Quality Prediction System for Large-Scale Digitisation Workflows†

Christian Clausner, Stefan Pletschacher and Apostolos Antonacopoulos
Pattern Recognition and Image Analysis (PRImA) Research Lab
School of Computing, Science and Engineering, University of Salford
Greater Manchester, M5 4WT, United Kingdom
www.primaresearch.org

***Abstract*—The feasibility of large-scale OCR projects can so far only be assessed by running pilot studies on subsets of the target document collections and measuring the success of different workflows based on precise ground truth, which can be very costly to produce in the required volume. The premise of this paper is that, as an alternative, quality prediction may be used to approximate the success of a given OCR workflow. A new system is thus presented where a classifier is trained using metadata, image and layout features in combination with measured success rates (based on minimal ground truth). Subsequently, only document images are required as input for the numeric prediction of the quality score (no ground truth required). This way, the system can be applied to any number of similar (unseen) documents in order to assess their suitability for being processed using the particular workflow. The usefulness of the system has been validated using a realistic dataset of historical newspaper pages.**

Keywords—Document analysis; Quality prediction; Digitisation; Performance evaluation; Supervised learning; Numeric prediction; Ground truthing; Large-scale

I. INTRODUCTION

Large-scale digitisation projects are faced with a fundamental question: how to achieve maximum impact with the allocated resources. A balance has to be found between sufficient quality and quantity (in the sense of creating critical mass) in order for the results to be received as useful. Quality, however, is not straighforward to assess – especially in advance. Hence, there is a great need for tools which can be used by planners and project managers to make informed decisions regarding the selection and prioritisation of material.

The Europeana Newspapers Project (ENP) [1] was an EU-funded Best Practice Network under the theme CIP-ICT-PSP.2011.2.1 - Aggregating content in Europeana. The project successfully achieved its ambitious goal of recognising and making vast amounts (over 11 million newspaper pages) of searchable historical newspapers available via the two most prominent European cultural heritage websites, Europeana and The European Library. The project also had the goal of creating a quality estimation toolkit to support future digitisation and OCR projects in the decision making process.

In the literature, several approaches exist for evaluating the quality of images in general, mainly for selecting the most appropriate enhancement methods to apply.

With respect to OCR quality prediction, Blando et al. [2] represents the earliest attempt to classify page images as OCRable or not. The method requires page images to be segmented first and regions of text to be extracted. It works on bitonal (thresholded) images and has a reported prediction accuracy of 85%. The use of heuristic features however, motivated by observations on relatively uniform document collections, is not appropriate in large-scale digitisation applications, a view also supported by Ye and Doermann [3]. The latter train a classifier based on the correspondence of OCR recognition rate with different randomly selected small image patches. It should be noted that both of the above methods use character rather than word accuracy, despite the latter being more meaningful in most application domains.

The use of image features (albeit in small patches in each image) in [3] is a step forward in trying to associate input page images with OCR quality levels. However, several characteristics of an input page (the image as a whole, associated metadata and the results of simple processing) may be useful in predicting the quality of OCR more accurately.

Moreover, for such example-based approaches it is crucial to have a truly representative dataset to train them on.

It is worth noting that a distinction must be made between the methods relevant to this paper (OCR quality prediction based on a page image) and those methods attempting to evaluate the quality of existing OCR results in the absence ground truth (e.g. [4]).

The system proposed in this paper has in its disposal a large number of possible features associated with the actual document, not only the scanned page. A suitable classifier is trained on carefully selected features and the system is evaluated on a truly representative dataset of historical newspapers. The system is extensible and reconfigurable, using a freely available feature selection and classification system.

All aspects of the system are described in detail in the next section. In Section III, the system is experimentally validated and the results are discussed thoroughly. Concluding remarks are made in Section IV.

II. QUALITY PREDICTION SYSTEM

The aim of the estimation is, based on just the scanned pages, to predict the quality of digitisation results that a given OCR pipeline would produce. This can be achieved through feature-based numeric prediction using a specialised classifier. The estimation workflow has two phases: (1) *Training* of a classifier and (2) *Quality prediction* applying the classifier. The nature of the available features and the selection of the best features for classification are most crucial for the predictive strength of the quality estimation. The following sections provide more details on these points.

