Challenges in OCR today:

Report on experiences from INEL

Niko Partanen

31 January 2017

Introduction

This paper discusses some experiences and workflows from the INEL project (https://inel.corpora.uni-hamburg.de) (*Grammatical Descriptions, Corpora, and Language Technology for Indigenous Northern Eurasian Languages*). The long-term project is funded within the framework of the Academies' Programme, which is coordinated by the Union of German Academies of Sciences and Humanities. INEL focuses on languages from two language families: Uralic (Selkup, Kamas, Nenets and Komi) and Altaic (Dolgan, Evenki and Siberian Tatar). It also focuses on Ket, an isolate language.[1]

Digital preservation has many advantages for preserving written cultural heritage, although it is not a necessity in itself, as paper has also proven to be a long-enduring medium. However, digitalization increases the discoveribility and the possibilities for reuse of these materials. The next step is to bring these materials into common and scientific use. Opening these datasets is still in its early phases, and many related practices are underdeveloped. In Scandinavia, public and national libraries have recently begun to offer digitalized materials for public use, although the focus has been mainly on publications before 1910 [1] (http://www.dlib.org/dlib/july16/paakkonen/07paakkonen.html). Within language documentation projects, the concept of open data is barely discussed, and the most of the data is behind variously restricted licenses or with limitations to only academic use. This is unfortunate, since incorrectly structured and licensed materials will require considerable future effort and negotiation before they can be taken into active research use.

In order to do enable better usability, content must be accessible. In the case of written documents, this refers mainly to original text data. Optical Character Recognition (henceforth OCR) tools are irreplaceable in efforts to digitalize these resources. There are already broader and more systematic comparisons of different OCR tools available; see, for example, Tafti et. al. [2], but I aim to contribute to the discussion by pointing out certain particular needs of language

[1] Thanks to Rogier Blokland, Jussi-Pekka Hakkarainen and Alexandra Kellner for much appreciated feedback when writing this article

documentation, which may be relevant for the wider linguistic community as well. Within the INEL project [3] we have done an evaluation of currently available tools, and despite our satisfactory results, one must also acknowledge the existence of several alarming bottlenecks, unsuitable technical combinations and dead ends within the existing workflows.

There have been multiple large digitalization projects run by both academic institutions and libraries. Some of these, such as Endangered Archives (http://eap.bl.uk) run by the British Library, have focused mainly on audio and photographic media. Others, namely Fenno-Ugrica (https://fennougrica.kansalliskirjasto.fi/) and the Komi Kyv (http://komikyv.ru) maintained in Syktyvkar, are good examples of the systematic digitalization of written resources. Since many projects have produced vast quantity of scanned pages, the principal question is how we can use this data.

In INEL, we have been working with the digitalization of Selkup, Dolgan and Kamas materials. Our focus has been primarily on published fieldwork materials and other spoken primary data. With smaller languages, it is likely that the recorded and transcribed materials exceed the printed ones in size; generally speaking, the ratio of these two can be used as one way to estimate the usage situation or vitality of the language. The texts we work with in INEL tend to fall into two distinct categories: handwritten manuscripts and published printed transcriptions. This data is also unusually complex because the transcriptions are usually related to various archival audio recordings, but the relation may not be entirely transparent. For example, in the case of some Selkup transcriptions, the recordings available are different versions from those on which the existing transcriptions are based. The same transcriptions may have been published in several publications in somewhat different versions and using various transcription conventions. One recording may be stored in different copies in various archives.

This is also a very particular case for OCR, since the scripts used are often not included in available models, the number of diacritics (possible added by hand) can make the scripts extremely complex, and the fonts used may be highly customized. The OCR challenges could be very comparable to the OCR of printed early modern manuscripts, although there are also clear differences, especially concerning the volume of the texts. The copyright questions also get very complex when it comes to legacy materials, since the rights of the speakers, original researchers, archives and publishers may all interact in many ways, and may be impossible to reconstruct. That said, it seems we are still far away even from solving the copyright issues involved with a very simple set of data, so the question should not be made overly complicated at this point. It is obvious that as free license as possible, preferably Public Domain, would allow the widest reuse.

