THE UNIVERSITY *of* EDINBURGH

OCR Report

July 2023

Completed by OCR Intern Ash Charlton for Gavin Willshaw (Digitisation and Digital Engagement Manager) and the Cultural Heritage Digitisation Service at the University of Edinburgh Library.

Contents

1. Introduction to OCR

Optical Character Recognition (OCR) is the most commonly known method of text extraction from digitised documents used in the cultural heritage sector. It is a process that transforms images of text into a machine-readable format. Traditionally, OCR uses technology to digitally scan text and identify letters individually, therefore recognising one character at a time. Advancements have been made over time that introduce aspects of machine learning into OCR which change this dynamic slightly, which will be explored in more detail later in this report. This report explores OCR software options broadly, in addition to past, current and future proposed OCR processes and workflows that the University of Edinburgh library may introduce.

History

OCR grew in popularity in the cultural heritage sector from the 1990s, but has its origins far earlier, in the nineteenth century as reading devices to assist visually impaired people.[1] From 1809 to the 1930s, multiple iterations of assistance devices for visually impaired people were created, but it wasn't until the 1940s, with increased demand for data entry, that OCR really increased in popularity.[2] Rising to meet business demand for data entry and information processing, OCR technology developed through the second half of the twentieth century, with uses in the U.S postal system and Armed Forces.[3] Its ability to recognize texts was eventually applied to heritage text materials and it is 'an essential resource' in cultural heritage organisations .[4]

In the early days of heritage digitization and text extraction, OCR processing was done on image files that were not originally intended for this purpose, as seen in early digitization projects. Christy et. Al note:

> '[…] the path to digitization was not ideal: these documents were imaged in the late 1970s, transformed into microfilm during the 1980s, and the microfilms digitized in the 1990s. Because of the state of reproductive technologies during the late 20th century, as well as the circuitous path to digitization (through microfilm), the image quality is very poor and bitonal, with no greyscale images available. Furthermore, the original documents themselves, printed with premodern technologies, pose problems even for human readers of their pages, but much more so for optical character recognition (OCR) engines. For example, printed characters were not perfectly situated on a baseline, blackletter fonts were used, ink bled through the paper, and the typeface was broken and overworn. Moreover, these documents are aged: pages are missing, ripped, or blotted with handwritten marginalia and spilled ink.'[5]

OCR has improved significantly over the years as it has been adapted and implemented for different purposes, and it is a topic that is widely discussed in academic circles. OCR can be run on both printed and handwritten text, although printed text will yield more accurate results due to a higher level of standardisation due to typesetting and/or consistent fonts. Handwritten text is far more likely to feature inconsistencies and variation in size and shape of lettering, thus making it more difficult to determine patterns.

OCR requirements

The success of text recognition and extraction is dependent on several factors encompassing the original text materials and the images of the text. For successful OCR processing, images should meet the following criteria:

- Be brightly lit and in focus
- Have a good contrast between the text and page, for example, black text on a white or light-coloured page

---

[1] Herbert F. Schantz, *The History of OCR, Optical Character Recognition* ([Manchester Center, Vt.] : Recognition Technologies Users Association, 1982), 1, http://archive.org/details/historyofocropti0000scha.

[2] Schantz, 1–6.

[3] Schantz, 17.

[4] 'Issue 13: OCR', Europeana PRO, accessed 28 June 2023, https://pro.europeana.eu/page/issue-13-ocr.

[5] Matthew Christy et al., 'Mass Digitization of Early Modern Texts With Optical Character Recognition', *Journal on Computing and Cultural Heritage* 11, no. 1 (7 December 2017): 6:1-6:2, https://doi.org/10.1145/3075645.

- Any text in the image should be straight in the photograph and not skewed
- High-resolution images on capture (this cannot be changed after the photo is taken)
- Stored in a format that keeps the image quality as high as possible such as TIFF files; JPG files are smaller which compress data and lose image quality
- Kept in a useable format; OCR programs will accept certain file formats such as JPG, TIFF, PNG or in some cases PDF. It is best to check in advance to ensure you have a useable format for your specific software[6]

Even if all these criteria are met, there may be unavoidable aspects of the materials themselves that may hinder OCR capabilities and text quality. These include:

- Complex layouts, such as newspapers as there are multiple different text blocks to identify that the programme could be more likely to struggle with
- Combinations of printed and handwritten text
- Archaic letters that can be mistaken for others (likely to be more common in older printed documents), such as the long s [ſ]
- Older printed documents, especially handset type is more likely to feature inconsistencies in text due to inconsistent spacing or dropped lines of text
- Languages other than English may be of concern for some OCR engines, although many are now suitable for use with a multitude of languages

And finally, a suitable infrastructure is needed to support the use and or preservation of extracted text in a way that meets the requirements of both the user and institution. These needs will dictate the form that OCR outputs will take; for example, downloadable datasets may be stored and accessed as plain text files, or forms that present images with the text may incorporate a searchable text overlaid pdf, or parallel views of images and text transcriptions.

Figure 1 (below) demonstrates the ability to use the 'ctrl + f' shortcut to search PDF files – in this case, searching for the word 'debating' in *The Student* newspaper in the UofE collections that is clearly visible on the page, and returning a match for the word from the overlaid text that is not visible to the user (highlighted in grey).

[6] Centre for Data, Culture & Society, the University of Edinburgh, 'Text Extraction & Preparation', Managing Digitised Documents Pathway, n.d., https://www.cdcs.ed.ac.uk/training/training-pathways/managing-digitised-documents-pathway/text-extraction-preparation.

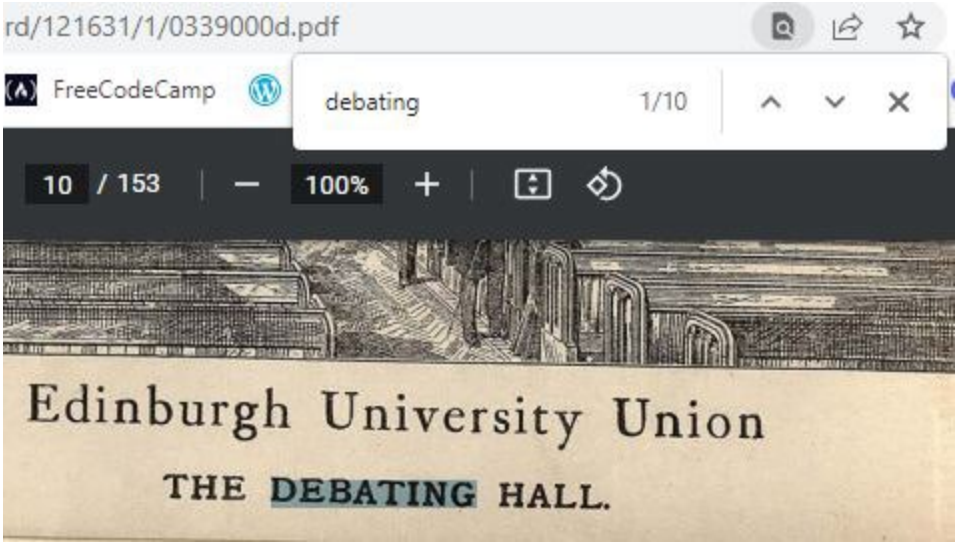

Figure 1[7]

Figure 2 is taken from the National Library of Scotland's dataset repository, the *Data Foundry*. Here the text extracted via OCR is available to download as a plain TXT file alongside image files and metadata of items from different digitised collections.

