Multi-modal page stream segmentation with convolutional neural networks

Article *in* Language Resources and Evaluation · September 2019

DOI: 10.1007/s10579-019-09476-2

2 authors:

Gregor Wiedemann
University of Hamburg
62 PUBLICATIONS **243** CITATIONS
SEE PROFILE

Gerhard Heyer
University of Leipzig
151 PUBLICATIONS **795** CITATIONS
SEE PROFILE

Noname manuscript No.
(will be inserted by the editor)

Multi-modal Page Stream Segmentation with Convolutional Neural Networks

Gregor Wiedemann* · Gerhard Heyer

Received: date / Accepted: date

Abstract In recent years, (retro-)digitizing paper-based files became a major undertaking for private and public archives as well as an important task in electronic mailroom applications. As first steps, the workflow usually involves batch scanning and optical character recognition (OCR) of documents. In the case of multi-page documents, the preservation of document contexts is a major requirement. To facilitate workflows involving very large amounts of paper scans, page stream segmentation (PSS) is the task to automatically separate a stream of scanned images into coherent multi-page documents. In a digitization project together with a German federal archive, we developed a novel approach for PSS based on convolutional neural networks (CNN). As a first project, we combine visual information from scanned images with semantic information from OCR-ed texts for this task. The multi-modal combination of features in a single classification architecture allows for major improvements towards optimal document separation. Further to multimodality, our PSS approach profits from transfer-learning and sequential page modeling. We achieve accuracy up to 95 % on multi-page documents on our in-house dataset and up to 93 % on a publicly available dataset.

Keywords Page stream segmentation · Document flow segmentation · Convolutional neural nets · Text classification · Digital mailroom

Dr.-Ing Gregor Wiedemann
Hamburg University, Germany
Department of Computer Science
Vogt-Kölln-Str. 30
22527 Hamburg
E-mail: gwiedemann@informatik.uni-hamburg.de*

Prof. Dr. Gerhard Heyer
Leipzig University, Germany
Department of Computer Science
Augustusplatz 9
04109 Leipzig
E-mail: heyer@informatik.uni-leipzig.de

1 Introduction

For digitization of incoming mails in business contexts as well as for retro-digitizing files from paper archives, consecutive pages can be separated manually into documents before scanning. However, to save costs, batch scanning of pages in a row without any manual separation is an alternative option. Such a batch scanning of multiple documents can be a major simplification of the processing workflow. In this scenario, scanned images of multi-page documents arrive at a document management system as an ordered stream of single pages but without information on document boundaries. Page stream segmentation (PSS) then is the task of separating the continuous document stream into sequences of pages that represent single physical documents.[1]

Applying a fully automated approach of document stream segmentation can be favorable over manually separating and scanning documents, especially in contexts of very large data sets (Gallo et al., 2016). In a joint research project together with a German research archive, we supported the task of retro-digitization of a paper archive consisting of circa one million pages put on file between 1922 and 2010 (Isemann et al., 2014). The collection contains documents of varying content, types and lengths around the topic of ultimate disposal of nuclear waste. Among others, it contains administrative letter correspondence and geological research reports, but also stock lists, protocols, meeting minutes and email printouts. The one million pages have been archived in roughly 20,000 binders which were batch-scanned in an automated process due to limited manual capacities for separation of individual documents. Especially the long time range of archived material poses severe challenges for any automatic document processing. In the data, among other things, we observe evolvement of different contents, a large variety of document quality ranging from hand-written letters, over type-writer documents to printouts with all kinds of printers. Further, the data contains manifold document types with varying layout standards, different fonts and use of tables, figures, and hand-written notes. This high variance in the data affects optical character recognition (OCR) as well as PSS based on its results.

Our contribution: To address these challenges effectively, we introduce our approach to PSS based on convolutional neural networks (CNN). For the first time for this task, we combine textual and visual features into one network to achieve most-accurate results up to 95% on multi-page documents. Section 2 elaborates on related work for our task. In section 3 we describe our dataset together with one reference dataset for this task. In section 4 we introduce our neural network-based architecture for PSS. As a baseline, we introduce a support vector machine-based model (SVM) solely operating on text features. Then, we introduce CNN for PSS on text and image data separately as well as a multi-model approach combining both feature types. Section 5 presents a

[1] The task is also referred to as Document Flow Segmentation, Document Stream Segmentation, or Document Separation.

