LIBER QUARTERLY
The Journal of the Association of European Research Libraries
Vol. 33, (2023) 1–20 | e-ISSN: 2213-056X

Nautilus – An End-To-End METS/ALTO OCR Enhancement Pipeline

Pit Schneider

Department of IT and Digital Innovation, National Library of Luxembourg, Luxembourg, pit.schneider@bnl.etat.lu, orcid.org/0000-0001-9034-1551

Yves Maurer

Department of IT and Digital Innovation, National Library of Luxembourg, Luxembourg, yves.maurer@bnl.etat.lu, orcid.org/0000-0002-1862-1943

Ralph Marschall

Department of IT and Digital Innovation, National Library of Luxembourg, Luxembourg, ralph.marschall@bnl.etat.lu, orcid.org/0000-0001-8255-1786

Abstract

When a digital collection has been processed by OCR, the usability expectations of patrons and researchers are high. While the former expect full text search to return all instances of terms in historical collections correctly, the latter are more familiar with the impacts of OCR errors but would still like to apply big data analysis or machine-learning methods. All of these use cases depend on high quality textual transcriptions of the scans. This is why the National Library of Luxembourg (BnL) has developed a pipeline to improve OCR for existing digitised documents. Enhancing OCR in a digital library not only demands improved machine learning models, but also requires a coherent reprocessing strategy in order to apply them efficiently in production systems.

Uopen Journals | http://liberquarterly.eu/ | DOI: 10.53377/lq.13330

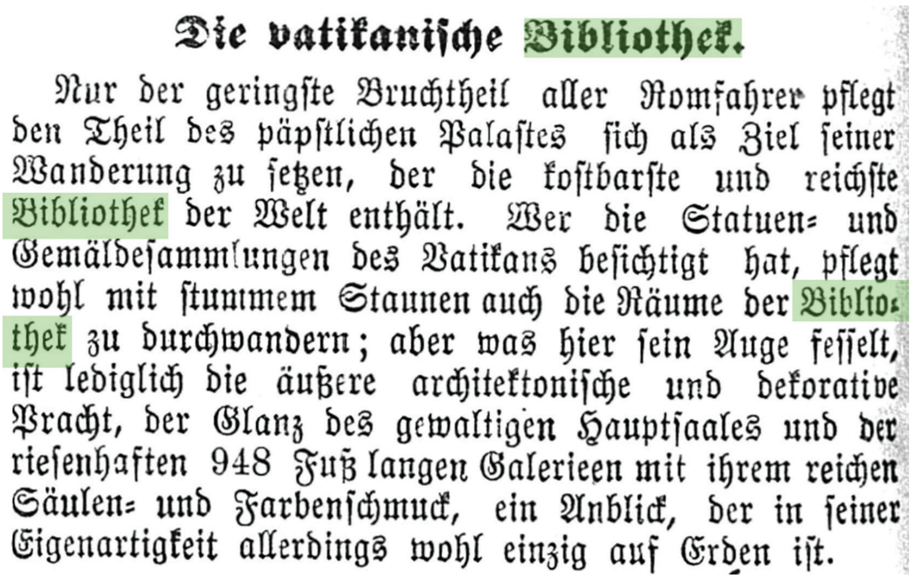
The newly developed software tool, Nautilus, fulfils these requirements using METS/ALTO as a pivot format. The BnL has open-sourced it so that other libraries can re-use it on their own collections. This paper covers the creation of the ground truth, the details of the reprocessing pipeline, its production use on the entirety of the BnL collection, along with the estimated results. Based on a quality prediction measure, developed during the project, approximately 28 million additional text lines now exceed the quality threshold.

Keywords: OCR quality; OCR correction; METS/ALTO; Luxembourg historical newspapers; ground truth

1. Introduction

The National Library of Luxembourg (BnL) has been digitising its newspaper heritage collections since 2005 in the METS/ALTO format with OCR processing and manual zoning of individual articles. The results are available on the eluxemburgensia[1] platform, which provides full-text search, displays the transcribed text alongside high-resolution images and can highlight search results on the image (Figure 1) using coordinates from the ALTO XML files.

Fig. 1: Example word-highlights, with one highlight spanning across two text lines.



Die vatikaniſche Bibliothek.

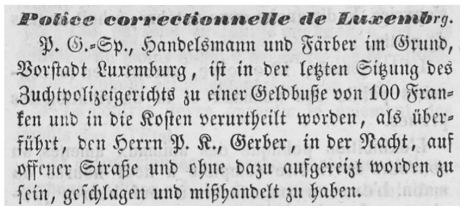
Nur der geringſte Bruchtheil aller Romfahrer pflegt
den Theil des päpſtlichen Palaſtes ſich als Ziel ſeiner
Wanderung zu ſetzen, der die koſtbarſte und reichſte
Bibliothek der Welt enthält. Wer die Statuen= und
Gemäldeſammlungen des Vatikans beſichtigt hat, pflegt
wohl mit ſtummem Staunen auch die Räume der Biblio=
thek zu durchwandern; aber was hier ſein Auge feſſelt,
iſt lediglich die äußere architektoniſche und dekorative
Pracht, der Glanz des gewaltigen Hauptſaales und der
rieſenhaften 948 Fuß langen Galerieen mit ihrem reichen
Säulen= und Farbenſchmuck, ein Anblick, der in ſeiner
Eigenartigkeit allerdings wohl einzig auf Erden iſt.