†This work has been funded through the EU Competitiveness and Innovation Framework Programme grant Europeana Newspapers (Ref. 297380).

A. System Overview

A typical use scenario involves two processing pipelines – one for training and one for the actual quality prediction. The first one (learning phase) produces a classifier and needs only to be used once or whenever the OCR performance for a new type of material (not present in the original training dataset) is to be predicted. The input to this pipeline is a small set of document images and the respective ground truth (page layout and/or text content, depending on the features that are to be used). The images are processed by the OCR engine and the results are compared against ground truth by a performance evaluation module. In parallel, features are extracted from the document images and associated metadata. Both the evaluation results and the feature values, are then used to train a classifier.

The second pipeline (prediction phase) only requires document images as input. This can be thought of as the production pipeline which can process large amounts of data using the previously trained classifier. The selected features are extracted from the images (including a temporary – on the fly – OCR step with the sole purpose of obtaining features) and the resulting values are fed into the classifier, which outputs the predicted OCR success rate.

B. Feature Set

It is desirable to examine all possible potential features that can be extracted to identify those which show a certain correlation with the classification target (OCR quality). This is, however, quite often difficult to determine – sometimes only combinations of features produce a correlation. Therefore, the most common approach is to define a variety of features and use automated feature selection to find the strongest ones

While their combined use in the overall classification is the main purpose, some features can be interesting in their own right, providing more direct insights into the condition or potential quality of OCR results. The feature "Region Overlaps" (see Table I), for instance, can hint at problems in the segmentation step of the OCR pipeline.

The next three subsections describe all the features that have been used in the quality estimation experiments within the Europeana Newspapers Project.

a) Features from metadata

Some basic features are usually available as metadata that is stored together with the document image. For the conducted experiments these were:

- Language (e.g. English, German, OldGerman)
- Font (normal, gothic, mixed)

b) Image, page layout, and text features

In addition, a range of features (potentially relevant to quality prediction) based on document image, page layout, and detected text were defined. Since page layout representation and text content require prior processing with some OCR system, these resources can either be obtained using the production workflow system under investigation (for which the quality prediction classifier is to be created) or, as an alternative, using an open source OCR system (or in fact any system that is publicly available).

To this end, two software tools for retrieving features were developed by the authors:

- PRImA FeatureExtractor (for image and layout related features)
- PRImA JFeatureExtractor (for text related features)

Figure 1 shows the general processing pipeline for extracting features from an image for quality estimation (metadata features are not shown).

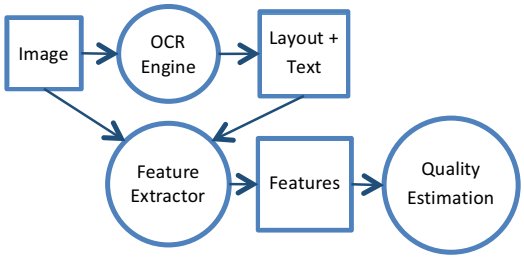

Fig. 1. Feature extraction from a page image and application.

It should be noted that the OCR engine used for feature extraction does not have to be the same as the OCR whose quality is to be estimated – the open source OCR system Tesseract [5] can be used instead. This approach was followed in the experiments that are described in Section III.

TABLE I. IMAGE, PAGE LAYOUT, AND TEXT-RELATED FEATURES

#	Feature	Description
1	Colour Mode	Colour mode of the document image: "1-bit" black-and-white (bitonal), "8-bit" greyscale or "24-bit" RGB colour.
2	Image DPI	Image resolution in pixels per inch as specified within the document image. Only the horizontal resolution is taken into account.
3	Image Tile Count	Represents the number of tiles the image can be split into (default tile 300x300 pixels). Cut off tiles at the edges of the image count as full tiles.
4	Foreground Pixel Density	Number of black pixels within the black-and-white (bitonal) image in relation to the overall area (0.5 means half of the image is black).
5	Connected Component Count	Represents the average number of connected components within a tile (default 300x300 pixels) of the black-and-white (bitonal) image.
6	Image Noise	This feature estimates the noisiness of the document image using a non-local means filter approach. For performance reasons, the feature area is limited to a maximum of 1000x1000 pixels (window at the centre of the image).
7	Image Brightness	Contains the average grey value of the greyscale image. The average grey value is equal to the image brightness. A value of 1 corresponds to a white image whereas a value of 0 corresponds to a fully black image.
8	Image Contrast	Represents the contrast of the greyscale image. It is determined by calculating the standard deviation of the grey value of all pixels. If the original document image is a black-and-white (bitonal) image the contrast is always 1.
9	Edge Detection	This feature is based on calculating the mean greyscale value of the result image created by the Laplacian operator applied to the greyscale image. The Laplacian operator is used for edge detection. The feature is therefore and indicator for the sharpness of the image (more sharp = more detected edges).