Table 1 Recent works on page stream segmentation and document image classification

Publication	PSS	DIC	Image	Text	Classifier
Rusiñol et al. (2014)		X	X	X	SVM
Daher and Belaïd (2014)	X			X	SVM, MLP
Daher et al. (2014)	X			X	KNN
Agin et al. (2015)	X		X		SVM, RF, MLP
Harley et al. (2015)		X	X		CNN
Noce et al. (2016)		X	X	X	CNN
Gallo et al. (2016)	(X)	X	X		CNN+MLP
Karpinski and Belaïd (2016)	X			X (layout)	rule-based
Hamdi et al. (2017)	X			X (layout)	rule-based, doc2vec
Hamdi et al. (2018)	X			X (layout)	DT
Our approach	X		X	X	SVM CNN+MLP

quantitative evaluation and a qualitative discussion of the results on the two datasets.

2 Related work

Page stream segmentation is related to a series of other tasks concerned with digital document management workflows. Table 1 summarizes the important characteristics of recent works in this field. A common task related to PSS is document image classification (DIC). For DIC, typically visual features (pixels) are utilized as input to classify scanned document representations into categories such as "invoice", "letter", "certificate" etc. Category systems can become quite large and complex. Moreover, single-page versus multi-page tasks can be distinguished. Gordo et al. (2013) summarize different approaches in a survey article on both PSS and DIC.

In general, two types of PSS can be distinguished: rule-based vs. machine learning-based approaches. Rule-based systems (RBS) rely on hand-crafted features from so-called 'descriptors' to determine whether a page belongs to the sequence of predecessor pages or represents the beginning of a new document. Descriptors can be greeting phrases, reoccurring named entities or customer IDs on pages, as well as (disrupted) sequences of page numbers in specific locations on the page. They are usually extracted with the help of regular expressions from OCR-ed text in combination with additional layout features. Layout features may comprise, for instance, absolute text box positions on a page, and formatting information such as font family or font size as delivered by some OCR-systems. Among others, rule-based PSS is investigated by Meilender and Belaïd (2009) and Karpinski and Belaïd (2016) who successfully applied it to a collection of homogeneously structured documents and forms. They combine rule matching with different sequence models and correction modules to determine an optimal separation of the continuous page flow.

The engineering of descriptors from text and layout position information can be a tedious task. Rule-based systems do not generalize well to heterogeneous datasets containing documents of different types and lengths from a long time range. To address this problem, machine learning (ML) based approaches to PSS became more and more popular. Hamdi et al. (2017) compare ML-based PSS against RBS and find that the former improves the performance especially on multi-page documents. As a kind of mixed type of PSS, Daher and Belaïd (2014), Daher et al. (2014) and Hamdi et al. (2018) successfully combine RBS with off-the-shelf classifiers such as SVM, decision tree (DT), and random forest (RF) to improve the performance of their systems.

RBS often treat PSS as a sequence optimization problem on the rather small set of manually engineered layout and text descriptors. To determine document boundaries, they are looking at features of a number of pages in a row. In contrast, purely ML-based approaches model PSS as a binary classification task and further do not rely on manual engineering of features only. An ML-approach solely based on image information is proposed in Agin et al. (2015) where each page scan of a sequence is classified as either continuity of the same document, or rupture, i.e. the beginning of a new document. For this binary classification, they employ 'bag of visual words' (BoVW) from document images together with font information obtained from the OCR-system as features, and test performance with three binary classifiers (SVM, Random Forest, and multi-layer perceptron). In contrast, Hamdi et al. (2017) derive features for PSS solely from text using 'doc2vec', a neural network-based embedding model introduced by Le and Mikolov (2014), which encodes document semantics in a fixed-length vector. They then compare vector representations of neighboring pages with cosine similarity and classify into continuity or ruptures based on a similarity threshold. While their system successfully separates of multi-page documents, it has some trouble identifying single-page documents correctly. This is not surprising since semantically similar (single page) documents can appear quite often one after the other in some datasets.

As for many other applications, recent developments in deep learning led to major improvements also for DIC and PSS. For DIC, the recent state of the art is achieved by Gallo et al. (2016), Harley et al. (2015) and Noce et al. (2016) who employ deep learning with CNNs in combination with transfer learning to identify document classes. Gallo et al. (2016) perform PSS on top of the results from a similar DIC process. Page scans from the stream are segmented each time the DIC system detects a change of the document class label between consecutive page images. Unfortunately, this approach can only be successful if there are alternating types of documents in the sequential stream. Often, this cannot be guaranteed, especially in the case of small document category systems. Since we only have 17 document categories and a majority of them belong to one category ("letter"), we need to perform direct separation of the page stream by classifying each page into either continuity or rupture. Second, the quality and layout of our data are extremely heterogeneous due to the long-time period of document creation. We expect a lowered performance by solely relying on either visual or textual features for separation.