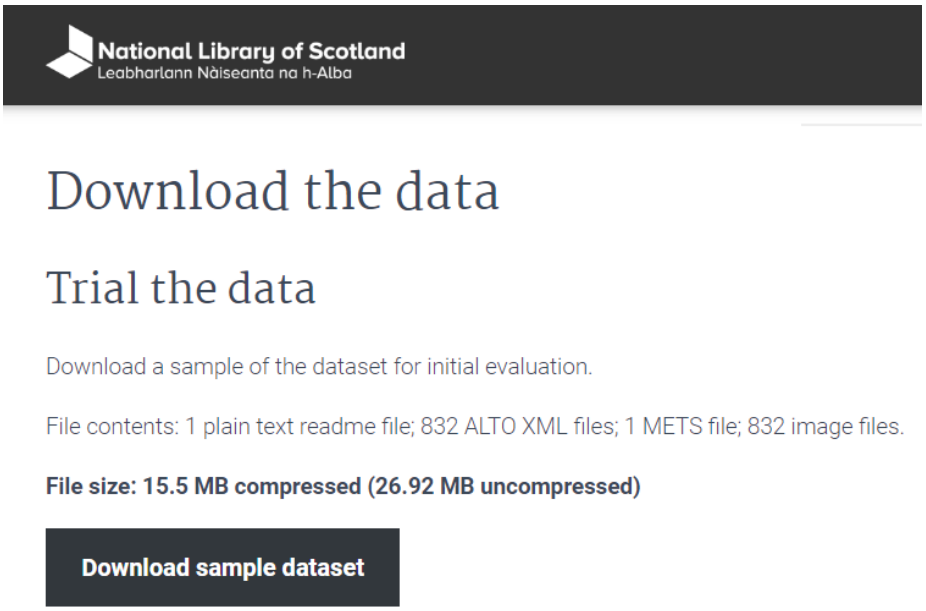

Figure 2[8]

Figure 3 shows an alternative presentation of extracted text from the National Library of Scotland's Digital Gallery, with an image and the OCR-generated text next to it, so the actual text is clearly visible to the user.

[7] 'The Student, Issues 1-12', accessed 29 June 2023, https://openbooks.is.ed.ac.uk//record/121631?highlight=*:*.
[8] 'A Medical History of British India – Data Foundry', accessed 2 July 2023, https://data.nls.uk/data/digitised-collections/a-medical-history-of-british-india/.

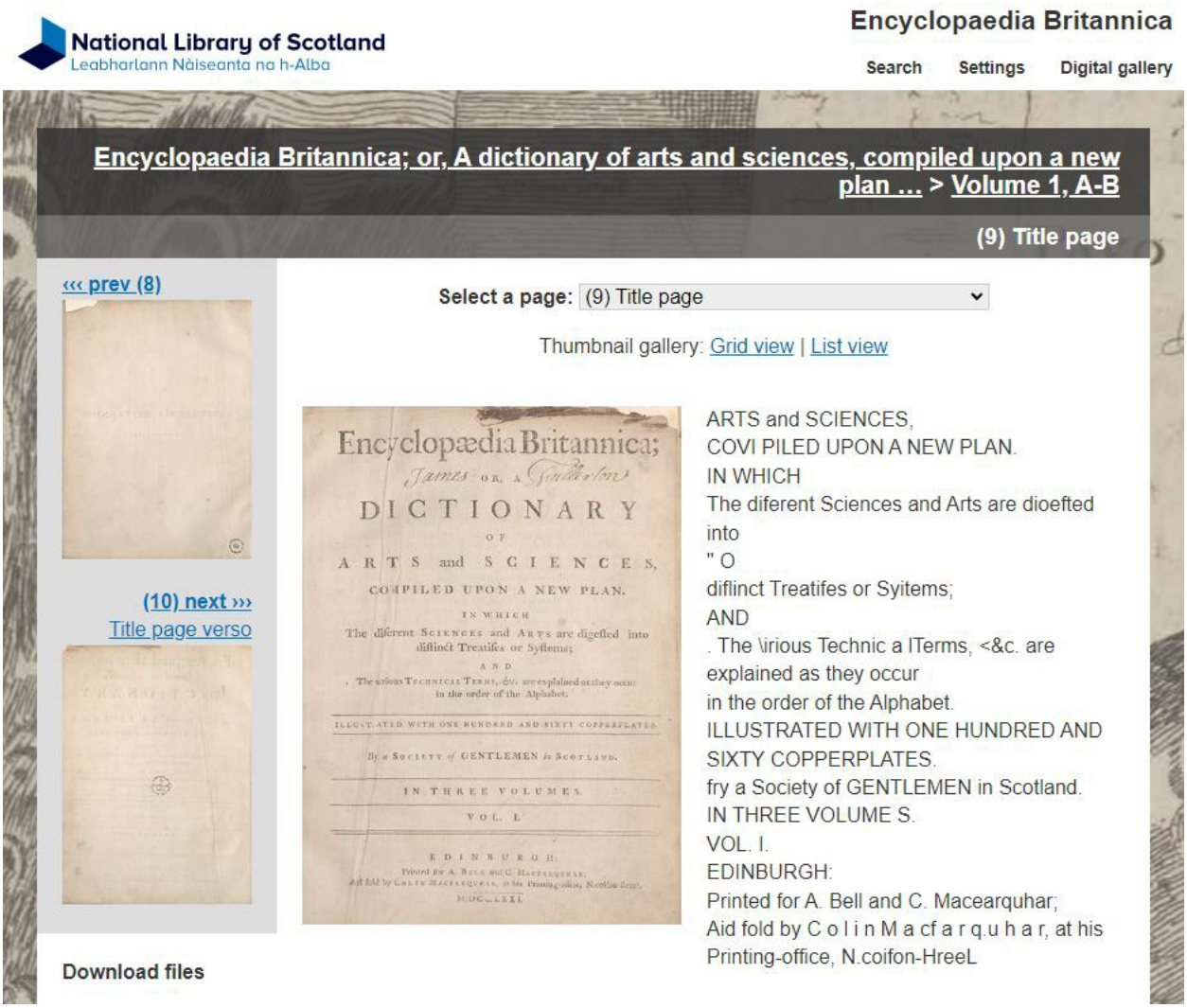

Figure 3[9]

These three examples highlight the differing ways that OCR-generated text can be presented to users, and consequentially, the varied ways in that it is required to be created and stored for use (PDFs, TXT files, etc).

The OCR process

Traditionally, OCR was carried out by the engine identifying the text letter by letter, studying the shape and then matching these against a database of stored characters to produce a final result. These key steps are explored in further detail below, whereby the image of text is broken down to build up the text:

[9] '(9) Title Page - Encyclopaedia Britannica; or, A Dictionary of Arts and Sciences, Compiled upon a New Plan … > Volume 1, A-B - Encyclopaedia Britannica - National Library of Scotland', 9, accessed 29 June 2023, https://digital.nls.uk/encyclopaedia-britannica/archive/188082824?mode=transcription.

Character Segmentation: Text regions are identified within the text (for example single text blocks, columns, or even further breakdown into paragraphs or other text regions that may appear on the page. Within the identified text regions, OCR algorithms separate individual characters or groups of characters. This step involves segmenting the text into distinct units for further analysis.

Feature Extraction: OCR algorithms extract features from the segmented characters to represent their unique characteristics. Features can include stroke direction, line thickness, curvature, or texture patterns. Various mathematical algorithms or machine learning techniques are employed to extract these features – this is a feature that is being drawn on more as the field of text recognition develops.

Character Classification: The extracted features are used to compare and match the segmented characters against a predefined set of character models or templates. OCR engines employ machine learning algorithms, such as neural networks or statistical models, to classify the characters based on the extracted features.

Since OCR became more freely accessible and widely used in a range of settings, the technology has advanced, and more engines incorporate machine learning techniques and natural language processing models, which can recognise words and draw on more 'intelligent' processing, rather than relying solely on individual character recognition.

Decision making

The decision-making behind OCR is tied to several different factors, including institutional resources, and the purpose the text is being created for. Resourcing factors include budget, staff numbers and time allocations for digitization (OCR can be integrated into the overall digitization process but may also be treated as a separate step in the workflow, depending on the software used). How the generated text will be used, displayed and stored will also influence the decision-making process, as different outputs may be required to create a dataset versus a pdf file with overlaid text, or integration into a DAMS.