#	Feature	Description
10	Brightness Unevenness	Represents the brightness distribution across the greyscale image. It is calculated as the normalised standard deviation of the brightness of an image tile (default 300x300 pixel) in relation to the overall image brightness.
11	Layout Region Count	Total number of regions (blocks) in the document page layout.
12	Text Region Count	Total number of text regions (blocks) in the document page layout.
13	Region Overlaps	Number of region overlaps (overlaps across different layers are disregarded).
14	Foreground Outside Regions	The number of black pixels of the black-and-white (bitonal) image that are not within layout regions.
15	Regions without Foreground	The number of layout regions that have less than 1% foreground pixels (black pixels within the black-and-white / bitonal image).
16	Missing Region Text	Number of text regions that do not have any text content (Unicode).
17	Text Line Count Mismatches	Number of text regions where the number of text lines within the text content does not match the number of child text line objects.
18	OCR Confidence	Represents the average text recognition confidence of the OCR engine that was used to analyse the page (e.g. Tesseract).
19	Word Count	Represents the total number of words within the text content.
20	Words with Digits	Represents the number of words that contain at least one digit.
21	Alphabetic Character Count	Represents the count of non-whitespace characters.
22	Whitespace Count	Represents the number of whitespace characters.
23	Digit Count	Represents the number of characters that are digits.
24	Punctuation Count	Represents the number of punctuation characters (according to POSIX Bracket Expressions: [!"#$%&'()*+,\-./:;<=>?@[\\\]^_`{|}~])
25	Average Word Length	Represents the average length in characters of a word within the text content.
26	Words Occurring Once	Represents the number of words that occur exactly once in the text in relation to the total number of words.
27	Word Repetition	Represents the number of unique words (excluding repetitions) in relation to the total number of words (including repetitions).
28	Words in Dictionary	Represents the number of words that could be found in the used dictionary in relation to the number of all words. Requires a dictionary as static input (for a given language).

Table I details all features that were defined and are used within the system. They range from basic count-based values to results of complex image and text processing operations. At this stage, the choice of features was driven by what could be calculated and/or readily obtained from the data.

c) Combined Features

It can be beneficial to combine two weak features to create one strong one. For example, the features "Words with digits" and "Word count" can be combined to "Words with digits (relative)" by dividing one by the other. It can also be observed that relative values (e.g. ratios) are better than absolute values for learning general concepts. Furthermore, some complex features can be split into several simpler features (for instance binary – 0/1). Table II shows the additional features that have been used in the experiments.

TABLE II. COMBINED AND SPLIT FEATURES

#	Feature	Description
29	Words with Digits (Relative)	Represents the number of words that contain at least one digit, divided by the total number of words.
30 31 32	Bitonal Greyscale Colour	Splits the feature 'Colour mode' into three separate features with 0/1 values. This can be beneficial for some classification methods.

C. Feature Selection and Classifier Training

Several methods for feature selection and classification have been reported in the literature. The open source tool WEKA [6], by the University of Waikato (New Zealand), provides a good selection of standard implementations. It also comes with an excellent user interface for experimentation and even allows creation of workflows.

In WEKA, for the problem at hand, only classifiers that produce a numeric value are of interest, since the target is to predict a success rate. This is referred to as numeric prediction.

a) Data preparation

In order to be usable in WEKA, the input data has to fulfil the following criteria:

- One table including feature values and target quality values (that are to be estimated by the classifier)
- Clean data (avoid missing values or invalid numbers such as "NaN")
- Specific WEKA file format (ARFF); can be converted from comma separated values (CSV)

It should be noted that WEKA uses its own vocabulary. Features are called attributes and a data table represents the instances of the attributes, where one instance equals a row of the data table. For ease of use the WEKA Explorer allows removing of attributes and instances from the data. It also supports different data file formats.

b) Feature Selection

WEKA offers several heuristics to select the best features for a classifier. After loading training and test data, the classification target has to be selected (e.g. "Bag of Words" OCR success rate).