Potentially, ML-based PSS may learn discriminating features from both, visual and textual information. To our knowledge, such a multi-modal classification so far was only applied to the task of DIC by Rusiñol et al. (2014) and Noce et al. (2016). Rusiñol et al. (2014) use TF-IDF and LSA features Deerwester et al. (1990) from text data, as well as pixel density descriptors from image data in two separate classification modules and then combine their predictions to improve the final results. Noce et al. (2016) highlight class-specific key-terms obtained from OCR-ed texts in the image representation of the page with colored boxes, and then apply a single image classifier. For PSS, no multi-modal approach has been introduced so far.

For our approach, we take the previous works by Rusiñol et al. (2014), Gallo et al. (2016) and Noce et al. (2016) as a starting point. We model PSS as a binary classification task combining textual features and visual features using deep neural networks. This architecture is compared against a baseline comprising an SVM classifier solely relying on textual features.[2]

3 Datasets

We evaluate our approach for PSS on two datasets. The first is an in-house dataset sampled from the digitized German archive of our project context. Unfortunately, all previous studies on PSS developed and evaluated their approaches on in-house datasets as well. Hence, there is no direct comparison to their performance possible. To allow a comparison of approaches to some extent in the future, we will include a second, public resource in our study. This resource is a dataset of annotated document scans from U.S. tobacco companies.

3.1 German archive data

The German dataset consists of a variety of document classes from a very long time period. Most short documents represent letter correspondences between several administrative institutions and private companies, stock and inventory lists, and meeting minutes. Longer documents usually represent expert's reports and scientific studies. Most of the documents were archived between the mid-1960s and 2010. Due to this, OCR-quality, document lengths, layout standards as well as used fonts differ widely.

After batch scanning, about 40 % of all binders from the German research archive have been manually separated into documents and annotated with

[2] Actually, it would be preferred to compare our system against other approaches from the scientific literature directly. Unfortunately, we neither encountered a ready-to-use implementation of any text-based PSS system nor a common, public-domain dataset for the task. Our archive dataset is of such heterogeneity (see Section 1) that we refrained from manual engineering of descriptor features. Instead, we opted for a strong baseline of a machine learning-based system which does not require extensive feature engineering and has successfully been used in related works (see Table 1).

Table 2 Distribution of document lengths and PSS page classes

Document length	**Archive26k**	**Tabacco800**
1	2745	465
2	831	179
3	319	39
4	201	15
5	141	10
6	110	4
7	55	6
8	50	2
9	38	11
10 and more pages	390	5
Single-page	2745 (56%)	465 (63%)
Multi-page	2135 (44%)	271 (37%)
# new document (rupture)	4880 (18%)	736 (57%)
# same document	22007 (82%)	554 (43%)
Total documents	4880	736
Total pages	26887	1290

document categories. The manually separated documents can serve as ground truth for our experiments on model selection and feature engineering for automatic page stream segmentation. For these experiments, we randomly selected 120 binders from the set of all manually separated binders. The binders represent 120 ordered streams of scanned pages, in total consisting of 26,887 pages. Table 2 shows a distribution of document lengths together with basic statistics of the number of pages per class. Similar to other events in language data (e.g. word frequencies), document length follows roughly a power-law distribution with emphasis on shorter documents. Although the majority of documents in the archive data are single-page documents (56%), the rather small share of very long multi-page documents leads to a very imbalanced distribution of our page classes (new vs. same document) in the dataset. 80 of the selected binders containing 17,376 pages were taken as a training set, 20 binders with 5095 pages were taken as a validation set, and the remaining 20 binders with 4416 pages as a final test set.

Scanned pages were resized to 224 $\times$ 224 pixels[3] and color-converted to black and white with the Otsu's binarization method (Otsu, 1979). Binarization in combination with downsampling the image resolution reduces information to speed up learning while highlighting valuable layout features for classification. Figures 1 and 2 show in their respective upper rows examples of first pages and subsequent pages from archive documents.

From original document scans, text information was extracted by optical character recognition (OCR) to obtain textual features in addition to scanned

[3] In previous experiments, we showed that higher image resolutions lead to better results in PSS. At the same time, performance slows down drastically. We choose the final image size as the largest input image size for a pre-trained VGG16 network from our transfer learning setup (see Section 4).