The BnL has never had in-house scanning facilities so external suppliers were hired through public tenders to perform the scanning, computer-assisted zoning into articles, setting article types (e.g. ARTICLE, DEATH_NOTICE or ADVERTISEMENT) and producing METS/ALTO. While manual parts of the digitisation chain, like handling, scanning and zoning have been thoroughly checked by the quality assurance team from the BnL, this was not done for the raw OCR transcription of the articles. It follows that the quality of the textual representation varies hugely and depends on the different OCR engines used at the time. The idea was that OCR could be improved in an automated way while keeping the benefit of the manual work.

The newspaper corpus that has been digitised is multilingual (German, French, Luxembourgish and some other languages), uses a variety of typefaces (Antiqua, Fraktur), and mixes all of these elements even on the same page. As shown on Figure 2, it was challenging for the suppliers to always use the correct OCR engine for each block of text.

Fig. 2: Example block where OCR (output at left) was wrongly performed using ABBYY FineReader Engine 10 for French (Antiqua) (source image at right).

Police correctionnelle de Luxembrg.
$). ©,*©p., ^anbelèinmm unb gctïber tnt ©vttttb, Sovftabt Suiemburg , tft tn ber te|ten ©t'Çuitg beô Budjtpcltjct'gcricljté ju etncrQefbbufïe non 100 gran* îen mib in bte tîoftcn ocvurt^cilt «jorben, afé liber* füf;vt, ben Servit ÿ. t?., ©erber, tn ber 9îad;t, aitf offenev ©trafic unb cljne ba$u aufgeretjt toorben ju fetn, gcfélagett unb mtfifmnbcft 311 ^abeit.



Police correctionnelle de Luxembrg.
P. G.-Sp., Handelsmann und Färber im Grund, Vorſtadt Luxemburg, iſt in der letzten Sitzung des Zuchtpolizeigerichts zu einer Geldbuße von 100 Franken und in die Koſten verurtheilt worden, als überführt, den Herrn P. K., Gerber, in der Nacht, auf offener Straße und ohne dazu aufgereizt worden zu ſein, geſchlagen und mißhandelt zu haben.

The idea of improvement was validated in a pilot project using the Tesseract (Smith, 2007) software, during which a metric based on dictionaries was used to make sure that new OCR was not worse than what we got from the supplier. However, it was noted that "*Tesseract did not perform consistently better than the original OCR, as we had expected. In fact it performed slightly worse in an overall comparison*" (Maurer, 2017), for both Antiqua and Fraktur fonts, so a new effort was started in 2020 where all the individual components were tested again and improved as necessary. This led us to the development of *Nautilus*, an end-to-end METS/ALTO OCR enhancement pipeline applied and released as open source[2] by the BnL.

The rest of this article is structured as follows: Section 2 presents the creation of a ground truth dataset. The main software pipeline is then laid out in Section 3, followed by a segment dedicated to the results related to its first application. Finally, Section 5 concludes the paper.

2. Ground Truth Generation

Nowadays machine learning models are at the core of OCR and their accuracy has a direct influence on the recognition quality for characters and words. As suggested by other, similar projects, like Kettunen et al. (2020), training on BnL data would produce the best results with Nautilus, for any underlying OCR engine that we would end up choosing. For that reason, the first project step involved deriving *ground truth* from the BnL's digitised documents. To be able to generate such a dataset, we were faced with decisions regarding the data sampling strategy, transcription guidelines and quality assurance.

2.1. Data Sampling

During data sampling, we followed the objective of striking a balance between a diverse and a representative ground truth set. We tried to reflect the diversity of the corpus by balancing the following properties:

- Language (German, French, Luxembourgish)
- Newspaper title (52 distinct titles)
- Publication date (1841–1954)

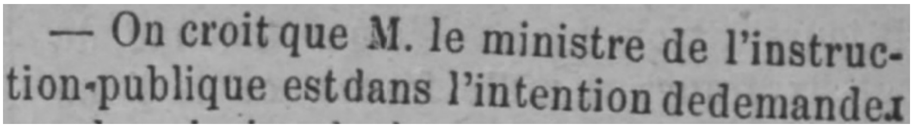
The final selection includes 6723 text blocks, ranging from 1 to 1054 in number of transcribed words. The public domain part of the ground truth set has been published on the BnL's open data platform.[3]

2.2. Transcription Guidelines

With financing made available by *The AI4Gov initiative* of the government of Luxembourg (The Luxembourg Government, n.d.), an external supplier was tasked with the transcription of these blocks. To guarantee its smooth execution, we needed a set of precise technical specifications. Those were fundamentally inspired by the OCR-D (Neudecker et al., 2019) project, but we ended up relaxing many requirements so that it was easier to work with but which also means that it is not 100% compatible with ground truth generated according to the real OCR-D guidelines.