Finding good classifiers is an iterative process of selecting features and classifying using different approaches until a satisfactory result has been obtained. WEKA provides several classifiers for numeric prediction. Table III lists classifiers that showed the most promise for the task of numeric prediction for quality estimation.

a) Classifier Training

Classifiers are trained purely on the training set and afterwards evaluated using the test set. To state the success of a classifier we use the mean error (absolute difference between predicted and actual value). Since we want to predict OCR and

layout evaluation quality, the mean error can be expressed as a percentage. For instance, a mean error of 12% means that, on average, the predicted result differs 12% from the actual result (0% would be the optimum).

TABLE III. EXAMPLES OF CLASSIFIERS IMPLEMENTED IN WEKA

Classifier (WEKA)	Description
Gaussian Processes	Implements Gaussian Processes for regression without hyperparameter-tuning. [7]
Linear Regression	Class for using linear regression for prediction. Uses the Akaike criterion for model selection, and is able to deal with weighted instances.
SMOReg	Implements the support vector machine for regression. The parameters can be learned using various algorithms. The algorithm is selected by setting the RegOptimizer. The most popular algorithm (RegSMOImproved) is due to Shevade, Keerthi et al and this is the default RegOptimizer. [8]
IBk	K-nearest neighbours classifier.[9]

D. *Quality Prediction Tool*

In order for the presented approach to be usable by a wider audience (researchers, librarians, archivists etc.) a dedicated quality prediction command line tool was implemented within the scope of the Europeana Newspaper Project.

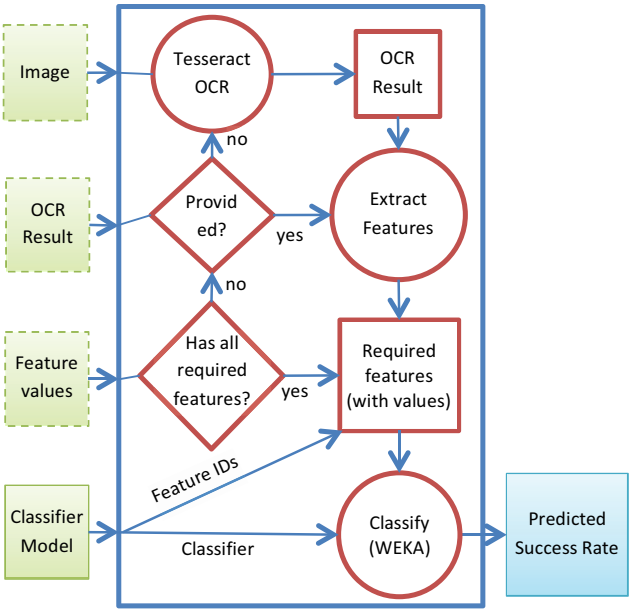

Fig. 2. Diagram of the PRImA Quality Prediction tool (optional input data with dashed outline; at least one required)

The tool has been designed to determine from a given classifier model which features are required for the quality prediction. It then analyses the provided input sources (Comma Separated Values (CSV) with pre-computed features, direct feature input, OCR result, and document image) and extracts any missing features by running an integrated OCR engine and/or using feature extraction methods. To this end, the Quality Prediction tool is linked with several other tools developed by the authors:

- TesseractToPAGE tool (a wrapper for the Tesseract OCR engine)
- Page Converter (to apply optional text filter rules)
- Text Exporter (to serialise the text content of OCR results in PAGE format [10])
- Feature extractors

The estimated quality value is returned directly to the command line and can be output to a text file or any other type of data stream. In addition, a file containing a table with all features and the quality value can be produced optionally (WEKA file format). Fig. 2 provides a schematic overview of the functioning of the tool.