2. Different OCR software & tools

As OCR has developed and the field of text extraction has grown, more options for text processing have become available, both for heritage materials and for wider applications. A quick search for 'OCR software' returns hundreds of thousands of options – a choice that quickly becomes overwhelming. So how do you decide which software best suits your needs? This section outlines some of the options available.

For the University of Edinburgh theses project (see below) a proposal was written that included a comparison of a range of OCR software available, so this report will not duplicate all the information presented there, as it makes an excellent comparison already, however this report will touch on some of the more relevant software mentioned by the report to provide further detail.[10]

[10] 'Theses OCR Software Proposal', n.d., https://uoe.sharepoint.com/:w:/r/sites/Digitisation/_layouts/15/Doc.aspx?sourcedoc=%7B799F568D-D686-42F4-B4DC-B80AD1615830%7D&file=Theses%20OCR%20Software%20Proposal.docx&action=default&mobileredirect=true.

Free and open-source text extraction software

There are several free and open-source options available, however these are so numerous and available in so many formats and programming languages that the choice can be overwhelming. Just searching OCR libraries available through the Python Package Index, there are 855 available packages. Searching OCR on GitHub returns thousands of results (https://github.com/topics/ocr), although some of these appear to be less relevant for cultural heritage than others. For the more popular free and open-source options such as Tesseract, there is heavily detailed documentation, and its popularity and ease of use means that there is generally more written (for example in articles and in comparisons between software) and therefore potentially a better support base for trouble-shooting issues.

The Theses OCR software proposal offers an overview of other free options, including the pros and cons. Whilst there are many more than are included on the list, it would be impossible to list all of them, so the data from the proposal is included below:[11]

Software	Pros	Cons	Cost	Latin?
Tesseract	-Highly accurate -Easy to learn -Works with Python -Free & open source -Best free software for handwritten text -Can automate OCR process -Tesseract gives better results than ABBYY FineReader after image processing*	-Accuracy issues with handwritten text but it is still best free software for this. -Does not handle complicated or noised page layout well.	Free	Yes
SimpleOCR	-Fast and lightweight -For both Windows and Mac -Can batch scan files -Free software	-Does not support Latin	Free	No (Tesseract 3.02)
Calamari	-Straightforward to use -Free & open source	-Tricky dependencies -Can only do text recognition so requires another engine to increase contrast, segment, etc. -Not as accurate as Tesseract or SimpleOCR - Not as good at dealing with handwritten text. As it can only do text recognition it is unable to improve quality.	Free	
OCRopus	-Free & open source	-Requires higher resolution images than other software -Lots of errors if below 300 dpi -Difficulties with sideways or upside down documents -Not as accurate as Tesseract or SimpleOCR	Free	
EasyOCR	-Free & open source	-Not as accurate as Tesseract or SimpleOCR	Free	Yes

[11] 'Theses OCR Software Proposal'.

Tesseract[12]

Of the options listed above, Tesseract is one of the more popular and cited sources. Tesseract is a free open-source option that can be used in multiple programming languages, including Python and R, and gives good quality OCR results. It started development in the 1980s and has been reiterated and developed since then. As such, it has a strong user base and community of contributors, meaning that there is a level of community support from other users, especially useful when troubleshooting, but also a wealth of documentation. Tesseract can recognize more than 100 languages 'out-of-the-box', and can also be trained to recognize other languages.[13] Tesseract supports a range of image formats, such as PNG, JPEG and TIFF, as well as the variability to offer a range of output formats, including plain text, hOCR (HTML), PDF, invisible-text-only PDF, TSV and ALTO (the last one - since version 4.1.0).

The Tesseract documentation outlines that better text extraction results can be obtained from better or higher quality images, however does not appear to have specific text cleaning options built in – likely as its uses will be very varied and text cleaning can be a subjective processes dependent on materials.

As with many OCR engines, it does not support handwritten text recognition particularly well, but does work successfully with many printed texts, dependent on image quality. As an open-source option, any staff using Tesseract would require training and ideally have at least a basic understanding of the programming behind the engine to work with any issues encountered, and to maintain updates of versions (where deemed necessary), and make informed decisions about adjustments to the software. Reviews of Tesseract have identified that it can offer better results than ABBY FineReader after image processing was carried out.[14] Tesseract is the software identified in the theses OCR proposal as the most viable option for carrying that project forwards.[15] It can also be integrated into workflow processes such as Goobi.

Paid text extraction software

As noted in the previous section, the theses OCR software proposal includes a breakdown of advantages, disadvantages and costs for paid OCR options too:

Software	Pros	Cons	Cost	Latin?
Google Cloud Vision API	-Highly accurate -Much better at recognising handwriting than main competitors Tesseract and ABBYY FineReader. - More accurate than Tesseract in identifying words if text very small, even after additional pre-processing.	-Moderate cost, particularly compared to free software	$1.50/1000 units up to 5 million/month	Yes

[12] 'Tesseract OCR', C++ (2014; repr., tesseract-ocr, 24 April 2023), https://github.com/tesseract-ocr/tesseract.

[13] 'Tesseract User Manual', tessdoc, accessed 7 July 2023, https://tesseract-ocr.github.io/tessdoc/.

[14] Marcin Heliński, Miłosz Kmieciak, and Tomasz Parkoła, 'Report on the Comparison of Tesseract and ABBYY FineReader OCR Engines', n.d.

[15] 'Theses OCR Software Proposal'.

Kofax OmniPage	-Highly accurate -Works on both Mac and Windows	-Expensive	Starts at $4,999. Runtime licenses start at $2100 for 500k pages	Yes
ReadIRIS	-Works on both Mac and Windows -Fast and accurate -One-off license fee	-License fee is relatively high when compared with free software alternatives	£49 for one license	Yes
Adobe Acrobat Pro DC	-High accuracy rate -Easy to learn and use -Low monthly fee but pricey if using annually	-Does not recognise Latin -License fee is high for what you receive -Many extra features are not needed for this project (i.e. editing PDFs)	£19.97/month	No
Nanonets	-Easy to use interface -Comprehensive technical documentation -Integration available through APIs	-Expensive if using Pro (over 100 pages) -Does not recognise Latin -Geared towards corporate environment for processing forms and financial info	USD$499/month $0.1/page	No
ABBYY FineReader	-Highly accurate -Good for retaining document structure -Can detect old European and Fraktur/Gothic fonts -Reasonable annual fee	-Poor accuracy with small font	£59/year for Mac £84/year Windows	Yes
Transkribus	-Has built-in OCR-engine (ABBYY FineReader) and can run text through this first -Focuses on both single characters and sentences for context -Good accuracy with handwritten historical documents -Works best with purpose-built models	-Difficulties when pages have multiple fonts or languages -There are better tools for typed text	EUR €18/120 credits, custom-tailored packages for institutions	Yes

ABBYY FineReader[16]

ABBYY FineReader is currently used in the Cultural Heritage Digitisation Service and is often cited in articles and blog posts as used by other heritage institutions. It is currently integrated into the workflows with the book scanners, and produces text of a reasonable accuracy level, whilst recognizing text blocks consistently and with reasonable accuracy. It is easy to access, maintain, troubleshoot due to its popularity, and is one of the cheaper options (per license).

[16] 'ABBYY FineReader PDF Software: Open, Read & Edit PDFs', FineReader PDF, accessed 11 July 2023, https://pdf.abbyy.com/.

Transkribus[17]

This is also used withing the University Library, as a tool to aid with transcription of the Lyell Notebooks. It is styled as Handwritten text Recognition software (HTR), but can be used on both handwritten and printed text. It yields a good level of recognition on handwritten texts, especially when using custom-built models (which can be trained by the individual), but does have other combination models available for users too. It has good success rates on printed text too, however there is a need to purchase credits to run the text recognition, meaning that costs can quickly increase, depending on the number of credits used. This is likely a more viable option for projects than widescale library use.