During the execution of the project, we collaborated in a responsive manner with the supplier, to address unexpected cases and questions. To name a few examples, the transcription for upside down (Figure 3), horizontally reversed, invisible or artwork characters needed to be defined, once the first cases were discovered.

Fig. 3: Unexpected upside down "r" (2nd line, last character).



— On croit que M. le ministre de l'instruc-
tion-publique estdans l'intention dedemandeɹ

In general, we defined the transcription process itself to follow three steps for every given block:

1. Identifying the text lines within the block.
2. Identifying the characters within each text line.
3. Take the existing ALTO file and update lines, words and coordinates (works for ALTO 1.0 and above).

Two decisions, tied to the first two steps, were taken to potentially allow the training of more robust character recognition models:

- Ground truth text line bounding boxes were not allowed to be skewed (i.e. forced to be perfectly horizontal with respect to the block), entailing that the text line itself had the possibility to be slightly skewed.
- Every single character transcription was assigned a confidence value, based on three discrete classes. This enabled including/excluding characters that were harder to recognise for the transcribers (e.g. because of smudged print) into the model training process.

2.3. Quality Assurance

To validate the level of ground truth quality, the requirement for a global accuracy of 99.95% was set. To enforce this target, we checked a small subset of blocks using a combination of custom software and manual verification. The software was designed to perform as many automatic checks as possible, such as the filtering of non-whitelisted characters or the detection of overlapping line bounding boxes (potentially indicating an error).

Returned data batches generally featured a couple of smaller quality issues, which were promptly rectified with subsequent batch iterations. Among those problems, to name a few, were invalid ALTO files, wrong word bounding boxes or a wrong number of confidence values. In the end, after five months of preparing, specifying and transcribing the data, the ground truth project concluded and the desired accuracy target was met.

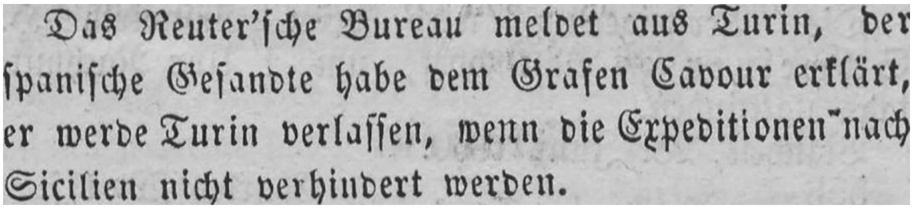
3. Nautilus

We next steer the focus to the Nautilus pipeline itself. The software encapsulates an end-to-end workflow with regard to the objective of enhancing OCR quality of existing METS/ALTO data.

3.1. Block Definition

To proceed with the specifics of the pipeline, we first require a more precise definition of a *block*. A block generally represents an image containing text of an individual paragraph or even a small article (ALTO *TextBlock* tag). In terms of layout, a block is always contained within a single text column. We refer to a block image (e.g. Figure 4), identified through index i, as B_i. Similarly, a new Nautilus output (derived from B_i) is denoted as B_i^{new}, the *original* OCR text as B_i^{ori} and the *ground truth* counterpart as B_i^{gt}.

Fig. 4: Small sample block used to demonstrate the pipeline, in the following referred to as B_s.



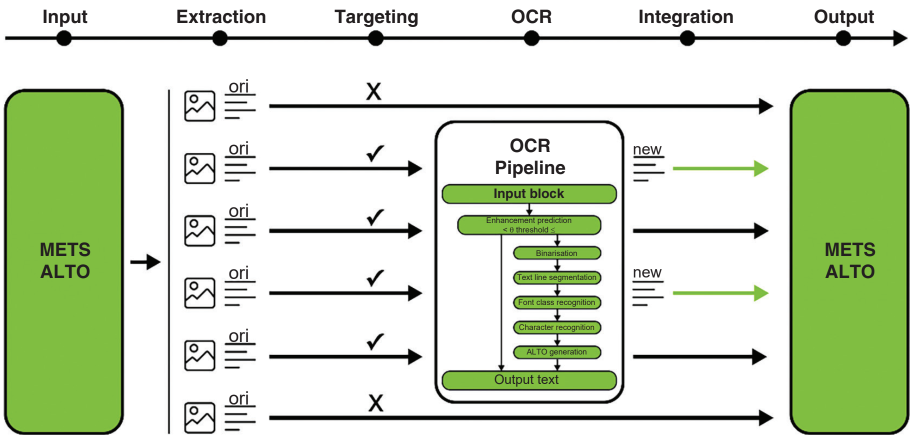
Das Reuter'ſche Bureau meldet aus Turin, der
ſpaniſche Geſandte habe dem Grafen Cavour erklärt,
er werde Turin verlaſſen, wenn die Expeditionen nach
Sicilien nicht verhindert werden.