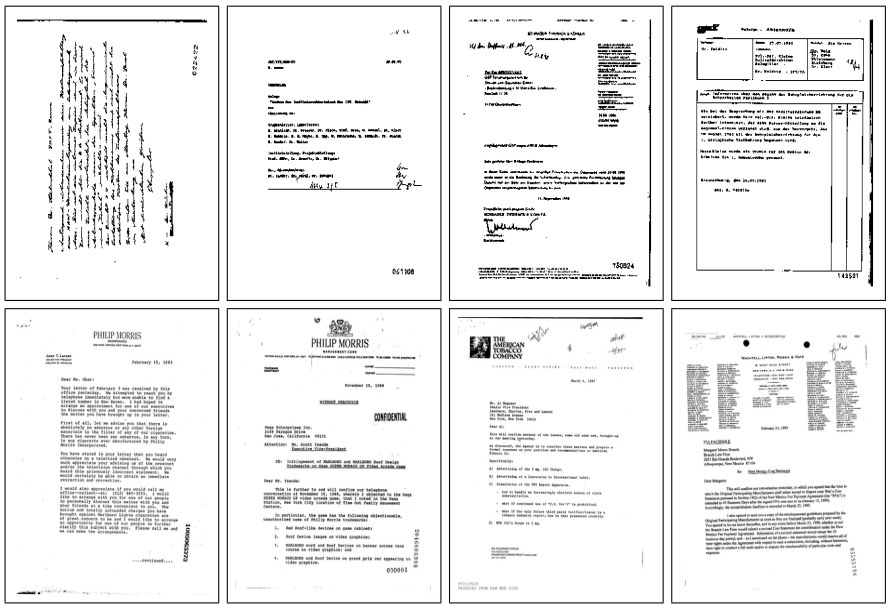


Fig. 1 Examples for first pages (class *new document*); from Archive26k (above) and Tobacco800 (below).

page images. On average, one page contains about 168 word tokens. In the following, this dataset is referred to as *Archive26k*.

3.2 U.S. Tobacco company correspondences

As a second evaluation set, we run our process on the *Tobacco800* document image database (Lewis et al., 2006). To our best knowledge, the dataset is the only publicly available real-world resource of document images used in the context of DIC containing multi-page documents. In contrast to our in-house dataset, evaluation of PSS on this dataset will allow comparing the performance of our approach to future studies.

The Tobacco800 dataset is a small annotated subset of the Truth Tobacco Industry Documents, a collection of more than 14 million documents originating from seven major U.S. tobacco industry organizations dealing with their research, manufacturing, and marketing during the last decades. The documents had to be publicly released due to lawsuits in the United States.

The annotated subset for our experiments is composed of 1,290 document images sampled from the original corpus. Similar to the Archive26k dataset, it contains multi-page documents of different types (e.g. letters, invoices, handwritten documents) and thus is well suited for evaluation of our task. Samples from the Tobacco dataset were also used in Harley et al. (2015) and Noce et al.

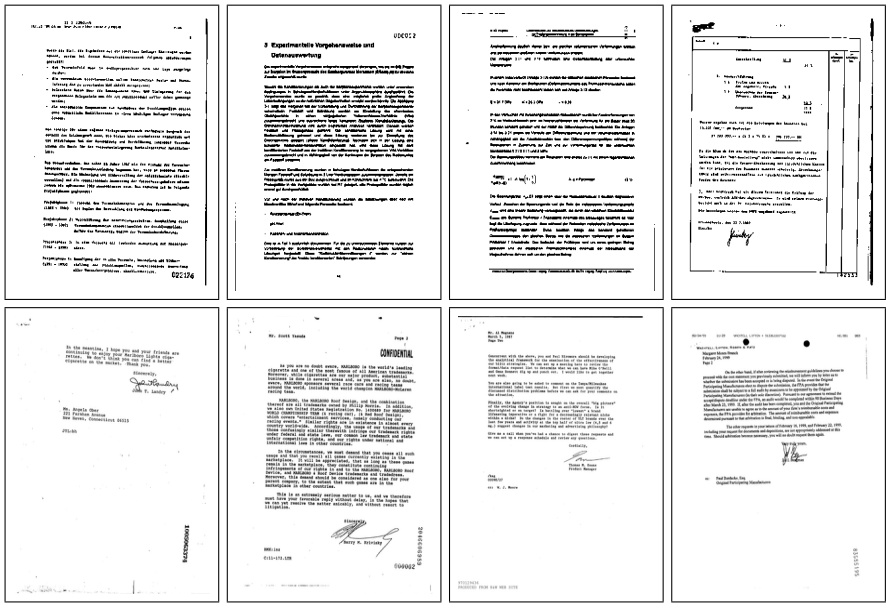


Fig. 2 Examples for subsequent pages (class *same document*); from Archive26k (above) and Tobacco800 (below).

