https://doi.org/10.2352/issn.2168-3204.2020.1.0.72

Machine Learning and IIIF in the reality check of daily digitization projects using the example of the Goobi Community

Steffen Hankiewicz; intranda GmbH; Göttingen, Germany
Oliver Paetzel; intranda GmbH; Göttingen, Germany

Abstract

Machine Learning and IIIF are popular topics today when it comes to digitisation projects and digital humanities. But are these really practical topics or just buzzwords? Are these rather exclusive technologies of some elite cultural and research institutions? Or can everyday digitisation projects with less exquisite materials really benefit from such technologies?

The example of the community around the open source software Goobi shows what the reality of numerous digitisation projects really looks like. What is no longer just theory and can be used in everyday life without having to develop software yourself? And what added value can actually be expected here?

Introduction

Digitisation projects at cultural institutions have sprung up like mushrooms in recent years. While 20 years ago it was only a few rather elite institutions that were able to produce high-quality digitised material, today many small libraries, museums and archives already have their own digitisation studios with professional equipment. This makes the pure production of high-quality images relatively easy nowadays. But is this really enough? After all, uncategorized directories of images are nothing more than unused data sources.

However, the luxury of manual metadata extraction to generate further research data and thus new possibilities for the use of the digitised material is time-consuming and costly. For this reason, in almost every cultural institution, digitised material is stored on external hard disks and in network shares, which "only" have to be indexed in order to be able to publish or use them "soon". Can new technologies help to answer the question of when this "soon" is?

Machine Learning only for the big boys?

In recent years, machine learning has advanced to become one of the current trend technologies that promise solutions for many problems. Google, Facebook and co. are showing us how, for example, user data and metadata can be used by means of artificial intelligence to create new lucrative business models. And some major players from the cultural institutions scene (e.g. British Library [1] and Library of Congress [2]) are also launching projects to take their first steps in the field of machine learning. Initially, the focus seems to be on concrete, manageable projects with a clearly defined data stock, to automatically open up new metadata. It is foreseeable that these prestigious projects can expect positive results, especially since these are pilot projects with an enormous budget and at the same time a manageable, homogeneous data stock. But what is more exciting for the cultural institutions scene is the question of when small digitisation projects will be able to benefit from such new technical processes without having to have their own development know-how. And will this then also produce the desired results with their own materials?

Our motivation

As developers of the open source software Goobi [3], we are in close contact with numerous cultural institutions in several countries and support a large number of digitisation projects. Over the last 15 years, Goobi has spread enormously in terms of coordinating workflows and publishing digital collections and continues to present us as developers with new challenges. In addition to providing support for ongoing digitisation projects, our main focus is on the constant development of the software. The particular difficulty here is to make these developments as generic as possible so that other users and projects with their respective materials and workflows can also benefit from the developments. We also try to apply this approach to our development in the area of machine learning. However, this turns out to be much more complex than initially hoped for, not least because the set of data (materials, publication types, printing formats, languages etc.) is extremely heterogeneous across all projects.

Previous method of operation

Software development on Goobi over the past few years has primarily been in terms of efficiency and usability. This was particularly evident in the way new functionalities were implemented. Either new functions were developed within plug-ins that were encapsulated by the software core so that they corresponded as closely as possible to the individual usage scenario. Or the development was carried out within the software core of Goobi in such a way that other Goobi users could use it in as many different projects as possible. In the past, it has always been possible to find a good solution for typical database functionalities with forms for data capture and visualisation, which has had a positive effect on the efficiency and operation of the software. This can be measured mainly by mouse paths, the number of clicks, the necessity of scrolling and the basic question of how intuitive the operation of a software is.

New requirements

As Goobi has become more widespread in more and more institutions, some of which have widely differing requirements, we as developers have been confronted with new issues more and more frequently over the years. In addition to pure data capture on the basis of metadata and form fields, questions relating to possible data analyses or data processing based directly on the image files are playing an increasingly important role. Here, however, we repeatedly encounter the following two major challenges:

- It is often very individual data within a specific project with a relatively small data volume.
- The implementation of data analysis is so specialized that it is difficult to transfer it to other projects in order to apply it there as well.