The targeting of existing blocks and not changing their coordinates in the output file means that the METS logical *structMap* does not need to be changed. This simplifies the integration of the resulting improvements into the METS/ALTO package.

3.2. Mets/ALTO Pipeline

The six steps seen in Figure 5 can be detailed as follows:

Fig. 5: METS/ALTO pipeline overview.



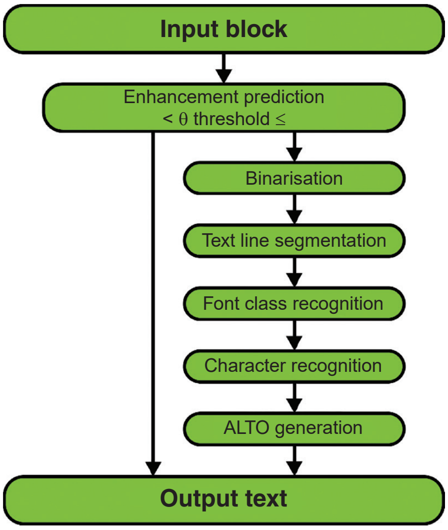
1. **Input**
   A METS/ALTO package is expected as input. This means that, next to the METS file, every scanned page of the target document has to be represented as one single ALTO (original OCR) and one archive image file.
2. **Extraction**
   Through parsing of the METS/ALTO files, every B_i^{ori} is extracted and paired with the corresponding B_i, which is obtained by cropping the respective page image.
3. **Targeting**
   Not every pair represents an enhancement target, however. The pipeline allows control over the type of blocks that are subject to reprocessing (e.g. blocks of type *DEATH_NOTICE* and *ADVERTISEMENT* were discarded because their layout worked poorly with Nautilus).

4. **OCR**
   The OCR step can be seen as a meta-pipeline, to which every target pair is fed. This is the basis for a potential conversion to B_i^{new}. Alternatively, the unmodified B_i^{ori} is retained. Further elaborations on the OCR pipeline will follow with the next subsections.
5. **Integration**
   Next, every B_i^{new} remains to be integrated into an updated METS/ALTO package. This concretely demands the update of every targeted *TextBlock* subtree within the respective ALTO file. Finally, the METS file is modified to incorporate the new ALTO checksums, file sizes and creation dates.
6. **Output**
   To complete the circle, Nautilus outputs the input METS/ALTO package, but with hopefully improved OCR quality.

3.3. OCR Pipeline

Diving in a bit deeper, located between the input block and the output text, are six distinct software components (Figure 6), which together form the OCR part of the pipeline.

Fig. 6: Sequentially structured OCR pipeline.



3.3.1. Enhancement Prediction

Using a regression model trained on block features, denoted as *Enhance*, this component aims to predict the amount of improvement in OCR quality when retaining B_i^{new} instead of B_i^{ori}.

The regression model operates on a quality measure *q*, which is defined for B_i^{ocr} as

$$q(B_i^{ocr}) = 1 - \min(|B_i^{ocr}|, \mathrm{edit}(B_i^{ocr}, B_i^{gt}) / |B_i^{ocr}|$$

with

- the cardinality operator returning the amount of characters in the block (including whitespaces).
- *edit* being the popular Levenshtein (1965) distance.

It follows that the difference $q(B_i^{new})$-$q(B_i^{ori})$, predicted by Enhance, ranges from –1 to 1.

Any threshold value (denoted as θ) can subsequently be applied to fork the pipeline and eventually terminate right away, should Enhance (B_i^{ori}) be lower than desired. In that case, based on conducted experiments, the processing time of the pipeline generally stays below 5% of the time needed for a regular run (using all six components). This reduction in processing cost is joined by the two other motivations for enhancement prediction, namely:

- Reduction of the risk to degrade the quality of some blocks.
- Access to estimated improvement statistics.

For more detailed explanations, a dedicated article (Schneider & Maurer, 2022) has been compiled, discussing all our enhancement prediction findings.

Leveraging B_s^{ori} from Section 3.1, containing a couple of unrecognised characters (e.g. *Grasen* instead of *Grafen*), a considerable enhancement prediction comes with Enhance (B_s^{ori}) = 0.067.