Post-extraction processing

Much like deciding on text extraction processes, post-extraction processing for corrections is subjective and is dependent on many factors. The type of post-processing will depend on both the materials and the resources available within the department – post-processing can be labour and time-heavy to set up, so may not be a viable option for many cultural heritage institutions.

Out-of-the-box options

Out of the box options are available but many are aimed towards researchers rather than being designed for larger scale institutional operations and successful in integration into larger workflows.

VARD2[18]

Vard2 is a java operated text processing tool, aimed at Early Modern English texts. This is intended as a pre-cursor tool to text analysis to yield more accurate results. As a free, out-of-the-box option, it comes ready to use in its downloadable interface, and offers a degree of adaptability with manual and automatic processing options. It could potentially be a time-consuming option for text correcting, depending on the data volume and manual input, however may be of benefit for small-scale projects where text requires some corrections.

overproof[19]

overproof is a web-based system that runs on servers, allowing OCRed text to be uploaded and cleaned. It claims to 'reduce the number of articles missed by a keyword search due to OCR errors by over 50%', however, it does not yield entirely accurate corrections – to guarantee so would require additional manual checking and processing. This is also a paid software that operates on words processed per month.[20]

Open-source options

There are many freely available programming options available online that can assist with post-processing, however it is unlikely that text correction options you find on the web will be a fix-all solution. The ultimate purpose of the text extracted will determine the type of post-text corrections carried out. For example if you are creating a digital version of the text to accurately represent the original text in the book, you may retain words that are split by hyphens at the end of lines. If you are creating a text that does not require being formatted the same as the original text, the decision could be made to remove the

[17] 'Transkribus', READ-COOP, accessed 16 January 2022, https://readcoop.eu/transkribus/.

[18] 'VARD - About', accessed 9 July 2023, https://ucrel.lancs.ac.uk/vard/about/.

[19] 'OverProof - Home', accessed 9 July 2023, https://overproof.projectcomputing.com/.

[20] 'OverProof - About and Help', accessed 9 July 2023, https://overproof.projectcomputing.com/about#pricing.

hyphens (or other page features that may prevent easy reading) in order to convey the text in a more clear and readable format. It would be impossible to list all these possible options here as there are many variations and ways to engage with the open-source options, because they are often created and adapted to meet specific needs.

Regular expressions (essentially short-hand coding prompts) would be one way to execute simple changes, but require programming knowledge and understanding of how code functions in order to write them to adapt specifically to the specific institutional or collections needs. The Programming Historian has an introductory course to regular expressions (or regex, as it is sometimes called) in Python, complete with example code and text regex has been applied to.[21] There are lots of other online resources to find different snippets of code that can be used for text cleaning.[22]

GitHub provides a wealth of information in scripts written and shared by individuals. Many of the scripts available on GitHub target specific problems of the texts they were designed for, and so may not be directly applicable to your own collections, therefore careful evaluation is needed. One example shared by Ted Underwood states it is '[m]aybe, at best, it's a collection of resources you could cannibalize to build your own workflow' – indicating the need to adjust and adapt.[23]

Pre-existing tools such as the Edinburgh Geoparser have components of programmed text clean up that are explained in their documentation – again, these require a level of understanding and the capability to adapt these to use, however there are options out there.[24]

Although there is a strong community of open-source text cleaning options, these are never a straightforward selection and application to individual processes and require tweaking to work specifically with certain collections, based on individual needs. Most of these are designed by people working with specific collections or projects, and therefore tailor them specifically. Although some of the openly available programming cleaning options will be usable, they may require someone with advanced programming knowledge to implement them successfully and select the appropriate options. As such, although these types of options offer flexibility, there are skills and time requirements to make them work.

3. OCR approaches by other institutions

There is little transparency of OCR processes in the cultural heritage sector, so it is difficult determining the processes used by institutions without a comprehensive survey of the sector. One of the few articles outlining library-specific processes is by Olson and Berry, based at the University of Western Ontario, that compares different OCR software (primarily Adobe Acrobat, ABBYY FineReader, Tesseract).[25] As

[21] Laura Turner O'Hara, 'Cleaning OCR'd Text with Regular Expressions', *Programming Historian*, 22 May 2013, https://programminghistorian.org/en/lessons/cleaning-ocrd-text-with-regular-expressions.
[22] Priyanka, 'Common Regular Expressions for Text Cleaning in Python', *Medium* (blog), 18 November 2022, https://medium.com/@priyankads/common-regular-expressions-for-text-cleaning-in-python-5a13b832d340.
[23] Ted Underwood, 'DataMunging', Python, 8 April 2023, https://github.com/tedunderwood/DataMunging.
[24] 'Geotagging — The Edinburgh Geoparser 1.3 Documentation', accessed 9 July 2023, https://groups.inf.ed.ac.uk/geoparser/documentation/v1.3/html/geotag.html#the-tokenise-component.
[25] Leanne Olson and Veronica Berry, 'Digitization Decisions: Comparing OCR Software for Librarian and Archivist Use', *The Code4Lib Journal*, no. 52 (22 September 2021), https://journal.code4lib.org/articles/16132.

Olson and Berry observe in their introduction, literature on OCR tends to fall broadly into two categories; short blog posts offering an overview of software, or more technically-focused digital scholarship perspectives.[26] Although value can be found in some of these, these posts or articles are aimed at a slightly different audience and may be difficult to apply to a library setting. There is a distinct lack of literature from libraries and cultural heritage professionals regarding their text extraction methods, so the examples below are only a small sample of what is discoverable on the web.

Where information is available regarding OCR and text extraction processes, this is usually from larger, national libraries or organisations that may have access to more funding avenues, or significant research projects, again made possible through grant awards. There were some University library examples, however these were difficult to find. Speculatively, this may be due to institutions using standard options such as ABBYY FineReader, which produce 'good enough' results for a reasonable price and there may not be the scope or resources in institutions to explore other options fully. For a more comprehensive study across the UK, or on a larger scale, a survey aimed at cultural heritage institutions would yield a more wide-ranging an accurate representation of practice across the sector.

Text extraction

Texas A&M University: Early Modern OCR Project: Tesseract

From 2012 the Early Modern OCR Project (eMOP) used open source software Tesseract on early modern texts (1473-1800). The project primarily focused on improving the OCR-generated from images with sub-par quality from early digitization projects that used microform images from the 1970s and 1980s that were then digitized in the 1990s, producing poor-quality image reproductions that were subsequently OCRed.[27] The result of the project was using a combined approach with Tesseract and Ocular to generate more accurate text transcriptions, with the aim to incorporate another OCR engine to further enhance quality, although the project page has not been updated since 2015 so it is unclear if further work incorporating this was undertaken.[28] They have, however, released workflows and instructions for operating their chosen engines and the function of the project, as well as collaborative outcomes with ECCO and EEBO that allow for viewing of the OCR.[29]

University of Cincinnati: Mixed Approach

A 2020 experiment by the digital imaging team at the University of Cincinnati compared six different text extraction options to determine which would produce the best results. These six options were ABBYY, Google C.V., Transkribus, Equidox, Acrobat Pro and Tesseract.[30] They conducted the comparison on six different printed documents of differing text qualities and fonts to give an overview of how each of the software options performed. Their comparisons found that Transkribus performed the best overall, using a

[26] Olson and Berry.

[27] Christy et al., 'Mass Digitization of Early Modern Texts With Optical Character Recognition', 6:1-6:2.

[28] The Initiative for Digital Humanities, Media, and Culture, Texas A&M University, 'Early Modern OCR Project', accessed 28 June 2023, https://emop.tamu.edu/about.

[29] 'Outcomes', eMOP, accessed 9 July 2023, https://emop.tamu.edu/; 'EMOP Workflow: Lucidchart', accessed 9 July 2023, https://lucid.app/lucidchart/7da90ad2-ab8b-4cb7-ba01-9ddaad1f5385/edit.

