OCR: Unleash the hidden information

Anssi Jääskeläinen, Liisa Uosukainen, South-Eastern Finland University of Applied Sciences / Digitalia research center, Mikkeli, Finland

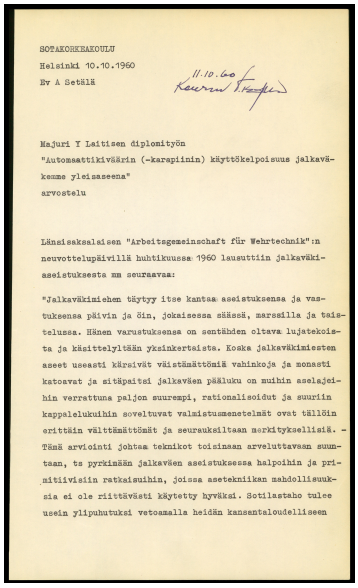
Abstract

Most of us, even though it is not very rational, commonly take pictures of texts. In a conference it is very unlikely not to see participants taking pictures of presentation slides. Similarly, national archives scan documents without doing an OCR (Optical Character Recognition). Resulting image, in spite of its resolution, quality or file format is not searchable by its content. Unless someone types in a large amount of metadata according to Dublin Core for example. While this is an acceptable behavior in an archival world, an average people is willing to fill the maximum of five fields. Therefore a clear need for an easy and most importantly a free way to get pictures, scanned documents etc. to be fully searchable is a mandatory need.

A Digitalia research center has been working on to create an effective workflow that automatically analyzes the document content, generates OCR information as well as gets the most relevant keywords for the content. Furthermore, the workflow produces an archival graded PDF/A file if requested by the user. This workflow has been fully integrated into our Citizen Archive solution to handle everything automatically in the background. With this sophisticated solution usability, findability as well as reusability of the preserved content will be greatly increased. In short this equals better archival user experience and less manual work to be done for both the archivist and the end user.

Motivation and Problem

National archives, as well as other archives are full of analog material, such as text, books, photographs and maps. In addition, there are digital archive collections, which continuously keep on growing in consequence of digitization of old analog material. Digitization procedures vary among archives, but according to our knowledge from the field, most of the content is still preserved as plain images. A good example of this behavior is the Karelian Database case in Finland which contains millions of scanned handwritten double pages from the old parish registers from the lost Karelian region. Every single double page has been digitized for more than 20 years ago as a tiff image. Now the project has been going on for more than 15 years to manually write the data from the digitized pages into the Karelian Database [3], which currently contains more than ten million hand fed records. The previous scanning example happened a long time ago, but even now with multiple OCR technologies available, the digitization process is still done without content / text recognition [1] unless specifically requested by the customer. Naturally this will cost extra, but with available technology it should not. As a proof of claim, Figure 1, which is a page from a Master's thesis from the 1960s', is presented. According to Figure 1 metadata, it was scanned on the sixth of September 2017 by the National Archives of Finland. The publicly available presentation version is 300 dpi jpg file with 2158x3583 pixels and the preserved version is most likely a higher resolution tiff file. Obviously there is no way to find this thesis according to its content since OCR information does not exist. However, if this thesis would have been OCR'd, it would be very simple for search engines to index the content which would make this file findable e.g. via Google search. This same issue has been raised into discussion by Moss and Endicott-Popovsky who ask the justified question why the ubiquitously available OCR is not applied to digitized records like it is applied for example in banks, and Google Books [5]. They state a fourfold answer:



SOTAKORKEAKOULU
Helsinki 10.10.1960
Ev A Setälä

11.10.60

Majuri Y Laitisen diplomityön
"Automaattikivääriin (-karapiinin) käyttökelpoisuus jalkaväkeemme yleisaseena"
arvostelu

Länsisaksalaisen "Arbeitsgemeinschaft für Wehrtechnik":n neuvottelupäivillä huhtikuussa 1960 lausuttiin jalkaväkiaseistuksesta mm seuraavaa:

"Jalkaväkimiehen täytyy itse kantaa aseistuksensa ja varustuksensa päivin ja öin, jokaisessa säässä, marssilla ja taistelussa. Hänen varustuksensa on sentähden oltava lujatekoista ja käsittelyltään yksinkertaista. Koska jalkaväkimiesten aseet useasti käsittävät välistämättömiä vahinkoja ja monasti katoavat ja sitäpaitsi jalkaväen pääluku on muihin aselajeihin verrattuna paljon suurempi, rationalisoidut ja suuriin kappalelukuihin soveltuvat valmistusmenetelmät ovat tällöin erittäin välttämättömät ja seuraukseltaan merkityksellisiä. - Tämä arviointi johtaa teknikot toisinaan arveluttavaan suuntaan, ts pyrkimään jalkaväen aseistuksessa halpoihin ja primitiivisiin ratkaisuihin, joissa asetekniikan mahdollisuuksia ei ole riittävästi käytetty hyväksi. Sotilastaho tulee usein ylipuhutuksi vetoamalla heidän kansantaloudelliseen

***Figure 1**. Page from a master's thesis (1960) http://digi.narc.fi/digi/view.ka?kuid=65173788*

1. Handwritten documents
2. Unstructured poor quality documents
3. Volume of the data
4. The amount of errors that would occur during the OCR process

It must be admitted that handwritten and fraktur documents are a true problem, thus OCR engines are not good with it by default. These documents are out of the scope of this paper, but a lot of research has been, and is being going on in this area. One of such projects is the European Commission funded Horizon2020 READ project[1] which continues the work done in FP7 project TransScriptorium[2]. They have achieved quite good results in the area of recognizing these text types.

Unstructured data is not an obstacle to any modern OCR engine and the quality of the document can be enhanced programmatically for example with ImageMagic. Therefore the second given answer is nonsense.

Third obstacle according to Moss and Endicott-Popovsky[5] is the volume of the data. It is true that enormous amounts of data exist. However, so does computing power! Our test server with 32 simultaneous threads can OCR approximately one page per two seconds when run in parallel mode. With single thread, the run time would be on average around 50-60 seconds per page. Even if the total page mass would be 50 million pages for example, it would only take around three years to process it all with our test server. Full scale optimized production environment with 512 parallel threads for example would handle this task around 16 times faster. Therefore also the third answer is nonsense.

The fourth and final statement by the authors [5] is just strange. They state that the amount of errors is a reason not to conduct OCR. While this might be OK in highly academic research where accuracy is everything, for a normal daily usage we claim that something with mistakes is far better than nothing. Furthermore, mistakes first needs to be done and found before those can be fixed.

Some actors in the field have already realized the potentiality of this previous information that remains "invisible" inside the documents. National Archives of Australia for example encourage the use of OCR but it is not enforced [2] and NARA accepts OCR'd text if the resulting PDF quality is not degraded [7]. Finally, in a survey conducted by the National Archives of Finland, which results are unfortunately not yet publicly available at the

1 https://cordis.europa.eu/project/rcn/198756_en.html
2 http://transcriptorium.eu/

time of writing, 55% of the respondents, which were mainly national archives have utilized some kind of OCR method for the captured digital images. So there is light at the back of the tunnel.

Average citizens

The problem is not only within archives. In conferences and seminars, many participants take photographs of presentation slides. Later on, if they remember that some slide contained nice information to be shared with colleagues or to be utilized, it will be very frustrating to go through all the images to be able to find the one with that nice information. Similar situation happens when scanning a photograph, a contract, or a newspaper article. The result will be an image without any information about what is in it. The IS&T copyright form, for example was required to sign and scan. The problem further culminates when the number of scanned or photographed documents raises. How to find the file that contains the needed information from the file/folder mess?

Solution

Our suggested solution is the Citizen archive, a centralized digital repository for the personal archivists to collect and manage their personal and familial data. An embedded workflow engine enables the designing of extra tasks, such as OCR script or content analysis for the pre-ingesting or ingesting processes.