OCR workflow

Most of the time, the first step after digitalization is to find some solutions for performing OCR on the text. There are specific tasks for which a graphical user interface is particularly suitable. One of these is the post-processing of OCR results. Another is manual correction of the detected text regions. Both of these tasks require a visual comparison between the recognized text and the original image. There are plenty of different tools that offer the possibility of manually correcting OCR'd text, and virtually all of these have very similar user interfaces. The page image is on the left, the text field on the right, and moving the cursor around the text field has some kind of correspondence on top of the image. This raises the question of why so many projects have ended up deciding to develop new tools for post-correction. Fenno-Ugrica (https://github.com/NatLibFi/ocrui-frontend) has their own tool, as does Deutches Textarchiv (http://www.deutsches-textarchiv.de/doku/software). PoCoTo (http://thorstenv.github.io/PoCoTo) has been developed in CIS Munich within a larger EU project, and, in contrast to the others, it seems to have features related to batch correction of systematic mistakes.

Compared to this, the initial training of an OCR model is very well suited to a command line environment, especially with respect to the transparency and reproducibility of the training process.

Current solutions

The most critical commonly lacking features are:

- Lack of find and replace with regular expressions [4]
- Lack of good XML export

The first issue is connected to the automatization of the proofreading process, since some mistakes are always systematic. The export format is more serious issue. The following attributes would ideally available alongside the output format:

- Recognized word (or character)
- Coordinates on page
- Recognition confidence
- Some style information
- Information about the OCR system used
  - Version
  - System settings
  - OCR model used (with version)
  - Information about page processing, if that was performed

As a format, some kind of XML seems to be the current standard. However, certain formats often employed, such as plain text files, Word documents or PDFs are clearly not satisfactory. The National Library of Finland has also developed their own export format which contains ALTO XML, metadata and the text in plain text, and which was created to suit the needs of users who benefit from access to local copies [1] (http://www.dlib.org/dlib/july16/paakkonen/07paakkonen.html). It can be expected that new solutions for distributing this kind of data will emerge, since users will certainly have the need to download the data locally and to update it with the newest resources once they are added.

At the moment, the OCR market is dominated by ABBYY FineReader (https://www.abbyy.com/finereader) and Tesseract (https://github.com/tesseract-ocr/tesseract) [5]. The former is entirely commercial software, which makes it very problematic from the perspectives of open science and software. Tesseract, and its derivations such as OCRicola, are open source, but they offer only the core functionality of OCR recognition and thereby must be used in conjunction with other tools, and, at least in INEL, we have not yet found a satisfactory combination.

The main problem with ABBYY-based solutions is that the two versions of this program, Desktop and Engine, differ very greatly from one another and it is not possible to combine their functionalities. One is used through a graphical user interface, and the Engine version is meant to be used in more automatized manner from the command line. The main functionality that Abbyy Desktop lacks is XML export. This means there is no export format that would directly store the page position information. However, in the Desktop version, it is possible to train new language models. The problem is that the models trained in Abbyy Desktop cannot be used in Abbyy Engine. With rare scripts and endangered languages, the current language models are insufficient and some additional training is necessary in order to achieve a good output. It would already be a half-way solution to be able to use the models trained in Abbyy Desktop also in Abbyy Engine.

Why Abbyy Desktop does not have an XML export function is certainly a valid question. For a long time, I suspected this was due to corporate business interests: there are some reasons it is more lucrative not to offer XML export for a casual user. Perhaps it would allow them to do too much, instead of relying on the company's more expensive options. However, after using Abbyy for a longer while, I have begun to suspect that there is also a softer reason. Could it be that even Abbyy does not know how to store user-edited position information? When a user edits a text in Abbyy, there are moments where a word entirely loses its highlighting, as if the software would not have information about its position any longer. There are similar problems with the Revizor editor developed by National Library of Finland in the Fenno-Ugrica project. Especially at the line boundaries the program seems to lose its information about the correct line. This all signals that is must be extremely difficult to store word position information in user-edited text. The assumption is very logical in the sense that the user can easily copy and paste text around, delete some words and retype them, and somehow the software would still need to keep track of positions. If the task is too difficult for even the industry leader, we may in deeper trouble than we thought, as it is unlikely that smaller open source projects could reinvent this function. I remain optimistic that in the longer run, the open source community will emerge as the winner. One demand for this is certainly to stop building a new software for every project and cooperate more between different involved parties.

In INEL, we are currently working through Abbyy Desktop plain text file and HTML exports, which both keep the paragraph structure and page numbering. It has been possible to distinguish page numbers as sequentially growing numbers that are the only items on the line, and chapter numbers as those that are on the same line as the title or the first line of the text; in any case, not alone. However, these definitions are not very generic and also require manual adjustation. The data retrieved this way is aligned with Russian sentences using hunalign (https://github.com/danielvarga/hunalign), again with some manual supervision, and converted into a Toolbox file with a simple script that transforms the file structure to fit Toolbox demands. This is eventually imported into FLEx, with line, chapter and page information being stored on the note tier. The solution selected in INEL works for the careful digitalization of individual texts one by one, but would not be satisfactory for use on large scale digitalization projects.