[30] Sidney Gao Coordinator Digital Imaging, 'OCR: Who Does It Best?', University of Cincinnati Libraries Digital Collections Documentation, 7 August 2020, https://uclibs.github.io/digitization-workflow/2020/08/07/ocr-comparison.html.

publicly-available model (thus reducing the need to create and train their own model which is time-consuming), however Google Cloud Vision performed the best on their document with faded type, and this was the second most highly ranked option. Adobe Acrobat Pro and Equidox ranked as the two options providing the lowest accuracy levels and the blog post observed that these two would not be used on collections.

The blog details the difficulty of selecting text extraction tools, as accuracy is not the sole factor to consider in the process, such as text volume and workflow integration. They summarise: '[a]lthough we know that Tranksribus did the best job on typewritten documents, it also can't efficiently handle the volume of PDFs we create. We will, however, keep it in our toolkit for special situations'.[31]

University of Western Ontario

As mentioned earlier in the report, the University of Western Ontario conducted a study to evaluate OCR software on their materials, primarily Adobe Acrobat, ABBYY FineReader, Tesseract – three common options that have been referred to in much of the literature so far. Their study found ABBYY FineReader 'to be the most useful when the goal is to OCR a document that will be uploaded into a repository or database, where you want the text searchable and a high accuracy for the OCRed words.'[32] Their recommendation was to use Tesseract for 'a text file for a digital humanities project, or to add a transcription or full text metadata into a repository' and also note its success with recognising information in columns and tables.[33] Although its uses for working with PDFs were recognized by Olson and Berry, Acrobat's OCR function was not recommended.

State Archives of Zurich: Transkribus

In 2019 the State Archives of Zurich ran a project that transcribed 50,000 pages of eighteenth-century meeting minutes from the Zurich Council.[34] They trained their own Transkribus AI model by transcribing 203,189 words, producing a model with a CER of 4.80%, although their project page estimates error variance of 5-8% due to handwriting and image quality.[35] The 50,000 pages took three years to complete; significantly faster than manual transcription would have been for the project. There is no reference of cost in the project summary, although this would have incurred costs with Transkribus.

National Archives of the Netherlands: Transkribus

The National Archives of the Netherlands, the country's largest archive, used Transkribus to generate text from 3 million images.[36] Like the State Archives of Zurich, they created a custom AI model, which resulted in a CER of 7%, based on 6000 pages of training data. The model they created, of Dutch handwriting from the seventeenth to nineteenth centuries is available for other users to use on their own materials.[37]

---

[31] Coordinator.

[32] Olson and Berry, 'Digitization Decisions'.

[33] Olson and Berry.

[34] 'How the State Archives of Zurich Published 50,000 Pages with Read&search', READ-COOP, accessed 28 June 2023, https://readcoop.eu/success-stories/state-archives-of-zurich/.

[35] 'Zürcher Ratsmanuale 1700-1798', accessed 28 June 2023, https://ratsmanuale-zuerich.transkribus.eu/.

[36] 'Transcribing 3 Million Scans at the National Archives of the Netherlands', READ-COOP, accessed 28 June 2023, https://readcoop.eu/success-stories/national-archives-of-the-netherlands/.

[37] 'Dutch Handwriting 17th-19th Century', READ-COOP, accessed 28 June 2023, https://readcoop.eu/model/dutch-handwriting-17th-19th-century/.

Text correction/post-processing

Much of the documentation about post-processing methods and corrections came from researchers and academics working with material after it had been made available by institutions, however there are some examples of post-processing. There may be fewer references to libraries carrying out this work as the text quality is left at a 'good enough' quality.

Texas A&M University: Early Modern OCR Project: Tesseract

The Early Modern OCR Project (as outlined above) incorporates text correction into its workflow, incorporating EEBO and ECCO documents into crowd-sourced correction tool, TypeWright where texts can be corrected by individuals or groups of people.[38] This is by no means the only example of this type of project, but it is a largescale example that published their workflows.

Text presentation

State Archives of Zurich

The State Archives of Zurich have hosted their outputs in a Transkribus sister site where words can be searched, matching sections in text for user understanding, displaying generated text next to images of the original texts.[39] This is evidently a specific example, directly in partnership with Transkribus.

National Library of Scotland

The National Library of Scotland have made their text available in multiple formats; especially their digital gallery and Data Foundry, which hosts open Datasets. The Entry requirements demonstrates examples of how these files can be viewed; in the digital gallery there is an option to view images and text transcriptions side by side, download a PDF, or simply view the images. The ability to view the text alongside the images allows for direct comparisons between them. In the Data Foundry the texts have been created as open and free downloadable datasets, each of which gives an indication of the expected text quality of the data. These range from 'Original OCR: no clean-up', to 'Cleaned up OCR', to other descriptors such as 'Handwritten Text Recognition (HTR)-generated'. These offer an insight into the data, however unless a user already knows what OCR is, there is no explanation of the implications this has for the text.

4. Library & University Collections (L&UC) past & current OCR approaches

The University of Edinburgh's Library and University Collections have used OCR in different ways across various projects in the past. This section outlines the methodologies and approaches used in past projects and current operations and workflows with OCR application to materials.

Session Papers

The Session Papers (papers from the Scottish Court of Session) were a project worked on following an image capture digitization project in 2016-17. There were approximately 250,000 items but a sample of

[38] '18thConnect - TypeWright', accessed 11 July 2023, https://18thconnect.org/typewright/documents.

[39] 'Read&Search: Kanton Zurich Staatsarchiv', accessed 11 July 2023, https://ratsmanuale-zuerich.transkribus.eu/search?t.

4,000 pages were used to evaluate effective digitization techniques, with the aim to a mass digitization project to follow.[40] The papers were in bound volumes with a shelfmark, but no further item data to offer insights into what each volume contained. There were several limitations with the papers, including the use of Scots language, physical damage such as creases and holes in the paper, and challenging material features including low quality paper and print.[41] Conservation concerns were also raised due to the size and bindings of the volumes, where tight bindings would make imaging (and therefore OCR) difficult without disassembling some volumes.

The aim of the OCR for this project was to use the extracted text to assist with metadata creation and to make the Session Papers searchable.[42] The project used Python (programming language) and OpenCV (a library of programming functions) to detect text blocks and extract the text.[43] This tailored code created for the project is available via GitHub.[44] The final project report outlined the experimentation with different OCR engines for a first pass of OCR, which included OCRopus, CuneiForm, Tesseract v3 and Tesseract v4-alpha.[45] [46] The report identifies Tesseract v4 and OCRopus as providing the best results, of which Tesseract v4 was selected due to its continued development and potential wider applications. The project identified that the text generated was not good enough to meet the project goals (including metadata extraction), so the decision was made to work on post-processing methods to improve the text.

The post-processing to improve the extracted text was done with the Edinburgh Geoparser.[47] This included joining words impacted by end-of-line hyphenation and correcting long 's' forms that had been misrecognized as 'f', e.g. 'mafter' instead of 'master'.

Overall, the OCR for the Session Papers was heavily tailored to the project, accounting for the specific material type, and incorporating text post-processing clean-up, as well as the basic extraction. The pilot of the Session Papers OCR project was successful, but identified the need for significant staffing resourcing in order to successfully complete the digitization of the entire collection of papers.

---

[40] Information shared in a PowerPoint provided by Norman Rodger.

[41] Information shared in a PowerPoint provided by Norman Rodger.

[42] Mike Bennet, 'Automated Item Data Extraction from Old Documents', *University of Edinburgh Library Labs Blog* (blog), 23 June 2017, https://libraryblogs.is.ed.ac.uk/librarylabs/2017/06/23/automated-item-data-extraction-from-old-manuscripts/.

[43] 'Python.Org', Python.org, 22 June 2023, https://www.python.org/; 'OpenCV.Org', OpenCV, accessed 29 June 2023, https://opencv.org/.