The Citizen archive has been developed in the South-Eastern Finland University of Applied Sciences during the past few years but its roots go back to 2013. The first version, called OSA (Open Source Archive) was the foundation for the Citizen archive. It was developed and launched in 2013 and is a service oriented solution suitable for a long time preservation [4]. The OSA has since been applied by civil sector organizations and non-profit associations and is now being modified more to fully accommodate personal archives as well.

The Citizen archive provides a modern web user interfaces to manage, search and discover the archival content. According to the access level, the user is allowed to ingest and modify the material in the archive, as well as search or browse the content. The archive administrator manages the user accounts and other administrative tasks, such as submitting the proposed records for destruction.

Our pilot projects have proven that a large number of scanned image collections with a limited, mostly technical metadata ingested into an archival solution is not enough for a good user experience [6]. Automatically extracted technical metadata is information about the file itself. It covers the dates, times and location details as well as technical data of the used capturing device. Still, none of this important metadata provides information about the content of the file.

OCR script, described thoroughly in the next chapter, is used to enhance the pre-ingest process of the Citizen archive. At first the primary specification is done by the archivist himself by selecting the type of the record among the available metadata models. The Citizen archive has determined metadata models for documents, pictures, letters, e-mails according to the Dublin Core and Finnish JHS and SÄHKE specifications. OCR script generates the full text content of the document. The procedure concurrently analyses the text and generates a set of keywords, which are embedded in the document metadata. The Citizen archive's pre-ingest process captures the metadata automatically from the file and maps the metadata values to the selected metadata model. Finally the

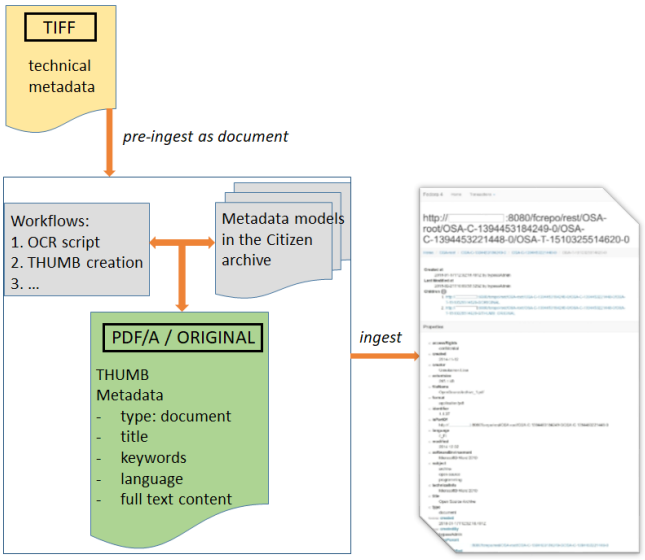


***Figure 2**. The simplified ingest process in the Citizen archive which produces Fedora resource*

metadata and the full text content are ingested to the repository with the original file and the thumbnail image of the first page. This simplified process is presented in Figure 2. The Citizen archive requires only the minimal set of metadata to be mandatory available in the ingest process. The descriptive metadata can naturally be collectively enriched later on by the authorized users of that particular archive.

Using OCR technology and automated content analysis the document management, findability and accessibility in the Citizen archive are enhanced. The archive is capable of producing better search results compared with the plain tiff documents. OCR'g enables the full text search of the digital content in the Citizen archive. The keyword search finds items according to the subject metadata field that contains terms from a controlled vocabulary. Valid keywords among the digital record types and full text content improve the accessibility of the archival content and more organized collections of the personal records are reached in the repository.

