Proceedings of 1st International Workshop in Computational Linguistics for Uralic Languages (IWCLUL 2015); ‹http://dx.doi.org/10.7557/scs.2015.2›

Infinite Monkeys of Babel – Crowdsourcing for the betterment of OCR language material

Wouter Van Hemel
National Library of Finland
Library Network Services
PL 26 (Teollisuuskatu 23)
FI-00014 University of Helsinki
wouter.van.hemel@helsinki.fi

Jussi-Pekka Hakkarainen
National Library of Finland
Research Library
P.O. Box 15 (Unioninkatu 36)
FI-00014 University of Helsinki
jussi-pekka.hakkarainen@helsinki.fi

November 7, 2014

Abstract

The OCR editor is the National Library of Finland's most recent foray into the budding phenomenon of crowd-sourcing. Under the motto of *many hands make light work*, users can swiftly correct the typical mistakes in OCR scanned text of source materials – often of challenging visual quality – using nothing more than their browser. Improving the quality and availability of the digital text would make it easier to directly study the original sources, and indirectly contribute to other tools depending on accuracy such as word list generators and dictionaries.

1 Introduction

The OCR editor project of the National Library of Finland hopes to be a useful tool to improve the quality of OCR digitisation and thus to bring works of diverse physical quality and sources into the digital age. The editor is an interactive web application, enabling many people to contribute simultaneously and revise the OCR text of source materials in the system.

The project has multiple goals:

- make sure the transfer of source material into the digital age does not in any way take away any of the quality of the original;
- make it easier to study the material by availability and dissemination (i.e. internet); ”editor as reader” or distributor;
- make automated corpora or word lists to improve the editor itself and other tools.

2 Software architecture

Let’s start with a brief technical overview of the tool’s architecture to put any technical terms in perspective.

The software consists of two major components: a **front-end** and a **back-end**.

The **front-end** – or editor itself – is a *JavaScript* application that maps ALTO XML attributes such as language and word coordinates to words in a web page. Through its two-pane graphical interface, users can correct wrongly recognised words side by side with the actual scanned imagery, or improve meta-attributes such as the language of individual words.

The more visual editor part is supported by a server **back-end** application in *Python*. This component serves the actual data to the editor and handles revisions, users, collection listing and other tasks. It also takes care of ”administrative” duties such as locking documents whose editing process has been considered finished so they can be exported – either as text or XML.

The more technically inclined readers can now feast their eyes on the graphical representation of the editor’s technical architecture to get a broad overview of its inner workings.

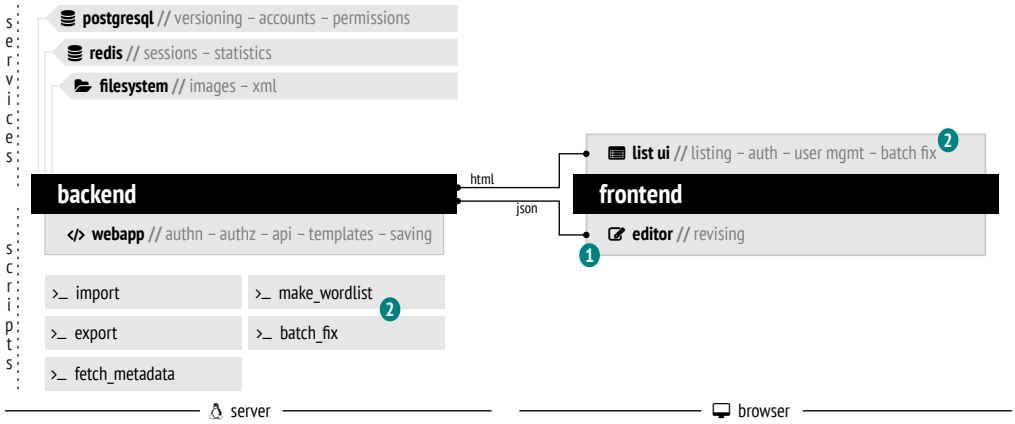

Figure 1: Overview of the technical architecture

3 Process overview

How do we get from faulty OCR text to a usable corpus?