[44] Mike Bennett, 'Sp-Experiments', Python, 8 June 2017, https://github.com/mbennett-uoe/sp-experiments/blob/94ce9184cd1c0c3a5bdb720ba77f3ceaa6d76317/sp_crop.py.

[45] Information shared in the Session Papers final report (2019) provided by Norman Rodger.

[46] 'OCRopus OCR Engine(s)', ocropus.github.io, accessed 29 June 2023, https://ocropus.github.io/; Алексей Черемных, 'Cognitive OpenOCR - Распознавание Текста', Свободные программы для Windows - КонтинентСвободы.рф, accessed 29 June 2023, https://континентсвободы.рф/cognitive-openocr-raspoznavanie-teksta/; 'Tesseract OCR'.

[47] 'The Edinburgh Geoparser – Language Technology Group', accessed 29 June 2023, https://www.ltg.ed.ac.uk/software/geoparser/.

Digitised Theses

In 2016 the library set out to digitize 15,000 PhD theses to be made available through the Edinburgh Research Archive (ERA).[48] These are available online via ERA and can be downloaded as PDF files that have an OCR layer, making the theses searchable, however typical OCR accuracy issues are present, meaning that certain information can be missed.

See below for an example of words not being correctly identified when searching the document, from a 1986 typed thesis.[49] The light grey highlighted word on the bottom line of the image shows where the PDF recognizes the first instance of 'moral' occurring in the text, however the word circled in red at the top of the text is the same word mentioned at an earlier point but is not recognized in the PDF. The embedded OCR must have recognized a letter incorrectly, however due to the nature of searchable PDFs, we are unable to see what the generated text reads, thus users may not necessarily know when the text they are searching contains errors.

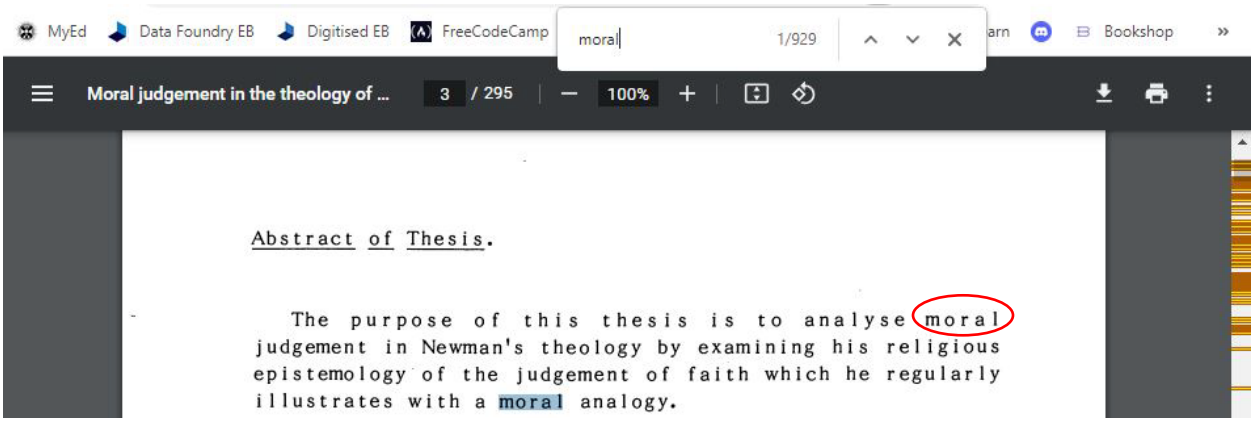

Figure 4

A new theses project aims to produce higher quality text from OCR processes to provide better usability, and higher quality text leading to better research. The proposal document, after evaluating the needs of the project and several free and paid software options, recommends the use of Tesseract 4.0 as it is free and open-source, and due to an abundance of training resources, is easy to learn. Tesseract also suited the language requirements (it can process both English and Latin), and 'Tesseract provides superior OCR on high quality documents and better results due to extensive document processing features.'[50]

Scans & Open Books

Cultural heritage digitisation is carried out by the dedicated team in the Cultural Heritage Digitisation Service for user orders, or for other projects. The process differs slightly depending on the materials, but they can largely be split into two categories.

Books and less fragile materials are scanned on the book scanners, with an inbuilt system. The images are scanned, and the pages combined into a multipage PDF file. Printed books are processed with OCR using

[48] 'ERA Home', accessed 29 June 2023, https://era.ed.ac.uk/.

[49] Gerard Magill, 'Moral Judgement in the Theology of John Henry Newman', 1986, https://era.ed.ac.uk/handle/1842/12249.

[50] 'Theses OCR Software Proposal'.

ABBY FineReader, a PDF editing software that supports text recognition. Due to the time constraints for working on orders and department projects, the OCR produced with ABBY remain unchanged from output as there is not enough resource to do any level of corrections to improve accuracy. There is therefore an acceptance of the 'raw' OCR output as an accepted text quality, when realistically it is not at the higher levels it should perhaps be. The scans are hosted on Open Books, which are freely available and many items are licensed under a Creative Commons CC BY Licence.[51]

The text generated via OCR and available embedded in the PDFs offers a degree of accessibility to the collection, in that searches can be carried out on the materials. However, due to the nature of OCR processes and material text variation, these are not entirely accurate texts and can offer a misrepresentation of the text, and if a word contains errors, it will not be returned in search results. See the OCR requirements section for examples.

The high-quality photography carried out in the department is often for more fragile materials, objects (rather than books), or where a higher level of detail is required for capture. These are captured with high quality cameras and exported as TIFF files and there is no inbuilt process or specific workflow for OCR with these. Where images are of handwritten text, this is of lower importance as programmes such as ABBY are designed for printed rather than handwritten text. However, if high quality scans were taken of printed text, there would be no text extraction workflow in place.

Lyell Notebooks

The Lyell Notebooks project focuses on handwritten notebooks by Scottish geologist, Charles Lyell. These present a different challenge to printed materials as handwriting is rather subjective to the person writing and even where the handwriting is done by one person, still liable to differences in the way they write, meaning that the usual OCR processing techniques are ineffectual.

In March 2021 the Lyell Notebooks team decided to use handwritten text recognition software Transkribus to decipher the text and aid in the transcription effort.[52] The team manually transcribed 2 of the notebooks (MSVII and Notebook No 4) to create a training model of the handwriting that could then be applied to other notebooks.[53] Speaking of the generated text in a blog post discussing the project, the team states '[t]he transcriptions are not completely accurate and need to be manually checked, so it's time consuming, but the results produce rich descriptions that will enable good searching'.[54] Using the Transkribus model, the team have continued with a degree of manual transcription, but have used Transkribus to identify difficult or unfamiliar words, especially useful for Latin terms that volunteers were unfamiliar with, such as "Fissurella graeca".[55]

[51] 'Open Books', accessed 30 June 2023, https://openbooks.is.ed.ac.uk/; 'Creative Commons — Attribution 4.0 International — CC BY 4.0', accessed 11 July 2023, https://creativecommons.org/licenses/by/4.0/.
[52] 'Transkribus'.
[53] Elise Ramsay, 'In Lyell's Own Words', *Through Lyell's Eyes* (blog), 30 April 2021, https://libraryblogs.is.ed.ac.uk/lyell/2021/04/30/in-lyells-own-words/.
[54] Elise Ramsay, 'Transcription – Through Lyell's Eyes', Through Lyell's Eyes, 19 December 2022, https://libraryblogs.is.ed.ac.uk/lyell/category/transcription/.
[55] Elise Ramsay, 'In Lyell's Own Words'.

Other publicly available models in English have been used by the team such as 'Transkribus English handwriting M3', a model trained on seventeenth-twentieth century handwritten material in English.[56] A first pass of the text is completed with Transkribus, then volunteers manually check the results and adjust where necessary. This not only functions as a way of deciphering tricky lettering to produce complete transcriptions, but also as a palaeographical training resource for the volunteers to improve their handwritten recognition skills and aid in producing better transcriptions for the project.