OCR Approach and Results

The OCR approach by Digitalia is fully based on existing open source products that are bound together by using Python script. At the time of writing the workflow supports all the formats that ImageMagick can handle, including PDF documents. Simplified workflow is presented in the next listing

1. Python script handles multiprocessing and reads the files to be OCRed
2. Supported files are analyzed, enhanced if needed and converted to PNG images with ImageMagick convert command
3. Converted files are fed to the Tesseract 4 OCR engine which currently is equipped with Finnish, Swedish, English and German best "traineddata" files. Tesseract is configured to produce PDF files with embedded OCR information

4. Created individual PDF files are combined with Ghostscript and converted into an archival graded PDF/A format.

Benchmarks were run with multiple large files including a master's thesis which contained 91 pages. Total runtime for the master's thesis was 213 seconds. A noteworthy but unrelated fact is that before the Spectre and the Meltdown vulnerability fix the run time was 181 seconds. Tesseract 4 which we are using, is still under development and it is known to be slower than the version 3. Still, we managed to achieve the average speed of about 2.35 second per page. 20% of the time per page was used by Imagemagick conversion and 3% by the Ghostscript. This leaves 77% of runtime for Tesseract which equals about 1.8 seconds per page. Not bad at all and we are certain that by compiling Tesseract with optimized settings as well as enhancing the workflow we could reduce this time with our current 32 thread server into about one second per page. The OCR speed could also be greatly enhanced by using the fast traineddata files but this would reduce the accuracy of the detection.

For the sample presented in Figure 1 in the first page the accuracy of the OCR was superb 99.65% (four mistakes and 1155 characters). Naturally the recognition rate depends on the quality of the original file so it might vary greatly. But for the average computer or typewriter written text without badly faded parts, skewed text or compression artifacts, the recognition rate can be expected to be quite high.

The whole automated OCR process has been integrated into our Citizen Archive solution. Every ingested document is OCR'd if requested by the user. It is also possible to OCR process already ingested images and documents. This OCR'g makes the ingesting of scanned or photographed documents into a repository more automated, thus the OCR'd file can be further analyzed to produce information about the content itself, such as special keywords, most common words and word class analysis.

Content analysis approach

Even though, merely having an OCR'd content is a great benefit, from our opinion it is just the beginning. Having this precious information available makes much more interesting possibilities available, such as automated metadata creation and content overview reports which enhances the archival user experience even further.

Let's assume that you have thousands of scanned and OCR'd records and it is time to import those into repository. You could do a batch run to inject all records with the same basic metadata while ingesting but it is not very descriptive. During the ingest process the OCR'd content is most likely indexed which partially solves the problem. However, if the same person is mentioned for example in half of the records, how can the search function know which document you tried to find without more accurate content descriptions? During the ingest phase, weeks could be used to go through the files and manually give them content related descriptive metadata, but this is likely a tedious and frustrating job to do.

After submitting the abstract, we have been working on this particular issue by enhancing the automated content analysis part. At the time of writing the analysis can produce both the machine and human readable form of the results and is capable of analyzing PDF files containing either English or Finnish. The technical back-end of the analyzer is based on multiple Python libraries such as nltk (Natural Language Toolkit), polyglot and libvoikko. The latest is a special library developed for the Finnish language. Following list introduces the phases done during the analysis

1. File type is recognized with DROID or by Linux file command. If neither is available a filename extension is used.
2. Language of the file and a confidence percentage for the language is checked with Python language detector
3. Content of the file is checked word by word.
    1. Total words and unique words are calculated
    2. The base form of the word is resolved by using either libvoikko or nltk
    3. The class of the word is resolved with polyglot
    4. Stopwords are recognized and marked based on multilingual stopword lists
    5. Named entities are recognized with polyglot
    6. The class hierarchy of a word is resolved by using local install of the Finnish thesaurus and ontology[3], which supports Finnish, English and Swedish
    7. Known related terms for the word are resolved by using the same installation as the step above.
    8. Top keywords are calculated based on the occurrences of the base word and word class.
4. Human readable report is generated and metadata is embedded in the file

As a result of running this script the found aspects are embedded in the metadata of the file, from where those are accessible by metadata extraction tools such as exiftool, Apache Tika or Adobe Acrobat.