These two major challenges contribute to the fact that the reusability of costly developments of data analyses for other projects, other institutions and other data may be severely limited. This will certainly manifest itself in a similar way in the projects of the British Library and the Library of Congress. For this reason, we are trying the following approach within the Goobi Community:

"If functionality based on machine learning is to be used, the future provision of new ground-truth data must also be guaranteed."

It is not always possible to actually implement this plan. In some projects, however, Ground Truth can be created without additional manual work.

Ground Truth as the key to long-term success

Put simply, modern algorithms based on machine learning must first be trained with clean raw data. By describing selected properties, an algorithm can learn to distinguish and classify objects. From such properties (e.g. body size, head shape, fur, nose), features emerge that allow the algorithm to distinguish and classify new objects (e.g. dog vs. cat) after training on reliable data as ground truth. The more precise and extensive the Ground Truth data is available, the more easily future new objects can be recognized. However, the difficulty here is that the generation of Ground Truth data is often a very time-consuming manual activity that must be carried out with a high quality of results in order for the subsequent training to be successful. In addition, there is the particular problem that after such a complex Ground Truth data generation has been carried out, further data is added at a later point in time, which ideally should also be considered in the future in order to be able to recognize the objects in another context (e.g. dog and cat from behind) or even to allow further object classifications (e.g. mouse and elephant). In such a moment, it is crucial whether the generation of Ground Truth data was planned in the past in such a way that it can be enriched by additional data of the same quality. Only in this way can the machine be trained with new or additional data at any time. This is also the only way to ensure that a complex implementation can be transferred to other projects and other types of data. If, on the other hand, the Ground Truth data cannot be enriched or extended, or only at great expense, long-term usability of the machine learning process is not guaranteed, since it is based on data that is no longer valid.

Machine learning within the Goobi community using concrete examples

Within the Goobi community, we have been working intensively on various approaches to machine learning and neural networks since 2014. Some of the milestones in these developments will be outlined here, together with brief technical explanations of implementation and Ground Truth generation and an assessment of the extent to which the results of implementation can be used by other users with different data.

Example 1: Named Entity Recognition within full text

The first project, where Machine Learning was to be used for Goobi in 2014, involved the automatic labelling of named entities within full texts. Based on the algorithms of Stanford Natural Language Processing, the aim was to automatically mark people, places and entities within ALTO files from the previous OCR process. The goal of the implementation was in particular to enable different summarizing views of the contents of books in order to visualize the mentioning of e.g. persons within defined page areas of a book.

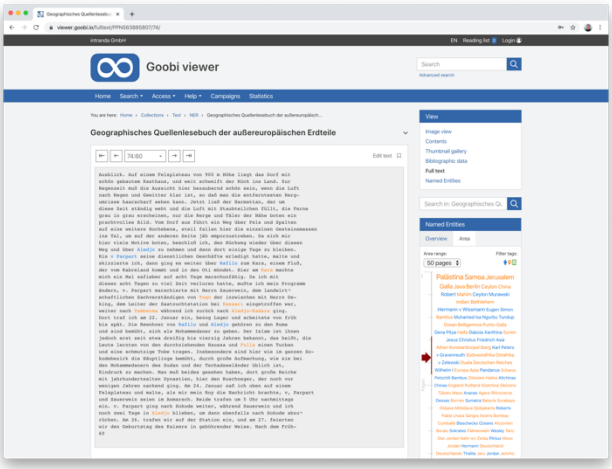

Figure 1. *Result display of the Named Entity Recognition*

Technical background of the implementation

The implementation is carried out in the form that standardized ALTO files, such as those that can be generated from OCR systems, are analyzed. The entities determined there within the continuous text are marked as tags and thus enrich the ALTO files with named entities.

Stanford NLP and specifically its Named Entity Tagger based on Conditional Random Fields was used as a machine learning platform [4]. Since there were no training data with suitable licenses for the German language available in 2014, we first had to create them ourselves.