III. EXPERIMENTS AND DISCUSSION

This section provides an overview of the experimental validation carried out on a uniquely representative dataset. Following a description of the experimental setup and of the different aspects of the system, the prediction results are compared to those of the actual workflow of the Europeana Newspapers Project and discussed in detail.

A. *Dataset*

The publicly available Europeana Newspapers Project (ENP) dataset [11] has been used in the experiments. It contains a diverse set of over 500 scanned newspaper pages representative of the digitisation projects of 12 national and major libraries in Europe. Full ground truth was created for page layout (region outlines and types) and text content. Furthermore, all pages were processed with two OCR workflows:

- GPP Binarisation [12] + ABBYY FineReader Engine (as defined and used to process 8 million pages for Europeana)
- ABBYY FineReader Engine only

For creating a classifier and reliably testing it, the data had to be split into training and test sets. Accordingly, 50% of the document pages of each institutional subset were randomly selected for training. The rest of the pages were used for testing the classifiers. The reason to confine the randomness to each subset (stratification) is to avoid getting a strong bias for one particular subset by chance.

B. *Performance Measures*

The performance measures used for judging the quality of OCR results were based on the comparison of obtained results against the ground truth, in terms of:

- text – without regarding the order of words – (Bag of Words) as well as
- segmentation and region classification (Layout Analysis Success Rate)

as described in [13].

C. *Feature Selection Results*

Extensive experimentation is pointing towards the WEKA "ClassifierSubsetEval" being the best approach to select features in the context of quality prediction and the given dataset. Thereby a classifier has to be selected beforehand. This method works especially well with a genetic search algorithm (as provided within WEKA). Several feature combinations are

tested with the chosen classifier and are then tweaked over a number of generations (evolutionary optimisation).

Based on several feature selection iterations, the usefulness of specific features could be estimated by counting how often each individual feature was selected. Table IV shows the features that were selected most (in order of relevance). The least relevant features (in this use scenario) are listed in Table V.

TABLE IV. MOST USED FEATURES (FEATURES OCCURRING IN BOTH COLUMNS ARE HIGHLIGHTED IN BOLD)

Features for "Bag of Words" prediction	Features for layout analysis quality prediction
Words in dictionary	**Foreground pixel density**
Words occurring once	Edge detection
OCR confidence	Whitespace count
Language	**Word repetition**
Image DPI	Image contrast
Layout region count	**OCR confidence**
Words with digits (relative)	Alphabetic character count
Text region count	**Words with digits (relative)**
Word repetition	Words occurring once
Font	**Text region count**
Foreground pixel density	**Punctuation count**
Punctuation count	Image noise
	Layout region count
	Word count
	Words in dictionary
	Greyscale

TABLE V. LEAST USED FEATURES

Features for "Bag of Words" prediction	Features for layout analysis quality prediction
Regions without foreground	Average word length
Colour	

D. Quality Prediction Results

As a baseline for comparison, we also calculate the mean error for a naïve prediction. To obtain this, a fixed prediction value is calculated based on the training set by simply using the average of the target (segmentation, OCR etc.) actual quality values. That fixed value is then used as "prediction" for all instances in the test set. The mean error of this approach can be seen as the best result that a quality estimation method with fixed prediction value can achieve. A trained classifier should therefore outperform the naïve approach, to be considered successful.

WEKA provides several classifiers for numeric prediction, of which a Support Vector Machine for Regression (SMOReg) and Gaussian Processes delivered the best results. Table VI shows results for different prediction targets, data subsets, feature selections, and classifiers. In these experiments Tesseract OCR results were used for feature extraction. The prediction targets however, are the quality of layout analysis and text recognition results of ABBYY FineReader (as used in the Europeana Newspapers Project workflow).

To get a better understanding of these figures, a colour coded list of results for individual pages was produced for the subset of Dutch documents referred to in Table VI (training set only); Those can be seen in Table VII. Using a colour gradient, green cells indicate that predicted and actual values are almost equal. Red cells highlight instances with relatively larger mean error. Since the direction in which the prediction is leaning can be important, over-prediction errors have been marked red and under-prediction blue.