Summary

Several questions arise from these previous and current projects, such as who should take ownership for the OCR of projects, and whether text generated from OCR should be preserved. In the order and project scans in the Cultural Heritage Digitisation Service, the person operating the book scanner for image capture is therefore in charge of OCR for those items as it is incorporated into the inbuilt processes there and is part of the book scanner training. For the other projects, OCR improvement was a specific aspect focused on, meaning that there were often a designated person/people, in charge of examining OCR. Text outputs generated from digitised images is not solely under the remit of the digitization team, however, as the management of collections and preservation of data also falls under the digital library team.

5. L&UC New Plans

How do we embed OCR practice into future workflows?

Goobi workflows

The library currently manages its workflows with Goobi workflow, an open-source software platform developed by Intranda. It allows for the management and coordination of digitization and metadata workflows in libraries, archives, and other cultural heritage institutions. Goobi supports a customisable approach to workflows, with the option to automate some digitisation processes, and has the facilities to support an OCR module, via its graphical user interface (GUI). There are options for a trainable infrastructure, whereby models can be trained to work better with certain texts or languages. Whilst there is a plugin module to support OCR available in Goobi, the University Library team have not yet used it and it presents in Goobi as greyed-out and not accessible until it is activated with Intranda. The Goobi plugin works on a central OCR service that previously was based on ABBYY technology, but has since migrated to a mixture of Tesseract, Kalamari and a combination of other engines to carry out text optimization, segmentation and binarization to determine the recognition process and text blocks to generate results. The theory behind this combination of engines appears to be to enhance usability and text extraction. To test the accuracy, we would need to carry out comparisons on a selection of different materials to gauge how this may work on our collections.

If activating the OCR module, the OCR step would require definition to function. This entails creating a new step in Goobi, and this should denote the actions required to complete the OCR process here, such as invoking the OCR engine, passing input files, and defining the output format. OCR is often CPU intensive, putting a high demand on computing systems. As such, the OCR is not carried out in Goobi itself, and instead calls out to another application – either on the same system or located elsewhere, where the materials are queued for processing – before passing the output back to Goobi. There would be an initial set-up of the OCR module with Intranda, quoted at typically six hours.

[56] 'Transkribus English Handwriting', READ-COOP, accessed 30 June 2023, https://readcoop.eu/model/english-handwriting-18th-19th-century-2/.

The module does not grant unlimited free processing, and page quantities for processing are paid for in advance to be consumed during the OCR process. Essentially blocks of pages are bought in advance and used as required after this point. If used, this system would require careful administrative and financial planning to ensure that enough pages were available for the quantity of materials being processed to avoid workflow disruptions. There is a € 900 per year license cost for the OCR infrastructure, plus a subsequent fee of €0.06 per page for gothic or pre-1850s fonts, or €0.03 per page for others. Based on figures provided by the department, around 50,000 images are created per year by the Cultural Heritage Digitisation Service; as such if this option in Goobi (at current pricing) was employed, the approximate annual cost would be around €4000.[57]

The Goobi module can offer a confidence value to indicate quality, but as is the case with all OCR engines, there is never a 100% accuracy rate due to a myriad of reasons such as font, scan quality and language amongst the potential issues listed previously in this report. Post-extraction correction is available via the Goobi module; however it would not remove all errors. It is likely that a test would need to be carried out to determine whether post-extraction corrections were worth pursuing or not, however as there would still be a degree of errors, for the purposes of the library this may not be feasible due to staff time and budgetary constraints.

6. Recommendations

As this report has demonstrated, there is no simple answer to the OCR question in cultural heritage institutions, and unfortunately there is not a huge amount of transparency or visibility for the OCR practices of many cultural heritage organisations. Based on the research conducted for this review, I can make the following recommendations:

- In the past the university has taken a mixture of approaches to OCR, to suit the needs of different aspects of the department and individual projects. Whilst it is tempting to hope for one unified solution, the needs of the department and the breadth of materials are too varied to select one software above others (at least at present). The use of Transkribus in projects such as the Lyell Notebooks has worked well for providing a learning tool for volunteers on the project and offers opportunity for collaboration and engagement with wider audiences, however it is not a feasible option for department-wide implementation.

  Goobi offers a comprehensive OCR module that would integrate into the existing Goobi suite the University has, and its use of combined OCR engines and services, as well as its trainable structure, make it a good option for an easy to adopt option with high quality results. It is significantly more expensive than the current options with ABBYY FineReader, with a yearly license cost and per-page costs for processing versus ABBYY's smaller yearly license fee. Budget permitting, Goobi is a good option, however the costs may be prohibitive, when ABBYY still returns a reasonably high accuracy rate on extracted text, and Tesseract (which Goobi incorporates, and is identified in a new Theses digitisation proposal) are available at lower cost or free (respectively). Both of which are sustainable options with strong bases of customer support available.

[57] Figures provided by Gavin Willshaw.

- Although there are options for post-extraction text corrections, it can be difficult to determine where to draw the line and determine when the OCR is 'good enough', as perfect, 100% accuracy cannot be achieved without intensive correction. Even when conducted automatically, it still needs manual correction and checking, resulting in additional human intervention and time and labour resources. As such, the department should aim to work with OCR engines that offer as high a degree of accuracy as possible, however, some wider acknowledgement of text quality for users may be beneficial – see below.

- Transparency in the OCR process and generated text outputs has been a theme across projects such as the eMOP and in the discussions raised around scholars and students using text that they are unaware of the quality for (e.g. in text overlaid PDF files).[58] This raises questions of awareness of processes and text quality; should the onus be on teaching in academia raising awareness of these issues, or on the heritage institution as the creator of these texts? Both, would be the ideal answer, however digital research skills with an awareness of text quality are not often built into academic programmes and cultural heritage institutions often do not alert users to text quality. The National Library of Scotland's indication of text quality for their datasets is not something that is seen often but is good practice and is particularly useful and indicative of the condition of the materials for anyone considering their use. Where digital texts created via OCR are available in library systems, it would be valuable to have a note for users to see indicating that there may be errors – this would be a time-consuming and lengthy process to carry out for individual texts and backdating the catalogue would not be feasible. An alternative option such as a web page outlining some basic information about the text creation process and challenges in using these texts may be beneficial, which the service could draw attention to.

- Digital preservation of the OCRed text also poses a problem, with debate over how worth it is preserving due to the quality. At present, there is never going to be 100% accurate OCR of these texts (unless transcriptions are corrected via crowdsourcing with manual intervention). As such, not only do these current OCR transcriptions provide access to our resources, but they are worth preserving as key digital outputs. If decisions are made in future to improve the text quality, these would need to be updated in digital preservation. As this is not likely to be soon, or anticipated in great volume, a plan should be made for this implementation if the need arises in future.

- After reviewing the literature and identifying individual projects and very few specific case studies, there is a need across cultural heritage at least on a national level to survey OCR practices and applications to evaluate if institutional and user needs are being met. This could potentially be a future project for the University of Edinburgh (resources permitting) that would add value for the department and would likely be well-placed for publication and circulation within cultural heritage institutions.

[58] The Initiative for Digital Humanities, Media, and Culture, Texas A&M University, 'Early Modern OCR Project'; Lara Putnam, 'The Transnational and the Text-Searchable: Digitized Sources and the Shadows They Cast', *The American Historical Review* 121, no. 2 (1 April 2016): 377–402, https://doi.org/10.1093/ahr/121.2.377.

7. Bibliography

‘(9) Title Page - Encyclopaedia Britannica; or, A Dictionary of Arts and Sciences, Compiled upon a New Plan … > Volume 1, A-B - Encyclopaedia Britannica - National Library of Scotland’. Accessed 29 June 2023. https://digital.nls.uk/encyclopaedia-britannica/archive/188082824?mode=transcription.

‘18thConnect - TypeWright’. Accessed 11 July 2023. https://18thconnect.org/typewright/documents.