Secondly, a human readable HTML report is generated by using the same results. The intention of the report is to give an overview of the content to any interested reader without a need to browse through the whole document. As a proof of concept we took a screen shot from the Archiving 2018 conference main page, then ran it through our OCR script after which we generated the human readable report from the content of the produced OCR'd PDF file. Figure 3 presents the results. Top of the figure shows the OCR'd file opened in Linux PDF reader with the word "*digitazation*" searched and the bottom of the figure presents the automatically generated HTML report. This is only our opinion, but the content of the report corresponds quite well with the document content and it gives a good overview. The proof of concept files can be downloaded[4] as a zip package.

Naturally, it has to be accepted that an automatically produced metadata is not and will newer be perfect and can sometimes even be totally misleading. We still claim that some kind of automated metadata creation at the time of ingest even with errors is better than forcing users to manually enter hundreds of required metadata fields, thus no one does that willingly. The same aspect has been recognized by NARA, for instance already in 2005. The authors have written "*Descriptive and structural metadata creation is largely manual; some may be automatically generated through OCR processes to create indexes or full text" [7]*

3 https://finto.fi/en/
4 https://digitalia.xamk.fi/archiving2018/ocrsample.zip

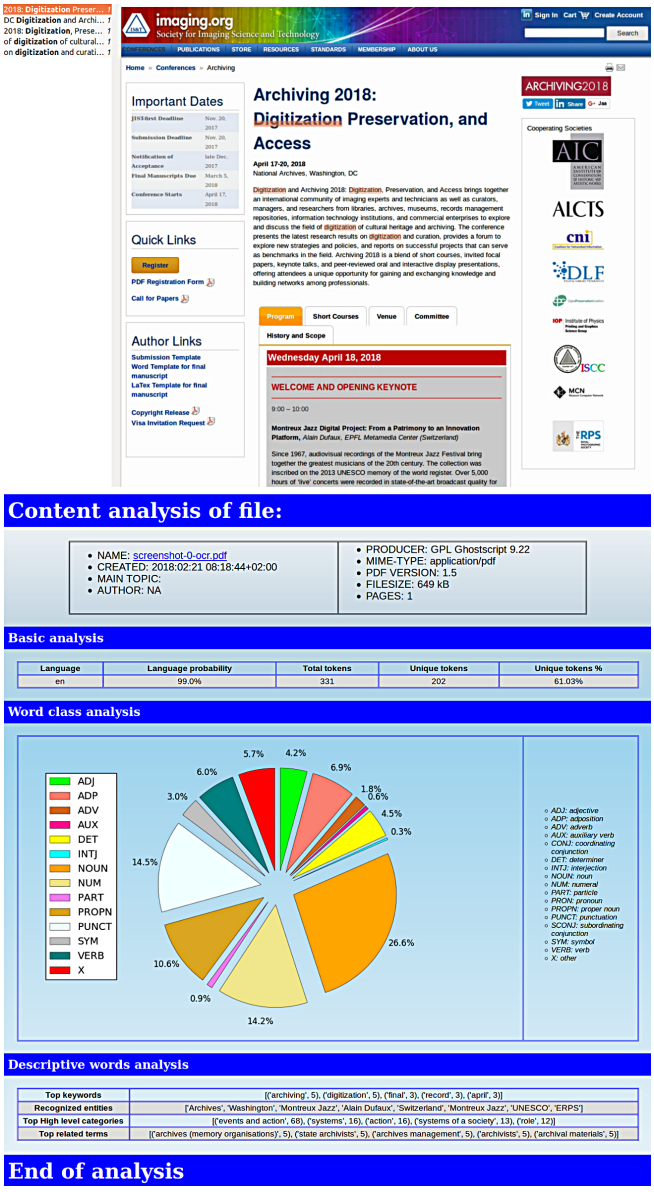


***Figure 3**. OCR'd screenshot and the generated content report*

Conclusions

In this paper, we have shown that archives are full of texts that are preserved in a form of an image. This directly means that the content of these files cannot be found by using text based searches unless the preserved documents are equipped with metadata. In an archival environment filling in metadata is a mandatory task but an average people will not fill hundreds of metadata fields. Do they even know what metadata is or how to add metadata in to a PDF file for example? Most likely not. This situation happens for example when a conference participant takes a photograph of a presentation slide. Most of us have done it and will continue to do so in the future. The resulting image without any information about its content sooner or later becomes irrelevant thus nobody founds it and even if it is found the utilization of the information is awkward.