OCR model training

OCR model training can be carried out at different levels. Abbyy has been used by us on the character and character combination level. Since the wordform information is not used at all, there is no interference from the Russian word frequency and character combination information that the software could use with Russian text. New combinations can also be trained, which can be especially helpful when teaching Abbyy to recognize rare and new characters that are not used in Russian orthography. The next step would be to feed the manually corrected word forms from the Dolgan corpus to the Dolgan Abbyy model. One benefit of this relatively shallow model is that it does not get mixed up by dialectal variation in the word forms. If the model has exact information about all possible word forms, the recognition quality can possibly be improved even further. However, according to Silfverberg and Rueter, word lists may still be superior to morphological analyzers [6]. It must also be taken into account that a more sophisticated model may also struggle more when the text it encounters is less standard.

OCRicola is a software built on the basis of Tesseract and has most of the same functionalities. The main difference between the tools is that OCRicola is able to use a finite state transducer (HFST) to generate the word forms used in recognition. In theory the morphological analyzer is also able to generate forms that are never present in the corpus but may occur in a new text.

Challenges with the Tesseract/OCRicola model are related especially to fonts and availability of a large enough training corpus. With OCRicola the question arises also about the existence of an available Finite State Transducer, but at least within the Giellatekno (http://giellatekno.uit.no) infrastructure a large number of Uralic languages are already covered.

Contrary to the Tesseract/OCRicola model, Abbyy takes care of most of the above-mentioned tasks by itself without user interference. This also makes the model development non-transparent. It is possible to use another Cyrillic model as a backbone and build upon that by adding the special characters needed for other languages. This makes Abbyy model training very broad and general, and allows even arbitrary matching between characters and recognized forms, which is a great benefit when compared to Tesseract/OCRicola, which demands that the trained character also be present in the font.

Indeed, recognition of languages such as English, Russian and German is almost flawless. In the same vein, it must be stated that even with new languages Abbyy often reaches very good results with a relatively small investment in the training. The only concrete problems with Abbyy are the lack of transparency behind the model training, lack of interoperability between different versions and some issues with missing export formats. As already mentioned, the lack of XML export format results in loss of information with the word and line positions on the page.

I have often encountered the idea that page position information is not necessary. There are several situations where this information is very critical. For example, different page elements such as page numbers are very easy to recognize from the layout when their position is known. The same goes for paragraph information, since line indent coordinates, when known, make paragraph detection rather reliable. Detecting it from unstructured lines is, on the other hand, rather hopeless.

It can be argued that there are cases where this information can be disregarded. When working with texts like the Bible or Koran, the original verse structure is enough to distinguish different elements. Some texts follow essentially the same model where each utterance can be identified as a part of a closed set. Some data in INEL is also like this, such as the translated sentence lists collected from many different languages in Siberia.   The rarer and more susceptible to subjective reading and interpretation the text is, more valuable it is to be able to reproduce the original text or to easily match the recognized words to positions on the page. Especially with old typefaces or handwritten manuscripts, the matter of interpretation tends to become more significant.

There are other good discussions about the steps of the OCR process; for example, Holley [7] goes through it step by step. Although that description is some years old, the principle has not changed, and table 1 in her paper provides a good overview of all necessary phases (http://www.dlib.org/dlib/march09/holley/03holley.html). I personally still find it questionable how well the current tools actually store information about the entire workflow. The measure of this would be whether we are able to associate an arbitrary token in the final recognized text with the matching coordinates in the original manuscript, microfilm or whatever the source has been. At least in regard to INEL workflows, the answer is unfortunately no.

There are experiences that a text resulting from OCR can already be corrected to some degree using simple ngram-based post-processing [8]. This is plausible, especially with data that is relatively well recognized, although the worst-recognized parts should probably be approached by redoing the OCR [9]. I would still be cautious about eliminating the need for manual post-correction, which, although impractical for millions of titles, can still be relatively cost-effective for hundreds or thousands of proofread books in a year.

Within INEL, the OCR is not a direct goal in itself, but the text extracted from the documents will be associated with the original recordings. There are also cases where the original recording has been lost or was never made. In the latter case, the document on which the OCR has been carried out is the closest primary data source which we can have. Thereby there must be a close connection between the original text document and the later XML file. The XML format currently used in INEL is the one produced by the EXMARaLDA (Extensible Markup Language for Discourse Annotation) System. The EXMARaLDA tools have a high level of interoperability with existing standards used in language documentation. At the Hamburger Zentrum für Sprachkorpora (https://corpora.uni-hamburg.de) we already have a long practice of publishing online different visualizations for spoken data (see Demo corpus on the HZSK website (https://corpora.uni-hamburg.de/drupal/de/islandora/object/spoken-corpus:demo)).