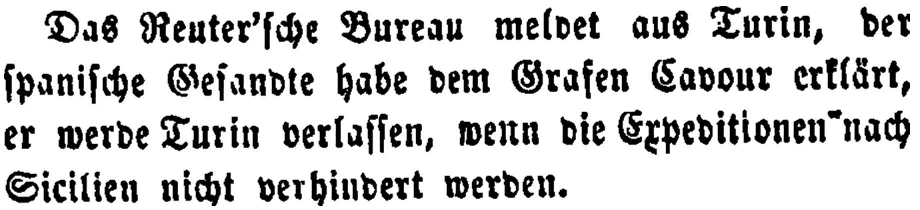
3.3.2. Binarisation

Next follows binarisation, having four major tasks, being:

- Transforming B_i into a binary image.
- Dilating the image using a 2x2 pixels structuring element to potentially repair broken characters.
- Padding the image so that the block has a leading and trailing white margin.
- Possibly inverting an image containing light text on a dark background.

The final implementation of the binarisation component is largely based on *OpenCV* (Bradski, 2000) functions. The result of the application on B_s can be seen with Figure 7.

Fig. 7: Binarised version of Bs.

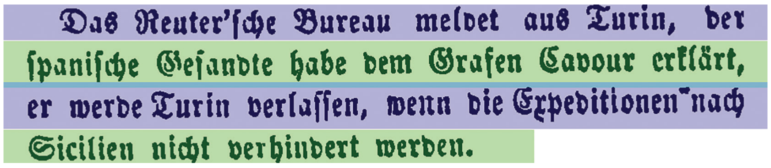


Das Reuter'ſche Bureau meldet aus Turin, der ſpaniſche Geſandte habe dem Grafen Cavour erklärt, er werde Turin verlaſſen, wenn die Expeditionen nach Sicilien nicht verhindert werden.

3.3.3. Text Line Segmentation

Segmenting the binary image into individual text lines is a required step for the subsequent font class and character recognition components. Our efforts regarding this step have been comprehensively documented by Schneider (2021), describing the development of our own segmentation algorithm. Named *CombiSeg*, the method leverages a combination of morphological image operations and horizontal histogram projections to return a set of text line bounding boxes (Figure 8). OpenCV has once again been used to support this implementation.

Fig. 8: Visualisation of the text line bounding boxes for B_s.



Das Reuter'ſche Bureau meldet aus Turin, der
ſpaniſche Geſandte habe dem Grafen Cavour erklärt,
er werde Turin verlaſſen, wenn die Expeditionen nach
Sicilien nicht verhindert werden.

The decision to develop and adopt a new solution was driven by two factors:

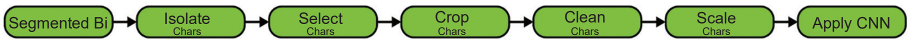
- Being a fast algorithm (Schneider, 2021), CombiSeg is a major contributor to the overall efficiency of Nautilus. Building a fast pipeline, that can be applied on a large volume of data in a relatively short time frame, was one of the main objectives.
- CombiSeg utilises some parameters that we were able to tune based on our own data. This level of adaptation is usually not given with an out-of-the-box solution.

3.3.4. Font Class Recognition

Based on the diverse nature of our data and some conducted tests, targeting individual font classes seemed to be a key ingredient for improvements. That is why the pipeline forks on two font classes before arriving at the main character recognition component. The basis for this is a convolutional neural network classifying Antiqua and Fraktur fonts after having been trained on a set of individual character images for each class.

Depicted in Figure 9, a small preprocessing pipeline is responsible for the conversion of a segmented binary image into a set of individual characters (chars).

Fig. 9: Font class recognition preprocessing pipeline.



1. **Isolate Characters**
   Using the text line bounding box information obtained from the previous component, we isolate individual characters within every line. This is done by deriving a list of connected components from binary B_i.
2. **Select Characters**
   Selecting a subset of characters follows a strategy which prioritised first characters within words, having a higher chance of being capital

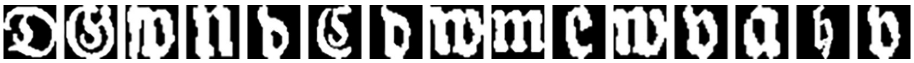
letters (bigger visual difference among font classes). Second, first characters within the entire line are discarded (except for the first line) since they are more likely to represent digits (e.g. enumerations).

3. **Crop Characters**
   Next, every character is cropped from binary B_i and stored as an individual image.
4. **Clean Characters**
   Possible adjacent characters, identified through the connected components map, are removed from every character image, in an attempt to strengthen the isolation process.
5. **Scale Characters**
   Finally, scaling is applied so that every character image is of expected dimension 32×32 and can be processed by the neural network.

The convolutional neural network itself consists of two *convolutional* layers, each one followed by *max pooling*, joined by two *fully connected* layers, separated by a *dropout* layer in between.