The solution developed at the Digitalia research center at South-Eastern Finland University of Applied Sciences solves this problem. The developed workflow automatically detects, reads, analyzes, converts and OCR reads these documents and produces an archival graded PDF/A file that contains OCR information and content related metadata. Our solution also analyses the OCR content and creates both machine readable metadata as well as human readable report of the content. When this solution is combined with our Citizen Archive solution things like automated archival metadata and auto classification becomes possible.

Capturing automated keywords from the scanned files enable the rough classification of the digital material. Some document types such as brochures, invoices and meeting minutes, can be automatically pre-defined according to the automatically found keywords. Moreover if OCR process have recognized sensitive things such as credit card numbers, personal information or maybe writings that would be classified as racist nowadays, the digital document can be pre-classified as a confidential material. Finally, this automatically added descriptive metadata reduces the need for the manual description of digital objects later on.

OCR technology even with Open Source exists, so why not taken it into use. We have shown that only true issue in the field is handwritten or Fraktur documents, everything else can be managed with computing power, novel solutions and know-how. Even Google Docs online service performs OCR on uploaded images. So technology is available but why is it not utilized? Thus OCR is invaluable in making the information within a digitized magazine, journal and newspaper collections far more accessible and making information retrieval much faster.

Future work

New natural language detection tools are being developed constantly. We will continuously monitor the development and switch into new technology if a need should arise. Meanwhile we are aiming to make the keyword detection even better and for this task we would like to ask your help. So please tell us what metadata you would like to have automatically generated based on the recognized OCR content. We will then do our best to make those dreams come true.

Sensitive words detection is still undone but hopefully before the conference we are able to achieve some results also on this aspect. We will also start working with open source image recognition libraries in the near future. Demonstration versions of our solutions can be tested via digitalia.xamk.fi, but at the time of writing a user name and password is required to access the demonstration applications. These can be obtained by contacting us via website.

References

[1] C. Adams, Making Scanned Content Accessible Using Full-text Search and OCR, 2014 avail. https://blogs.loc.gov/thesignal/2014/08/making-scanned-content-accessible-using-full-text-search-and-ocr/

[2] http://www.naa.gov.au/Images/scanning-specifications_tcm16-93663.pdf Australia National Archives, 22 Aug 2013

[3] J. Räisä, M. Loponen, The modernization, migration and archiving of a research register, Proc. 7th DLM Forum Triennial, pg. 24-27. (2014).

[4] M. Lampi, O. Palonen, Open Source for Policy, Costs, and Sustainability, Proc. Archiving 2013, pg. 271. (2013).

[5] M. Moss, B. Endicott-Popovsky, Is Digital Different? How information creation, capture, preservation and discover are being transformed, Facet Publishing, 2015
[6] P. Uotila, Using a professional digital archiving service for the construction of a family archive, Proc Archiving 2014. pg. 188 (2014).
[7] S. T. Puglia, J. Reed, E. Rhodes, Technical Guidelines for Digitizing Archival Materials for Electronic Access, Digital Library Federation, 2005

Author Biography

Anssi Jääskeläinen has an IT MSc. (2005) and PhD (2011) from Lappeenranta University of Technology. He has an extensive knowledge of IT, user experience and usability. His current research interests are in virtualization, format migration and Python development.

Liisa Uosukainen has M.Sc. (Tech.) from Lappeenranta University of Technology (1994). She has years of experience in software development. Her current interests are in digital data and digital archiving.