TABLE VI. RESULTS OF SELECTED EXPERIMENTS

Prediction target	Dataset (#pages in training and test set)	Baseline (mean error of naïve prediction)	Classifier	Mean error of prediction (test set)
Bag of Words OCR Success Rate	Full (300+300)	14.2%	Gaussian Processes	6.2%
			Support vector machine	6.1%
			Gaussian Processes	7.1%
	English documents (25+25)	8.1%	Support vector machine	2.67%
	Dutch documents (25+25)	11.2%	IBk	2.73%
Layout Analysis Success Rate (Scenario: Keyword search)	Full (300+300)	14.5%	Support vector machine	11.27%
	English documents (25+25)	7.8%	Gaussian Processes	4.42%

TABLE VII. RESULTS PER DOCUMENT PAGE FOR BAG-OF-WORDS EXPERIMENT ON DUTCH DOCUMENTS

Instance	Actual quality	Predicted quality	Error	Absolute error
1	45.3%	46.4%	1.1%	1.1%
2	63.9%	69.2%	5.3%	5.3%
3	90.3%	91.2%	0.9%	0.9%
4	87.8%	88.8%	1.0%	1.0%
5	80.0%	79.1%	-1.0%	1.0%
6	95.9%	94.7%	-1.2%	1.2%
7	54.5%	46.4%	-8.1%	8.1%
8	97.4%	94.7%	-2.7%	2.7%
9	90.9%	93.3%	2.5%	2.5%
10	92.6%	93.3%	0.7%	0.7%
11	63.8%	69.2%	5.4%	5.4%
12	89.8%	96.2%	6.4%	6.4%
13	97.9%	97.6%	-0.3%	0.3%
14	91.8%	96.8%	4.9%	4.9%
15	93.4%	94.0%	0.6%	0.6%
16	49.5%	46.4%	-3.1%	3.1%
17	89.7%	90.2%	0.5%	0.5%
18	95.5%	96.2%	0.7%	0.7%
19	86.3%	90.2%	3.9%	3.9%
20	86.4%	79.1%	-7.3%	7.3%
21	89.8%	90.2%	0.4%	0.4%
22	92.4%	92.0%	-0.3%	0.3%
23	95.3%	94.0%	-1.3%	1.3%
24	84.7%	88.5%	3.8%	3.8%
25	74.0%	79.1%	5.1%	5.1%

From the results it can be observed that the quality estimation is more precise, as expected, for smaller datasets with more similar documents (see the results for English and

Dutch subsets in Table VI). The prediction performance for the full dataset is, with an average error of 6.1% (for BagOfWords), sufficient for quality assessment on a general level. It should be noted that this error is lower than that of previously reported methods in the literature, although it is difficult to make an absolute comparison. Figure 3 shows the distribution of error values for all document pages. The negative values on the left denote under-prediction and the positive values over-prediction. Overall, for the majority of documents, the prediction error falls within the acceptable limits mentioned above. As one of the next steps of the authors' planned work, a close examination of the cases where prediction errors are either very low or very high (under or over) will potentially reveal ways of further improving the accuracy of the system.

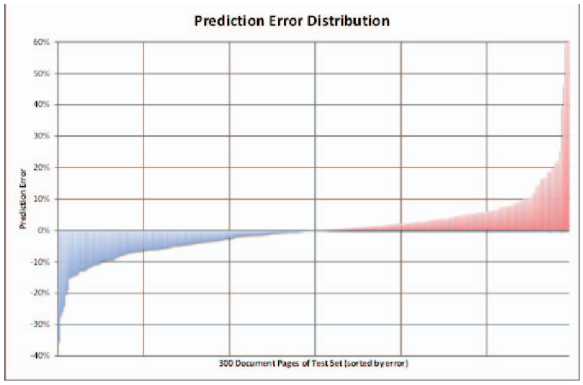

Fig. 3. Error distribution for the 300 documents of the test set (BagOfWords prediction using support vector machine).

Predicting the layout analysis performance seems to be a much harder problem than predicting text recognition results. The best average error that could be achieved for the full dataset is 11.3%. There is further scope for further research to identify and extract better features that may improve the quality estimation considerably.

Additional experiments were carried out using the ABBYY FineReader Engine for both feature extraction (instead of Tesseract) and as the prediction target. This use scenario is limited by the factor that the feature extraction tools require PAGE XML as input. The OCR results therefore have to be either exported directly in this format (e.g. by using the authors' FineReader Integration tool) or existing results (different format) have to be converted, which might involve loss of useful information.