Majority voting, through the run of a maximum of 15 preprocessed character images (Figure 10), ultimately decides on the font class.

Fig. 10: All 15 character images extracted from B_s, with every single one being classified as Fraktur.



3.3.5. Character Recognition

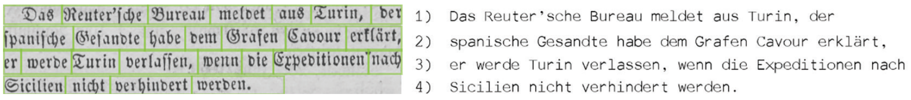
The most crucial component, character recognition, is implemented using the K*raken* software (Kiessling, 2019), which has been forked from the *OCRopus OCR System* (Breuel, 2008). Next to being rather easy to use, we integrated the software library due to its ability to return geometric information about the location of characters within B_i. This marks an obvious requirement for outputting the ALTO format.

Character recognition using Kraken is done by providing:

1. Binary B_i.
2. The text line segmentation information.
3. The font class (determining the Kraken model that is being applied).

The Kraken models are trained using the default network architecture and early stopping. Figure 11 shows the recognition output for B_s.

Fig. 11: B_s and B_s^{new} visualised side-by-side.



Das Reuter'ſche Bureau meldet aus Turin, der
ſpaniſche Geſandte habe dem Grafen Cavour erklärt,
er werde Turin verlaſſen, wenn die Expeditionen nach
Sicilien nicht verhindert werden.

```
1)  Das Reuter'sche Bureau meldet aus Turin, der
2)  spanische Gesandte habe dem Grafen Cavour erklärt,
3)  er werde Turin verlassen, wenn die Expeditionen nach
4)  Sicilien nicht verhindert werden.
```

3.3.6. ALTO Generation

Finally, ALTO generation is implemented through the addition of some logic to improve the character positions returned by Kraken, in order to obtain a final set of word bounding boxes (Figure 12). Every character recognition confidence score *conf*, falling in the range of 0 to 1, is remapped to *CONF*, such that

$$\text{CONF} = \text{round}(9 - \text{conf}*9).$$

Fig. 12: ALTO snippet of first TextLine of B_s^{new}.

```
<?xml version="1.0" ?>
<TextBlock>
    <TextLine HPOS="0" VPOS="1" WIDTH="767" HEIGHT="35">
        <String CONTENT="Das" CC="000" HPOS="0" VPOS="1" WIDTH="112" HEIGHT="35"/>
        <SP/>
        <String CONTENT="Reuter'sche" CC="01000000001" HPOS="112" VPOS="1" WIDTH="166" HEIGHT="35"/>
        <SP/>
        <String CONTENT="Bureau" CC="000000" HPOS="278" VPOS="1" WIDTH="120" HEIGHT="35"/>
        <SP/>
        <String CONTENT="meldet" CC="020000" HPOS="398" VPOS="1" WIDTH="119" HEIGHT="35"/>
        <SP/>
        <String CONTENT="aus" CC="000" HPOS="517" VPOS="1" WIDTH="76" HEIGHT="35"/>
        <SP/>
        <String CONTENT="Turin," CC="000000" HPOS="593" VPOS="1" WIDTH="116" HEIGHT="35"/>
        <SP/>
        <String CONTENT="der" CC="000" HPOS="709" VPOS="1" WIDTH="58" HEIGHT="35"/>
```

4. Results

The necessary testing results, required to bring the newly developed enhancement tool to production, were attained in early 2021. At the time, the enhancement prediction component was still based on a binary method, classifying the quality of B_i^{ori} (*Quality* in Schneider and Maurer (2022)).

4.1. Testing

Applied on a split-off ground truth test set, the following observations were made, using threshold $\theta=0.95$ (quality measure defined in chapter 3.3.1 for $q(B_i^{ocr})$):

- 67% of the blocks were predicted to already exceed θ.
- Among the remaining 33% (reprocessing targets), 91% were improved in a way that a subsequent quality prediction on B_i^{new} exceeded θ. By leveraging B_i^{gt}, the average enhancement per block could be reviewed and was found to be 0.097 (in contrast to 0.04 for all blocks).

Another reassurance was the font classification accuracy of 99%, which proved to be valuable to especially enhance Fraktur based blocks.

4.2. Batch Processing

We subsequently applied Nautilus on a little more than 100 thousand newspapers, containing just above 475 thousand pages (10 TB of TIF images), 8 million blocks and 175 million text lines. Processing time was reduced by splitting the corpus in 8 equally sized batches, which were fed to 8 instances of the software running in parallel. Based on the reprocessing rate of 33% while testing, we decided to retain the same $\theta=0.95$ threshold for this first application.

All the computations, which took nearly 15 days, were done on the same Linux-based virtual machine that was used for the training of the models. Enabled by the *Luxembourg Government* (n.d.), we were able to use a machine with 8-CPU cores, a *V100D-16C NVIDIA* GPU and 48 GB of RAM. The entire dataset was stored and loaded from a Network File System.