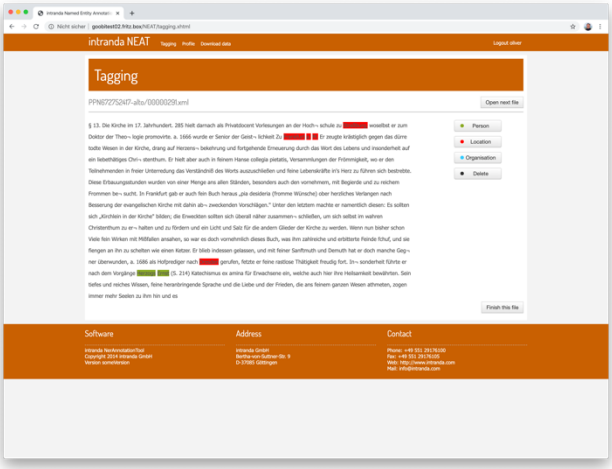

Figure 2. *User interface for the enrichment of further ground-truth data*

Ground-Truth: Data format, expandability and reusability

The Ground-Truth data format is based on the standard format for Stanford NER: A text file with each word on one line with the respective tag separated by a space next to it. The end of a sequence

(a sentence) is marked by a blank line. The training data can be created and extended with a number of tools, including the Named Entity Annotation Tool (NEAT) developed by us.

We initially based our training data on 19th century German textbooks. If you want to use other languages in the future, there is a relatively large number of freely available models on the Stanford NLP website.

Example 2: Generation of catalog records from tables of contents

In 2017, we developed a system for semi-automatic cataloguing of conference proceedings for the German National Library of Science and Technology in Hannover (Germany). The aim was to evaluate digitised versions of printed tables of contents in such a way that for each publication within conference proceedings a data record can be generated for a library catalogue for which the following elements can be recognised in each case:

- the entire entry
- the title
- the participating authors
- the institutions from which the authors come
- the page number of the article within the conference proceedings

A Goobi plugin was used to visualise the recognised entities for possible corrections.

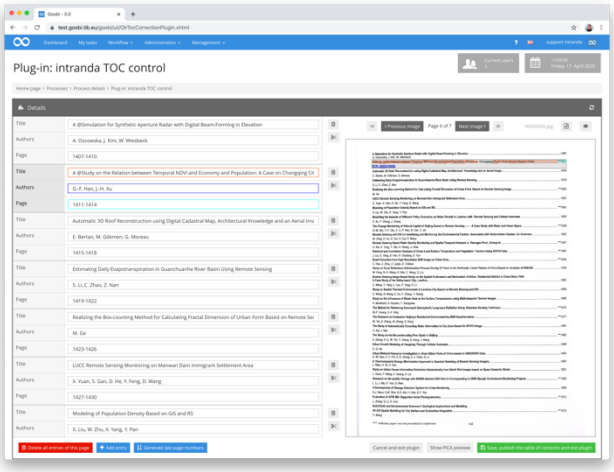

Figure 3. *Goobi plug-in for displaying and correcting recognised entities, each with title, authors, institutions and start page*

Technical background of the implementation

The analysis works like the Named Entity Recognition on OCR results in ALTO format. The output is a proprietary XML format that contains a structured table of contents with page numbers, titles, authors and institutions. These results can be corrected in Goobi workflow before enriching the catalogue.

The recognition itself uses a total of three machine learning algorithms:

- Hierarchical clustering with a customised distance function for clustering the lines recognised by OCR in entries in the table of contents
- A Support Vector Machine (SVM) to recognize the page numbers within the entries
- A Conditional Random Field (CRF) to classify the remaining words in each entry into the classes "Title", "Author" and "Institution

For the page number SVM we get an F1 value of 0.98, the CRF reaches an F1 of 0.91.

Ground-Truth: Data format, expandability and reusability

Starting material for the implementation were image files from the first pages of several books. These contained blank pages, title pages and the printed tables of contents. From these pages a full text per page was first generated by OCR. For the generation of a Ground Truth we then developed a web-based application that should allow several people to work simultaneously on the dataset of images and full text and to capture the entities with their associated data. By means of mouse clicks and key combinations 5 entities could be marked.

