game play scenarios will be defined and assessed by real players. We expect this evaluation will identify the weaknesses of the proposals to improve the playability of future developments.

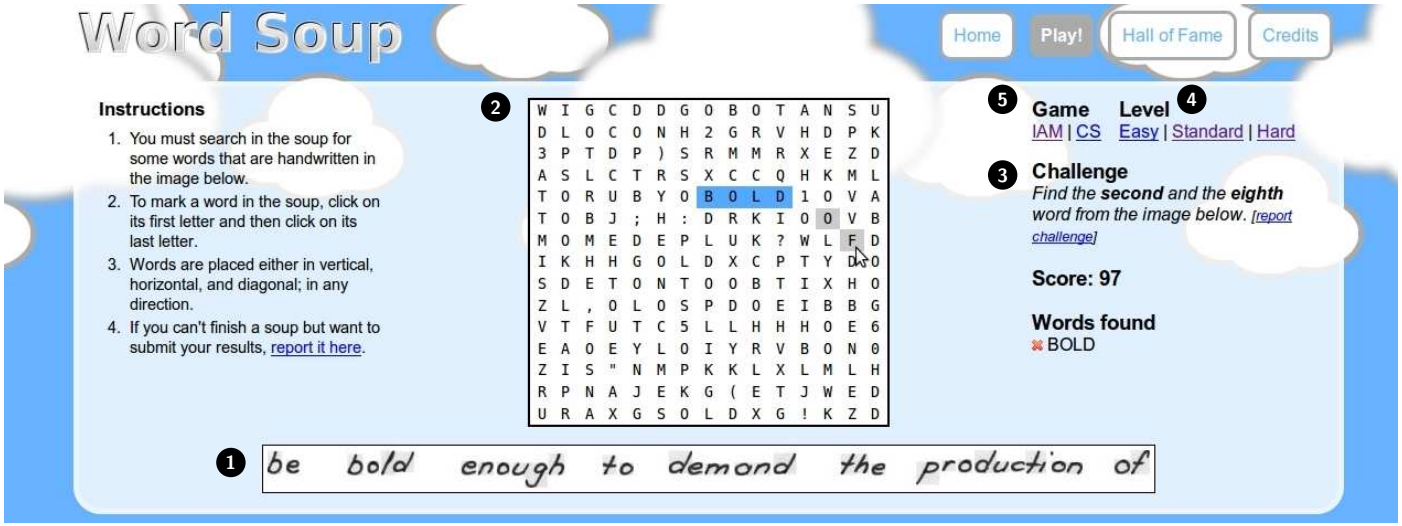


Figure 4: Game UI. A handwritten text image ❶ is presented to the user together with a word soup ❷ built from a pool of n-best lists. The system asks to find in the soup the words with less confidence ❸. The user can choose between 3 levels of difficulty ❹: easy (grids of 10 rows x 10 columns), standard (15x15), or hard (20x20). At present, the game is being used to transcribe 2 corpora ❺: The IAM database (example image line shown above) and the 'Cristo Salvador' book, a legacy Spanish document from the 19th century.

By now, a preliminary version can be accessed at http://cat.iti.upv.es/wordsoup/ (see Figure 4) so that it can be publicly tested before leaving the beta phase.

Acknowledgments

CasMaCat Project 287576 (FP7 ICT-2011.4.2), MIPRCV (CSD2007-00018) and iTrans2 (TIN2009-14511); and the CHI EA reviewers for their valuable comments and suggestions.

References

[1] Breuel, T. M. The OCRopus open source OCR system. In *Proc. SPIE* (2008), 0F1–0F15.

[2] Plamondon, R., and Srihari, S. N. On-line and off-line handwriting recognition: a comprehensive survey. *IEEE Trans. PAMI 22*, 1 (2000), 63–84.

[3] Robertson, S., Vojnovic, M., and Weber, I. Rethinking the ESP game. In *Proc. EA CHI* (2009), 3937–3942.

[4] Serrano, N., Giménez-Pastor, A., Sanchis, A., and Juan, A. Active learning strategies in handwritten text recognition. In *Proc. ICMI* (2010), 48–51.

[5] Toselli, A. H., Romero, V., Pastor-i-Gadea, M., and Vidal, E. Multimodal interactive transcription of text images. *Pattern Recognition 43*, 5 (2010), 1814–1825.

[6] von Ahn, L. Games with a purpose. *IEEE Computer 39*, 6 (2006), 92–94.

[7] von Ahn, L., Maurer, B., McMillen, C., Abraham, D., and Blum, M. reCAPTCHA: Human-based character recognition via web security measures. *Science 321*, 5895 (2008), 1465–1468.