Once the processing concluded, the data was re-ingested into the eluxemburgensia system, re-indexed into the *Solr* (https://solr.apache.org/) search engine and made available for our patrons.

4.3. Evaluation

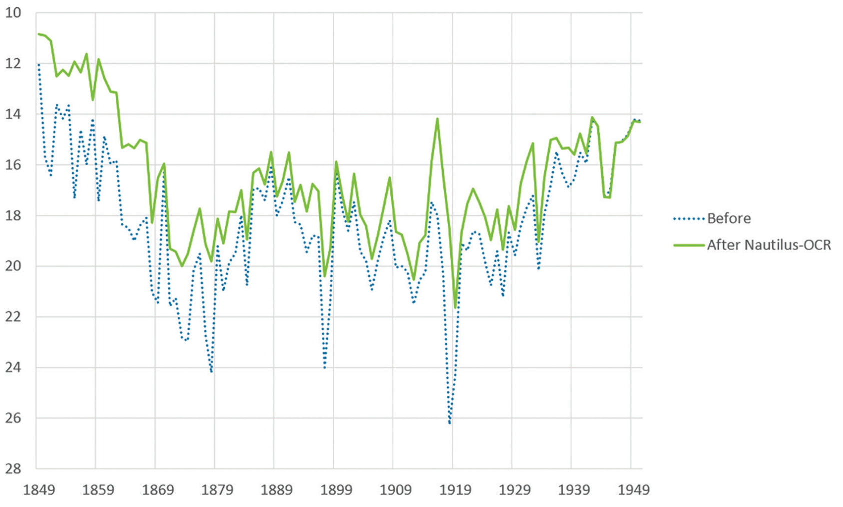
Although the absence of any ground truth counterparts rendered a precise assessment unfeasible this time, we were able to strongly tie the evaluation of the first Nautilus run to Quality, in addition to a vocabulary growth (*VG*) method proposed by Maurer (2017).

To come up with a more representative picture, the final metrics were remodelled to reflect the number of text lines (blocks are varying heavily in size). Considering only the 23% of text lines (containing 85% Fraktur), that were ultimately reprocessed (due to prediction $q(B_i^{ori}) \leq \theta$, Quality now assumes that:

- The fraction of lines that is exceeding the threshold equals 70% (Fraktur lines only: 75%).
- An additional 15% of words can be found in a dictionary of the same language (Fraktur lines only: 18%).

Those numbers translate to an estimated additional 28 million lines (16% of the total 175 million) that are predicted to feature the sufficient quality standard.

Fig. 13: Comparison of the Vocabulary Growth measure for the newspaper title "Luxemburger Wort" (1849–1950) before and after applying Nautilus.



The VG method proposed by Van de Camp (2008) and adapted by Maurer (2017) counts the number of unique words per million words. OCR errors introduce a lot of variations for a single correct word, so the VG measure is low when the OCR quality is high. That is why the vertical axis in Figure 13 is upside down.

As can be seen, the new OCR is consistently better or at least as good, according to this measure, but the improvements are uneven over the years and tend to be better for the older period. This can be explained by the fact that Nautilus carefully targets Fraktur and Antiqua in a way that the original supplier did not, and the period after the 2nd of March 1942 contains no more Fraktur, as explained in Luxemburger Wort (1942).

4.4. Comparison with Tesseract

Since our first tests were with Tesseract we also wanted to evaluate the performance of our pipeline against it, using version 5.0.0-alpha-815-g5761. For this comparison, we used a tool by Carrasco (2014) on the ground truth and Tesseract, respectively Nautilus, since it produces a measure that is also used in other papers, such as Hegghammer (2022). We found:

- 3.68 character error rate and 11.82 word error rate for Nautilus.
- 5.05 character error rate and 16.57 word error rate for Tesseract.

Tesseract was used with the *Fraktur*, *deu*, *ltz* and *fra* models as appropriate and the resulting text was post-processed to conform to the same guidelines as used in the ground truth (e.g. no long “s”, normalised dashes etc.)

5. Conclusion

With the development of an open-source software pipeline and the generation of ground truth data, the objectives of the BnL to improve OCR have been met. In addition to promising results from the first run of Nautilus, this approach could be the basis for future iterations. Individual components of the pipeline will have to be improved, extended, or new components added.

Image processing is currently applied in a very limited manner in the binarisation component. However, dewarping, rotating and cleaning are interesting treatments that low quality source images could certainly benefit from. Moreover, it is imaginable to relax the current text block level requirement in the future by introducing layout analysis as a precursor to Nautilus.

In terms of component replacement, tests could be done on the generated ground truth with OCR engines other than Kraken, such as Tesseract used by Kettunen et al. (2020). Both the text line segmentation algorithm and the enhancement prediction could also be further refined.