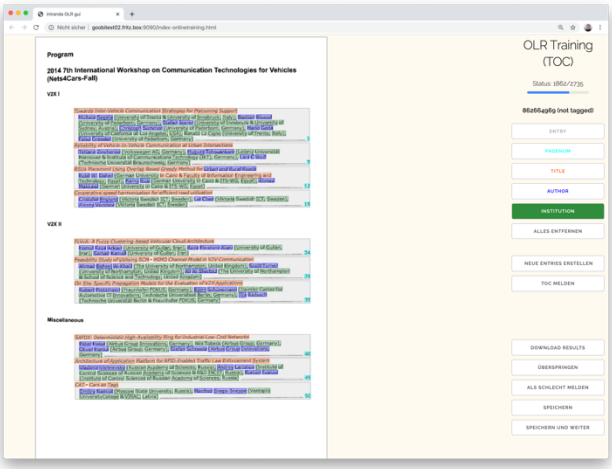

Figure 4. *User interface for the enrichment of further ground-truth data*

The heterogeneity of the tables of contents used for ground truth generation was enormous, so that many different print layouts of tables of contents are already taken into account. Nevertheless, it is expected that more layouts will have to be processed in the future. At the time of development, we did not foresee the long-term usability of the Ground Truth generation for further data and projects. For this reason, too much manual preparation work is currently still required for loading new images and performing OCR as the data basis. For future projects and data this should be changed in such a way that data import can also be carried out via IIIF interfaces and full text generation is carried out on the basis of embedded OCR.

Example 3: Automatic recognition of newspaper editions

Since 2016 we have been working on the automatic recognition of structures within digitized books. This is a very difficult task, especially against the background of different publishers, publication types and layouts. In order to approach this topic we first developed an automatic recognition of newspaper editions within bound newspaper volumes. Especially the uniform layout of

newspapers and their supplements is a great help here. In addition, there are already numerous newspapers that have been digitized and indexed with metadata and are publicly accessible. These can be queried and analysed very well via an interface such as OAI. The recognition of newspaper editions is now part of the daily routine of several Goobi installations, including at the University Library in Basel.

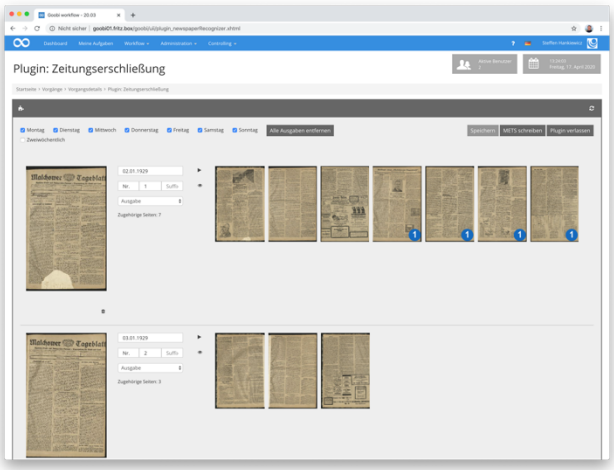

Figure 5. *Goobi plug-in for displaying and correcting recognised newspaper editions within newspaper volumes*

Technical background of the implementation

The input format for this project are scans of newspaper pages. The output is a label for each analyzed image. The two classes are "title page" and "not title page".

For the classification we used Convolutional Neural Nets (CNN). The first used net was a mobileet-v2, which we trained without pre-trained weights. With the release of the new EfficientNets we switched to an EfficentNet-B3 with weights pre-trained on ImageNet. This change has especially improved the performance on newspapers from other publishers that are not in the training set. On validation sets with images from newspaper publishers also included in the training set, both networks perform at eye level with over 99% accuracy.

Ground-Truth: Data format, expandability and reusability

The format of the Ground Truth is easy to handle due to the binary classification task and the input format. It consists of two folders with images in them: one folder with the title pages and one folder with other pages from the newspaper.

We have harvested our initial training set via OAI-PMH from newspapers that have already been indexed manually. For this purpose, the METS files for each year's issue of a newspaper were analyzed and the first and a random other page from an issue were downloaded.

This training data was used to create an initial model that was then used by us and various Goobi users. A Goobi plug-in then allowed us to easily correct the results. As soon as the newspapers are published in the presentation systems, we can harvest them using the method described above and use the extended training data to retrain the model.