Publishing the texts is naturally much simpler when there are no annotations. One method of publishing is simply rendering the text as plain HTML. For a normally structured book, appropriate dose of CSS already makes the reading experience in this format very enjoyable. On the other hand, if it is possible to render the text into HTML, it is also possible to produce other formats, such as EPUB files, which are very good for reading on mobile devices.

There are also some types of documents that would benefit from a more complex rendering online. This is particularly the case with such manuscripts where even the reading of different characters may be debatable. Although I do not discuss the text recognition of handwritten documents here in depth, it should be pointed out that this is the type of document where some new facsimilé type of online publishing options would be the most needed. What I would most like to see is a set of tools that allow very easy and shallow rendering of XML and medium-quality images, instead of building the solution on top of heavy servers and byzantine architectures that require complex customization to fit the needs of specific projects. As I have discussed here, the needs of different projects may in reality be more similar than we think. The XML files could be even stored in public repositories such as GitHub, which would bring the work closer towards actually open and collaborative science.

In an ideal world, OCR model development could be done with open source tools for languages that have the following resources:

- A large corpus available with a permissive license
- A good collection of fonts that closely match the ones used in publications
- Frequency lists from possibly even a larger corpus
- Consistent spelling that does not vary across publications

It is noteworthy that for large languages such as English or Finnish, all these resources and conditions exist. However, for many of the smaller languages, these pose a large issue. Yet I'm not so sure if the distinction between bigger and smaller languages is always so clear. As the Komi Zyrian corpus is also now over 30 million words (http://komicorpora.ru), I see no problems in adapting almost any approach which works with larger languages to Komi as well.

Challenges for language technology

OCR'd texts pose many tasks for natural language processing. First of all, there is the need to transform idiosynchronic writing systems into the same, ideally phoneme-level representation. The text often needs some kind of a morphological analysis in order to be more usable for linguistic analysis. Applying different tools to this end is highly desiderable. For example, with digitalized Komi-Zyrian and Saami materials, the GiellaTekno tools have already proven very useful, and maybe something similar could be applied later also to INEL corpora. It can also be mentioned that many texts are attractive targets for named entity recognition (NER) since they often repeatedly cover the same prominent historical events in different languages and repeat the same historical figures.

One particularly relevant tiny task I can foresee is the automatic matching of recognized text with the original pages. This can be done by automatically comparing the proofread text masses with the XML output using word coordinate information. The two texts are naturally different, as one variant is not proofread and contains many errors that were never present in the other version. However, I believe this is a minor obstacle, and detecting the most similar pages (especially knowing that the following page normally continues the text) should be very doable in the immediate future. I have not found any existing solutions to this, which again may reflect the immaturity in the use of OCR'd research data in fields where high quality text output is necessary or a high priority.

Contribution to language documentation

In corpus linguistics and language technology, the texts are usually treated as the target of investigation in themselves. When applied to the endangered languages, there is often a need to personalize them further and add into metadata information about the writers and translators, among other relevant details. Instead of variables such as publishing time and place, there is the obvious need to know writer's date of birth and place of birth. These already tell quite a bit about the possible native dialect of the writer, which again can be one way to explain idiosynchronic features of individual texts. Often this kind of information is stored already in public repositories such as Wikipedia, so one additional question is how to employ these connections the most effective way.

In many endangered language communities, the pool of active and prominent speakers is small. It is highly likely that the same individuals have been writing to local newspapers and even published longer pieces of prose, and have also been speaking in public events that have been recorded. Alternatively, they may have ended up being recorded by different linguists or ethnographers, resulting in archive items that can be relatively easily associated with each other. Similarly, if the writers are deceased, it can be expected that members of the community will recognize and remember them.

Lastly, it could also be speculated that with many languages, the existing language documentation resources are the largest available bodies of texts in that language. In these cases, one could also imagine a scenario in which the language documentation data could be directly used to create an OCR model. This is somewhat counterintuitive, but text is text, and if the match between phoneme representation in written and spoken varieties is close enough, which it often is with languages with new orthographies, the chances of this succeeding are high.

Parallel texts also open up additional research possibilities. In the Fenno-Ugrica project in Helsinki, one of the main ideas has clearly been to collect resources that exist in several languages spoken in Russia. The result is that there are now publications which are translated to many Uralic languages. When the books are compared, the texts are immediately useful, since they share exactly the same structure, often on the sentence level with only small deviations.