‘A Medical History of British India – Data Foundry’. Accessed 2 July 2023. https://data.nls.uk/data/digitised-collections/a-medical-history-of-british-india/.

Bennet, Mike. ‘Automated Item Data Extraction from Old Documents’. *University of Edinburgh Library Labs Blog* (blog), 23 June 2017. https://libraryblogs.is.ed.ac.uk/librarylabs/2017/06/23/automated-item-data-extraction-from-old-manuscripts/.

Bennett, Mike. ‘Sp-Experiments’. Python, 8 June 2017. https://github.com/mbennett-uoe/sp-experiments/blob/94ce9184cd1c0c3a5bdb720ba77f3ceaa6d76317/sp_crop.py.

Centre for Data, Culture & Society, the University of Edinburgh. ‘Text Extraction & Preparation’. Managing Digitised Documents Pathway, n.d. https://www.cdcs.ed.ac.uk/training/training-pathways/managing-digitised-documents-pathway/text-extraction-preparation.

Christy, Matthew, Anshul Gupta, Elizabeth Grumbach, Laura Mandell, Richard Furuta, and Ricardo Gutierrez-Osuna. ‘Mass Digitization of Early Modern Texts with Optical Character Recognition’. *Journal on Computing and Cultural Heritage* 11, no. 1 (7 December 2017): 6:1-6:25. https://doi.org/10.1145/3075645.

Coordinator, Sidney Gao, Digital Imaging. ‘OCR: Who Does It Best?’ University of Cincinnati Libraries Digital Collections Documentation, 7 August 2020. https://uclibs.github.io/digitization-workflow/2020/08/07/ocr-comparison.html.

‘Creative Commons — Attribution 4.0 International — CC BY 4.0’. Accessed 11 July 2023. https://creativecommons.org/licenses/by/4.0/.

Elise Ramsay. ‘In Lyell’s Own Words’. *Through Lyell’s Eyes* (blog), 30 April 2021. https://libraryblogs.is.ed.ac.uk/lyell/2021/04/30/in-lyells-own-words/.

———. ‘Transcription – Through Lyell’s Eyes’. Through Lyell’s Eyes, 19 December 2022. https://libraryblogs.is.ed.ac.uk/lyell/category/transcription/.

eMOP. ‘Outcomes’. Accessed 9 July 2023. https://emop.tamu.edu/.

‘EMOP Workflow: Lucidchart’. Accessed 9 July 2023. https://lucid.app/lucidchart/7da90ad2-ab8b-4cb7-ba01-9ddaad1f5385/edit.

‘ERA Home’. Accessed 29 June 2023. https://era.ed.ac.uk/.

Europeana PRO. ‘Issue 13: OCR’. Accessed 28 June 2023. https://pro.europeana.eu/page/issue-13-ocr.

FineReader PDF. ‘ABBYY FineReader PDF Software: Open, Read & Edit PDFs’. Accessed 11 July 2023. https://pdf.abbyy.com/.

‘Geotagging — The Edinburgh Geoparser 1.3 Documentation’. Accessed 9 July 2023. https://groups.inf.ed.ac.uk/geoparser/documentation/v1.3/html/geotag.html#the-tokenise-component.

Heliński, Marcin, Miłosz Kmieciak, and Tomasz Parkoła. ‘Report on the Comparison of Tesseract and ABBYY FineReader OCR Engines’, n.d.

Magill, Gerard. ‘Moral Judgement in the Theology of John Henry Newman’, 1986. https://era.ed.ac.uk/handle/1842/12249.

ocropus.github.io. ‘OCRopus OCR Engine(s)’. Accessed 29 June 2023. https://ocropus.github.io/.

O'Hara, Laura Turner. 'Cleaning OCR'd Text with Regular Expressions'. *Programming Historian*, 22 May 2013. https://programminghistorian.org/en/lessons/cleaning-ocrd-text-with-regular-expressions.

Olson, Leanne, and Veronica Berry. 'Digitization Decisions: Comparing OCR Software for Librarian and Archivist Use'. *The Code4Lib Journal*, no. 52 (22 September 2021). https://journal.code4lib.org/articles/16132.

'Open Books'. Accessed 30 June 2023. https://openbooks.is.ed.ac.uk/.

OpenCV. 'OpenCV.Org'. Accessed 29 June 2023. https://opencv.org/.

'OverProof - About and Help'. Accessed 9 July 2023. https://overproof.projectcomputing.com/about#pricing.

'OverProof - Home'. Accessed 9 July 2023. https://overproof.projectcomputing.com/.

Priyanka. 'Common Regular Expressions for Text Cleaning in Python'. *Medium* (blog), 18 November 2022. https://medium.com/@priyankads/common-regular-expressions-for-text-cleaning-in-python-5a13b832d340.

Putnam, Lara. 'The Transnational and the Text-Searchable: Digitized Sources and the Shadows They Cast'. *The American Historical Review* 121, no. 2 (1 April 2016): 377–402. https://doi.org/10.1093/ahr/121.2.377.

Python.org. 'Python.Org', 22 June 2023. https://www.python.org/.

READ-COOP. 'Dutch Handwriting 17th-19th Century'. Accessed 28 June 2023. https://readcoop.eu/model/dutch-handwriting-17th-19th-century/.

READ-COOP. 'How the State Archives of Zurich Published 50,000 Pages with Read&search'. Accessed 28 June 2023. https://readcoop.eu/success-stories/state-archives-of-zurich/.

READ-COOP. 'Transcribing 3 Million Scans at the National Archives of the Netherlands'. Accessed 28 June 2023. https://readcoop.eu/success-stories/national-archives-of-the-netherlands/.

READ-COOP. 'Transkribus'. Accessed 16 January 2022. https://readcoop.eu/transkribus/.

READ-COOP. 'Transkribus English Handwriting'. Accessed 30 June 2023. https://readcoop.eu/model/english-handwriting-18th-19th-century-2/.

'Read&Search: Kanton Zurich Staatsarchiv'. Accessed 11 July 2023. https://ratsmanuale-zuerich.transkribus.eu/search?t.

Schantz, Herbert F. *The History of OCR, Optical Character Recognition*. [Manchester Center, Vt.] : Recognition Technologies Users Association, 1982. http://archive.org/details/historyofocropti0000scha.

tessdoc. 'Tesseract User Manual'. Accessed 7 July 2023. https://tesseract-ocr.github.io/tessdoc/.

'Tesseract OCR'. C++. 2014. Reprint, tesseract-ocr, 24 April 2023. https://github.com/tesseract-ocr/tesseract.

'The Edinburgh Geoparser – Language Technology Group'. Accessed 29 June 2023. https://www.ltg.ed.ac.uk/software/geoparser/.

The Initiative for Digital Humanities, Media, and Culture, Texas A&M University. 'Early Modern OCR Project'. Accessed 28 June 2023. https://emop.tamu.edu/about.

'The Student, Issues 1-12'. Accessed 29 June 2023. https://openbooks.is.ed.ac.uk//record/121631?highlight=*:*.

'Theses OCR Software Proposal', n.d. https://uoe.sharepoint.com/:w:/r/sites/Digitisation/_layouts/15/Doc.aspx?sourcedoc=%7B799F568D-D686-42F4-B4DC-B80AD1615830%7D&file=Theses%20OCR%20Software%20Proposal.docx&action=default&mobileredirect=true.

Underwood, Ted. 'DataMunging'. Python, 8 April 2023. https://github.com/tedunderwood/DataMunging.

‘VARD - About’. Accessed 9 July 2023. https://ucrel.lancs.ac.uk/vard/about/.
‘Zürcher Ratsmanuale 1700-1798’. Accessed 28 June 2023. https://ratsmanuale-zuerich.transkribus.eu/.
Черемных, Алексей. ‘Cognitive OpenOCR - Распознавание Текста’. Свободные программы для Windows - КонтинентСвободы.рф. Accessed 29 June 2023. https://континентсвободы.рф/cognitive-openocr-raspoznavanie-teksta/.