As a result of this procedure, our data set has now grown to around 20,000 examples per class. The entire procedure allows fairly straightforward reuse by third parties because the results of the current model can be corrected in Goobi in a user-friendly way and then automatically added to the training set, which improves the recognition rate in subsequent years of the same newspaper.

Example 4: Automatic pagination

Within the metadata acquisition of digitised material, pagination also plays an important role in order to assign the original printed page numbers to the digitised material and thus make content more easily searchable. In most Goobi digitisation projects, pagination has been used from the beginning and such manual data entry usually takes between 1 and 5 minutes per book. With an implementation based on machine-learning, we would like to minimise this required working time even further. The fact that, despite different languages and publication types, page numbers can often be found at similar positions within book pages makes this implementation relatively simple. We started this development in 2019 and presented a first prototype at a Goobi user meeting. Within the next few months, this development will become a standard function in many institutions, which means that, on the one hand, some working time will be saved and, on the other hand, it will now be possible to record pagination information for those who have not previously recorded it.

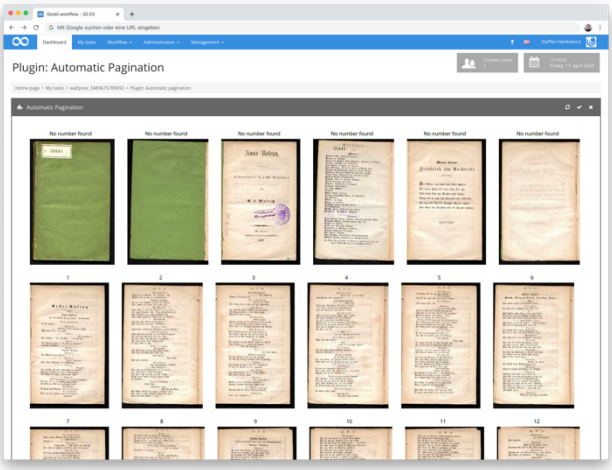

Figure 6. *Goobi plugin for displaying and correcting pagination information for each page*

Technical background of the implementation

Input format here are again OCR results in ALTO format. The recognition of the page numbers is done by a Support Vector Machine. Each word marked in ALTO is sent through the classifier and receives one of the labels "page number" or "no page number".

After this recognition, the candidates are cleaned up by manually written algorithms and possible outliers (for example, page number 12 is followed by page number 18) are marked. This information is then displayed to the user in Goobi workflow and the results can be corrected there.

The SVM achieves an F1 score of 0.98.

Ground-Truth: data format, expandability and reusability

The ground-truth data is available here in a JSON format in which a label is assigned to each word in an ALTO file. For our model we have manually marked all page numbers in 168 books. The books were printed between the 17th and 19th century. So the

training set is quite heterogeneous and should work well on most of the works, which makes it well reusable for further facilities.

Example 5: Performing OCR

Full texts are indispensable for an extensive search within digital collections. We have been using Goobi to perform OCR on a large scale for many years. However, instead of continuing to use commercial solutions, we decided in 2018 to migrate completely to Tesseract as an open source solution and only perform full text recognition using it. The fact that Tesseract version 4 can be trained on the basis of neural networks allows us to influence the recognition quality. In this context, however, it turned out that even independent of the pure recognition of letters and words, many other work steps have to be considered, which have to take place before the actual text recognition. For this reason, numerous pre-processing steps had to be carried out on the basis of the images in order to achieve better OCR results. In addition to high-quality binarisation of the images, this also includes segmentation of the pages in order to transmit only those fragments of an image to the OCR engine in which text content is present.

The segmentation in particular presented us with major difficulties with regard to the Ground Truth data situation. A manual creation of Ground Truth data for page segments was not realistically feasible in terms of personnel, so that we had to take a different approach.

We also had to find new ways to create Ground Truth with regard to the use of different fonts. For this purpose we developed an application that allows us to generate complete alphabets for various historical fonts that are actually used, and thus in turn generate historical texts.

With these Artificial Ground Truth data, which were first publicly presented at the Archiving Conference 2019, the OCR quality can be significantly influenced.