Another area of interest that could lead to improved accuracy is postprocessing. Here, promising approaches certainly come in the form of large language models and embeddings as in Nguyen et al. (2020) and in Soper et al. (2021).

With the improvement of an estimated 28 million text lines, the project of correcting OCR of the BnL collections has been a success. The assumption that OCR can be corrected at scale through automated processes has been confirmed. It remains to be seen what conclusions the BnL can draw from this in terms of new digitisation projects. One option is to update the tender requirements and enforce a minimum OCR level, whose quality could be assured by running batches through the enhancement prediction (Schneider & Maurer, 2022) to estimate whether Nautilus could still improve them. Alternatively, the specifications could remain unchanged since the BnL could run Nautilus automatically on newly digitised documents. In both cases, the BnL will be able to provide better services to the patrons and researchers.

References

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools, 25*(11), 120–125.

Breuel, T. M. (2008). The OCRopus open source OCR system. *Proc. SPIE 6815, Document Recognition and Retrieval XV, 68150F*. Electronic Imaging 2005, San Jose, California, USA. https://doi.org/10.1117/12.783598

Carrasco, R. C. (2014). An open-source OCR evaluation tool. *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage – DATeCH' 14* (pp. 179–184). https://doi.org/10.1145/2595188.2595221

Hegghammer, T. (2022). OCR with Tesseract, Amazon Textract, and Google Document AI: A benchmarking experiment. *Journal of Computational Social Science, 5*(1), 861–882. https://doi.org/10.1007/s42001-021-00149-1

Kettunen, K., Koistinen, M., & Kervinen, J. (2020). Ground truth OCR sample data of Finnish historical newspapers and journals in data improvement validation of a re-OCRing process. *LIBER Quarterly, 30*(1). https://doi.org/10.18352/lq.10322

Kiessling, B. (2019). Kraken - a universal text recognizer for the humanities. *Digital Humanities Conference 2019 (DH2019)*. https://doi.org/10.34894/Z9G2EX

Levenshtein, V. (1965). Binary codes capable of correcting spurious insertions and deletions of ones. *Problems of Information Transmission*, *1*, 8–17.

Luxemburger Wort, (1942). Neues Kleid. *Luxemburger Wort, 2.3.1942*(61), 1. https://persist.lu/ark:70795/g3vmw4/pages/1/articles/DTL47

Maurer, Y. (2017). Improving the quality of the text, a pilot project to assess and correct the OCR in a multilingual environment. *Relying on News Media. Long Term Preservation and Perspectives for Our Collective Memory*. https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa2-164455

Neudecker, C., Baierer, K., Federbusch, M., Boenig, M., Würzner, K.-M., Hartmann, V., & Herrmann, E. (2019). OCR-D: An end-to-end open source OCR framework for historical printed documents. *Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage* (pp. 53–58). https://doi.org/10.1145/3322905.3322917

Nguyen, T. T. H., Jatowt, A., Nguyen, N.-V., Coustaty, M., & Doucet, A. (2020). Neural machine translation with BERT for post-OCR error detection and correction. *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020* (pp. 333–336). https://doi.org/10.1145/3383583.3398605

Schneider, P. (2021). Combining morphological and histogram based text line segmentation in the OCR Context. *Journal of Data Mining & Digital Humanities*, 2021 (HistoInformatics). https://doi.org/10.46298/jdmdh.7277

Schneider, P., & Maurer Y. (2022). Rerunning OCR - A machine learning approach to quality assessment and enhancement prediction. *Journal of Data Mining & Digital Humanities*. https://doi.org/10.46298/jdmdh.8561

Smith, R. (2007). An overview of the Tesseract OCR engine. *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), Curitiba Brazil* (pp. 629–633). https://doi.org/10.1109/icdar.2007.4376991

Soper, E., Fujimoto, S., & Yu, Y.-Y. (2021). BART for post-correction of OCR newspaper text. *Proceedings of the Seventh Workshop on Noisy User-Generated Text (W-NUT 2021)* (pp. 284–290). https://doi.org/10.18653/v1/2021.wnut-1.31

The Luxembourg Government. (n.d). *The AI4gov initiative*. Retrieved November 2, 2022, from https://gouvernement.lu/en/dossiers.gouv_digitalisation%2Ben%2Bdossiers%2B2021%2BAI4Gov.html

Van de Camp, M. (2008). *Explorations into unsupervised corpus quality assessment* (Doctoral dissertation. Tilburg Univiersity, The Netherlands). Retrieved November 9, 2022, from http://ilk.uvt.nl/downloads/pub/papers/hait/camp2008.pdf

Notes

[1] URL: https://eluxemburgensia.lu.

[2] URL: https://github.com/natliblux/nautilusocr.

[3] URL: https://data.bnl.lu/data/historical-newspapers.