- The editor's team receives – or produces themselves – ALTO XML from scanned images.
- These XML ALTO files and images are imported into the editor
- The text of the works is edited
- Optionally, corrections are checked by an authority
- The resulting text can be exported as XML or plain text, or can be used to create word lists and other derivatives

3.1 Manual labour

Manual labour here refers to the actual making of corrections in the online editor, marked in the figure by 1.

Users log in to the editor and after selecting the desired document are greeted by a two-paned page. On the left side a high-resolution image of the original work is shown; on the right-hand side the OCR text. Hidden behind the words in the editor

are the ALTO XML metadata attributes such as coordinates and language. On the top of the screen users have access to a toolbar that enables viewer functionality such as zoom and panel positioning. On the bottom an optional virtual keyboard layout can be selected to aid in the input of characters that might not be present on the physical keyboard. After the user has edited the text, they can save the version which then writes a revised ALTO XML file for the relevant pages and updates system metadata.

The editor aspect is already well developed and has been tested rather successfully in classroom settings – such as literature classes of Tampere university. The students mentioned the interactive component of working together and seeing immediate results as a more pleasant experience than individually working with paper copies. Experts benefit from the ability to off-load some of the burden and having to spend less time merging multiple contributed improvements manually.

3.2 Automated correction

This facet of the editor, marked by ❷, is still under development. Automatic correction is challenging because of the lack of support (stemmers, dictionaries) for a lot of the languages. Also, there is an inherent danger in automated mass-correction: changes ought to be reviewed by somebody intimately familiar with the language in question to avoid doing more harm than good. Faulty corrections could form a risk for corpora and word lists, especially because of the statistical relevance in languages with a small amount of source material.

For now the project's aim is to let researchers identify the n most common OCR mistakes and only replace those literal words specifically listed as problematic. The actual corrections happen as a *batch process* in the back-end only after being approved by a researcher in the site's interface. Replacing the n most commonly wrongly recognised OCR mistakes would in many cases lower the amount of often tediously repetitive manual labour significantly.

Other approaches are under consideration; this area is open for experimentation, as we can empirically determine which approaches are feasible and which ones do more than good. We hope to receive more input from researchers.

4 Crowd-sourcing and policy

Another challenge the editor tries – or needs – to solve is one of policy. When we think of a single researcher, located in their proverbial dungeon with nothing more than the soothing glow of their computer monitor, the answer is clear. But not so in an interactive multi-user environment. Who can edit texts? Does the corpus benefit from a fully open, wikipedia-style approach where anyone can make changes – and hence affect the outcome of corpus quality and by-products such as word lists? Or should the editing be limited to students of relevant languages? Perhaps only experts should be allowed to make direct changes to the corpus?

While the editor does not attempt to formulate the actual policy a collection or language community should apply – it is a mere technical tool – the granularity of access rights and restrictions can configured for collections according to the wishes of administrators of the system's collections. It is possible and advised that collection "owners" set their own policies and guidelines for their respective collections. The OCRUI editor offers a hierarchical permission system that allows designated "administrators" to manage the users and permissions of the collections they hold the sceptre of. These users and permissions then apply for all documents and collections below that point in the tree.

Documents can also be *'locked'* by administrators, disabling further editing. This makes it possible to export a completely corrected (or at least "known-good") version for use in other tools.

5 Conclusion

This concludes our brief overview of some of the technical aspects and design of the National Library's OCR editor. With this software, the team hopes to make a minor contribution to the field of linguistics and the preservation of Uralic languages – especially those endangered in our ever shrinking world.

To quote Frank Zappa: *"Writing about music is like dancing about architecture"*. If you are interested in the software or if this paper left you wanting, or perhaps the actual experience of using the editor would help to get a better overview of its use, its strengths and weaknesses – don't hesitate to contact the team for more information.

http://ocrui.lib.helsinki.fi/

Acknowledgments

Acknowledgement – in alphabetical order – goes to the following people connected with the project, either currently or in the past: Jussi-Pekka Hakkarainen, Esa-Pekka Keskitalo, Victoria Kurkina, Anis Moubarik and Juho Vuori.