Technical background of the implementation

For the page segmentation we use a U-Net, which was trained with automatically generated data. The network abstracts well enough to provide very useful results on real-world data. After the pixels of an image have been labeled by the neural network, we neutralize all non-text pixels with an average of all background pixels and send the result to OCR. For certain difficult layouts, this pre-processing achieves a 20% reduction of the character error rate compared to baseline-tesseract.

Figure 7. *Examples of artificially generated page layouts with different images, frames and fonts*

Ground-Truth: Data format, expandability and reusability

The training data for the page segmentation is generated as mentioned above with random layouts by a specially written program. At the same time as the artificial layouts, a label file is always generated, which is an image with the same size that assigns a label to each pixel in the original image by color. Due to the random arrangement of the layouts, this neural network has so far worked for all layouts we have found. Only ornamental frames had to be manually cut out of real data and added to the automatically generated ones.

The creation of a primitive "font" for generating new training texts for tesseract and other line OCR works with the following steps:

Figure 8. *Transcription of text lines*

1. The lines are transcribed manually and the results are stored in a database. This information is essential for performing the second step.

Figure 9. *Selection of the letter to be determined within a line*

2. In the second step, the program gradually shows the user a line containing a letter of the alphabet, which the user then marks. In this way, a primitive font (with only one font size) is created with relatively little effort.

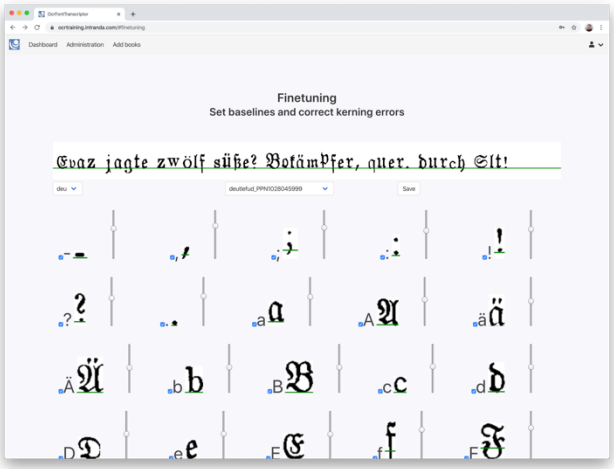

Figure 10. *Adjusting the baseline for an entire alphabet of a specific font*

3. This font only needs to be adjusted in a third step to set the baseline and improve kerning.

Such a font can thus be used to set new texts, which can then be used to train line OCR.

Both methods are already in use at our company and benefit all Goobi OCR users. The method for creating the primitive fonts can be used by all facilities with IIIF interfaces because the images and ALTO results with the line coordinates are retrieved via IIIF API.

Example 6: Layout analysis for straightening and cropping

The creation of digital copies is sometimes associated with enormous effort, so that any automation to simplify manual work is welcome. The same applies to the question of how generated images can be prepared in such a way that they can be optimally reused, e.g. for display within digital collections. In general, the mostly black backgrounds on which the books lay at the time of digitization are just as disturbing as areas of the respective opposite page beyond the book fold. And even minimal rotation of the books during scanning is already a disturbing factor for the display of the results or for the execution of OCR.

With the LayoutWizzard plugin for Goobi, we have been analysing image files and performing ideal post-processing since 2016. This enables us to automatically generate derivatives from the master images that are straightened, where black outer edges are cut off and image content is only taken into account up to the recognised book fold. However, as Goobi became more widespread and requirements were constantly being added, there was always a need to adapt the recognition logic. For this reason, we are currently converting this plugin to machine learning. The recognition of the book fold and the logic based on it for distinguishing right and left pages has already been converted here and delivers significantly better results than the previous rule-based implementation. The other components are expected to be converted successively in the coming months. The current 15 or so institutions in the UK, Germany, Israel and the USA that already use the Goobi cropping plugin on a daily basis will automatically be able to start using this development with an update and save manual work with the improved layout analysis.