While it could be expected that using the same OCR engine for both feature extraction and as the prediction target might result in a much more precise prediction, the outcome of the experiments only shows a minimal improvement compared to using Tesseract (as integrated within the presented system). The "Bag of Words" prediction error, for instance, is reduced by only 0.03% (from 6.12% using Tesseract to 6.09% using FineReader).

IV. Concluding Remarks

We have described a comprehensive system for quality estimation and thus prediction applicable to different OCR pipelines, based on open source software and tools that were specifically developed for this purpose. Predicting the outcome of digitisation pipelines, based on machine learning techniques, can complement pilot projects or can be used for pre-selection of data (triage) for different processing routes.

Experiments carried out on the Europeana Newspapers Project (ENP) dataset showed both the potential as well as the limitations of quality estimation in general. As expected, given a sufficiently large training set and a test set without too much variation, the quality prediction can be more accurate. On the other hand, small training sets and strong variation may lead to less reliable results. Examples for those two cases are the subset of English documents, which lead to a reasonably good prediction accuracy of 97.3% (for Bag of Words), and the full dataset of 600 pages which delivered a lower (but still acceptable) 93.9% prediction accuracy.

A large number of features (based on image, page layout, and text content) has been proposed and feature extraction tools were implemented. Perhaps surprising was, as shown by the experiments, that even the most basic features (such as region count) can be very useful for the prediction. More features can be added in the future as the software tools were specifically designed to accommodate this. In addition, current features could be enhanced, for instance by adding dictionaries for more languages ("words in dictionary" feature) or extending existing dictionaries.

References

[1] Europeana Newspapers Project (ENP): http://www.europeana-newspapers.eu/

[2] L.R. Blando, J. Kanai and T.A. Nartker, "Prediction of OCR Accuracy Using Simple Image Features", *Proc. ICDAR'95*, Montreal, Canada, August 14-16, 1995.

[3] P. Ye and D. Doermann, "Learning Features for Predicting OCR Accuracy", *Proc. 21st Int. Conf. on Pattern Recognition (ICPR2012)*, Tsukuba, Japan, November 11-15, 2012.

[4] A. Ben Salah, J.P. Moreux, N. Ragot and T. Paquet, "OCR Performance Prediction Using Cross-OCR Alignment", *Proc. 13th Int. Conf. on Document Analysis and Recognition (ICDAR2015)*, Nancy, France, August 2015, pp. 556-560.

[5] Tesseract OCR: https://github.com/tesseract-ocr

[6] Weka 3: Data Mining Software in Java: http://www.cs.waikato.ac.nz/ml/weka/

[7] David J.C. Mackay (1998). Introduction to Gaussian Processes. Dept. of Physics, Cambridge University, UK.

[8] S.K. Shevade, S.S. Keerthi, C. Bhattacharyya, K.R.K. Murthy: Improvements to the SMO Algorithm for SVM Regression. In: IEEE Transactions on Neural Networks, 1999.

[9] D. Aha, D. Kibler (1991). Instance-based learning algorithms. Machine Learning. 6:37-66.

[10] S. Pletschacher, A. Antonacopoulos, "The PAGE (Page Analysis and Ground-Truth Elements) Format Framework", Proc. 20th Int. Conf. on Pattern Recognition (ICPR2010), Istanbul, Turkey, August 23-26, 2010, IEEE-CS Press, pp. 257-260.

[11] C. Clausner, C. Papadopoulos, S. Pletschacher, A. Antonacopoulos, "The ENP Image and Ground Truth Dataset of Historical Newspapers", Proc. the 13th Int. Conf. on Document Analysis and Recognition (ICDAR2015), Nancy, France, August 2015, pp. 931-935.

[12] B. Gatos, I. Pratikakis, and S. J. Perantonis. 2006. Adaptive degraded document image binarization. Pattern Recognition, 39, 3 (March 2006), pp. 317-327.

[13] S. Pletschacher, C. Clausner and A. Antonacopoulos, "Europeana Newspapers OCR Workflow Evaluation", Proc. 3rd Int. Workshop on Historical Document Imaging and Processing (HIP2015), Nancy, France, August 2015, pp. 39-46.