Some resources are parallel in less obvious ways. There are historical events that have been covered in every newspaper in the Soviet Union and beyond. These are not parallel sentences as such, but still the texts are thematically linked in very intriguing ways. As mentioned, applying tools such as named entity recognition to this kind of data could be very interesting.

Even more abstract parallels can be drawn between spoken and written resources, since there are many narratives that have been recorded in different retellings and published multiple times. A good example of this is the Komi folktale Zarńia Bözha Kań, which is also connected to the Russian folktale Maša i Medved. It is published in two different variants which are both stored in Komi Nebögain collection in Syktyvkar, and has also been recorded by Erkki Itkonen, as told by Vasiliy Lytkin in 1957 (Kotus recording id: 1323_2az). It is still unknown how exactly this data can be used, but as far as I see, there are many open roads that can be explored.

Closing words

It is easy to talk about the differences in the data different projects deal with and produce, but there is a lot that is essentially the same in all OCR workflows. These similarities mean that most of the practices and solutions that work in one place will also work elsewhere.

There is some kind of a boundary between modern printed resources and early modern manuscripts and prints, and many tools are customized to work with one or another. I also outlined above that the needs for presentation are clearly different for the handwritten manuscripts and printed products. This said, the boundary is not that clearly cut. Researchers today continue to produce handwritten field notes, the handling of which does not essentially differ from what we have to do for the notes from Castrén's time. This is also obvious in the work we are doing in INEL.

Kuzmina's and Donner's fieldnotes have more in common than not, although they are separated by more than a century. Similarly, early phonogram recordings can be dealt with conceptually in a rather similar way to modern multimedia recordings.

References

1. Pääkkönen, Tuula, Jukka Kervinen, Asko Nivala, Kimmo Kettunen, and Eetu Mäkelä. 2016. Exporting finnish digitized historical newspaper contents for offline use. *D-Lib Magazine* 22.

2. Tafti, Ahmad P., Ahmadreza Baghaie, Mehdi Assefi, Hamid R. Arabnia, Zeyun Yu, and Peggy Peissig. 2016. OCR as a service: An experimental evaluation of google docs ocr, tesseract, abbyy finereader, and transym. In *Advances in visual computing: 12th international symposium, isvc 2016, Las Vegas, NV, USA, December 12-14, 2016, proceedings, part i*, ed. George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Fatih Porikli, Sandra Skaff, Alireza Entezari, et al., 735–746. Springer International Publishing.

3. Wagner-Nagy, Beata, Hanna Hedeland, Timm Lehmberg, and Michael Rießler. 2015. INEL: Eine Infrastruktur zur Dokumentation indigener nordeurasischer Sprachen. In *Konferenz "forschungsdaten in den geisteswissenschaften (Forge 2015)": 15. bis 18. september 2015 an der universität hamburg. Lecture2Go*. Hamburg: Projekt Geisteswissenschaftliche Infrastruktur für Nachhaltigkeit (gwin).

4. Mancinelli, Tiziana. 2016. Early printed edition and ocr techniques: What is the state-of-art? Strategies to be developed from the working-progress mambrino project work. *Historias Findigas* 4: 255–260.

5. Smith, Ray. 2007. An overview of the Tesseract OCR engine. In *Proceedings of the ninth international conference on document analysis and recognition (icdar)*, 629–633. The IEEE Computer Society Conference Publishing Services.

6. Silfverberg, Miikka, and Jack Rueter. 2014. Can morphological analyzers improve the quality of optical character recognition? In *Proceedings of 1st international workshop in computational linguistics for Uralic languages*.

7. Holley, Rose. 2009. How good can it get? Analysing and improving OCR accuracy in large scale historic newspaper digitisation programs. *D-Lib Magazine* 15.

8. Arkhangelskiy, Timofey, and Maria Medvedeva. 2016. Developing morphologically annotated corpora for minority languages of Russia. In *Proceedings of corpus linguistics fest 2016. Bloomington, IN, USA, June 6–10, 2016.*, ed. Sandra Kübler and Markus Dickinson, 1–6. CEUR Workshop Proceedings 1607. Bloomington: Indiana University.

9. Kettunen, Kimmo, and Tuula Pääkkönen. 2016. Measuring lexical quality of a historical finnish newspaper collection – analysis of garbled OCR data with basic language technology tools and means. In *Proceedings of the tenth international*

conference on language resources and evaluation (lrec 2016), ed. Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, et al. European Language Resources Association (ELRA).