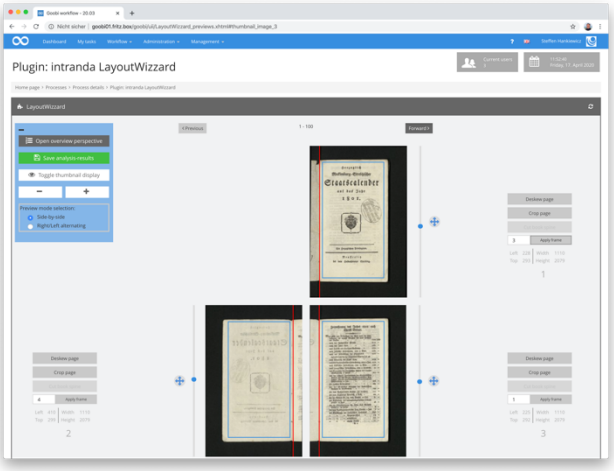

Figure 11. *Goobi plug-in for displaying and correcting recognised layout information for deskewing and cropping digitised material*

Technical background of the implementation

The input for the LayoutWizzard are always images. For the recognition of the fold with machine learning methods we use the same kind of neural networks as for page segmentation in OCR preprocessing. This means that we need images of the same size as the label, in which the fold is traced. However, the LayoutWizzard works with XML files in which the X-position of the fold is recorded as a number. To solve this problem, we have written a program that converts the manually corrected data of our customers into the input format of the neural network. With this data the net was trained. After the neural network has generated a label image for the image to be processed during operation, we do an interpolation and get again a number for the X-position of the fold. So the use of the LayoutWizzard does not change - only the automatically generated results get better.

Ground-Truth: Data format, extensibility and reusability

The data format is an XML format as described above. This can then be used to create training images for the neural network. All manual corrections made by a user can therefore be incorporated into the training data with very little effort. The models are still quite limited at the moment, because we only trained with data from our in-house scanned projects. However, due to the data storage and the simple conversion into training data, this procedure can be adapted for any interested institution.

Conclusion

As you can see from the examples given here, we have also gone through various phases in the Goobi Community in using machine learning methods for different purposes. It became apparent that the models we trained became better and better over time as we provided them with more and better training data. In this respect, the decision that further Ground Truth generation should ideally result directly from correction activities within Goobi is the most promising. However, if this is not possible because of the data available, we have come to the conclusion that Ground Truth

generation should be web-based wherever possible. This will ensure that several people can work on a data collection process simultaneously and in a decentralised manner. In many cases, we were also able to integrate assistants from the scientific institutions in order to enrich the existing Ground Truth data generated by us with their project-specific data. For this purpose, it is crucial that the generation of the Ground Truth is as intuitive and efficient as possible in order to achieve a usable amount of data as easily as possible. Where possible, the integration of further external data sources via standardized interfaces such as IIIF should be planned. This minimizes the effort considerably if new data is to be compiled for future trainings. At the same time, it considerably expands the pool of potential data for future trainings.

In summary, it can be said that machine learning methods are indeed slowly gaining a foothold in the world of cultural institutions. Selected projects with exclusive content have already shown demonstrable success. However, the decisive factor for the future will be that the generation of Ground Truth must be as simple as possible to allow repeated new training of the models. This will not only significantly simplify lengthy monotonous metadata captures and automate work, but will also create numerous new troves of research data.

References

[1] https://www.bl.uk/press-releases/2018/december/living-with-machines

[2] https://www.loc.gov/digital-strategy/

[3] https://goobi.io

[4] https://nlp.stanford.edu/software/CRF-NER.html

Author Biographies

Steffen Hankiewicz is a senior software developer, CEO and owner of the German software company intranda GmbH. He has been developing and implementing software solutions for digitization projects for more than 16 years. The open-source workflow management and publishing suite Goobi as well as several automatic tools for cropping, validation, conversion and many other software for handling 2D and 3D material are some of the current digitization tools he develops and supports together with his team in 17 countries.

Oliver Paetzel studied applied computer science in Göttingen (Germany), specializing in digital humanities. He joined the German software company intranda GmbH as a software developer in 2012 and concentrates on machine-learning technology and workflow automation. As product manager, he is also responsible for the development of the open-source workflow management tool Goobi.