(2016).[4] We split the dataset into training and test set according to the order of their alphabetically sorted filenames. The resulting training set contains 1031 images (ca. 80%) from 586 documents, the remaining test set contains 259 images (ca. 20%) from 150 documents. Again, we OTSU-binarize page scans to a black/white color palette and resize them to a 224 × 224 pixel resolution. The lower lines in Fig. 1 and 2 show examples of first pages, resp. subsequent pages from Tobacco800 documents. For text features, we apply OCR on each page resulting in an average page length of about 249 word tokens. Table 2 shows a distribution of document lengths in the dataset similar to those in the Archive26k dataset.

As the example pages show, both collections share similarities in their visual appearance. First pages compared to subsequent ones may contain distinct header elements. But in general, the human observer has difficulties to identify clear layout patterns discriminating between both classes, especially for the Archive26k documents. Therefore, visual features alone may not be sufficient for accurate PSS.

Regarding their textual content, the two datasets share certain similarities but also differ with respect to language, size, and creatorship. Both have in common that they cover long time periods and are thematically located

[4] They utilize the Tobacco 3482 dataset consisting of pages manually tagged with 10 different document categories (Kumar et al., 2014). The dataset is widely used in DIC research. Since it does not contain multi-page documents, it is not suitable for PSS.

within a rather narrow domain (nuclear waste disposal, tobacco industry). Nonetheless, they largely differ with respect to the characteristics of content creators. On the one hand, there is a state-run research library archiving material from a wide variety of institutions, while on the other hand there are internal documents from a rather small set of business actors with corporate design standards. Due to this, we expect different performance from textual and visual features for PSS on both datasets.

4 Binary classification for PSS

We approach PSS as a binary classification task on single pages from a continuous data stream. Pages are classified into either continuity of the *same document* (SD) or rupture, i.e. beginning of a *new document* (ND). For classification, we compare two architectures: SVM with specifically engineered text features (4.1) and a combination of convolutional neural nets with both, textual and visual features (4.2).

4.1 Baseline: Linear text classification

As a baseline, we use text classification together with specifically engineered features for PSS. For this first step, we rely on SVM with a linear kernel.[5] This learning algorithm has proven to be very efficient for binary classification problems with sparse and large feature spaces (Joachims, 1998), is computationally much faster than neural network architectures, and has been successfully applied for PSS before (Rusiñol et al., 2014; Daher and Belaïd, 2014).[6] We set class weights to account for the high imbalance between the two classes in our dataset,[7] and optimize the C-parameter for each SVM model on the validation set. We extract the following four types of features from the OCR-ed text data of each of the single pages:

N-grams: Page texts are tokenized at boundaries of character class changes, and the resulting tokens are converted to lowercase. We further delete punctuation marks and replace digits in tokens with a #-character. From the resulting word token sequences, uni-, bi-, and trigram features are created. Bi- and trigram features allow representing page text content preserving sequential text information to a certain extent. Especially due to OCR errors, the

[5] We use the Liblinear library by Fan et al. (2008)

[6] We refrain from using image features in this architecture because pixel features are not supposed to be linearly separable. First experiments confirmed that simple pixel features do not contribute discriminative information on top of text features to the linear SVM for our task. Of course, we could use a different SVM kernel for image classification. But, very likely we would lose the advantage of computational speed. Due to this, we stick to text features for our baseline method.

[7] We utilize inverse probability weighting on the training data to put more weight on the minority class (new document) during loss calculation.

data contains a lot of noise. Thus, relative feature pruning is applied to receive manageable vocabulary sizes and reduce noise from infrequent events in the data. From the n-gram feature set, all features were pruned which occur in less than 0.1% of all Archive26k training documents, or less than 0.2% of all Tobacco800 training documents. This step resulted in 32,002 (Archive26k), resp. 29,832 (Tobacco800) features encoding raw frequency counts of n-grams on each page.

Topic composition: In a second step, we obtain features of topical composition for each page from an unsupervised machine learning process. For this, we rely on Latent Dirichlet Allocation (LDA), also referred to as topic modeling (Blei et al., 2003).[8] From a topic model

$$P(\mathbf{W}, \mathbf{Z}, \theta, \varphi; \alpha, \beta) = \prod_{i=1}^{K} P(\varphi_i; \beta) \prod_{j=1}^{M} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t} \mid \theta_j) P(W_{j,t} \mid \varphi_{Z_{j,t}})$$

with K topics, M pages, and N word types in the vocabulary, topic proportions based on multinomial posterior probability distributions θ can be used as a dense feature vector comprising latent semantics of the modeled documents. In addition to the highly sparse n-gram feature set, they can provide useful information to any text classifier. For PSS, we expect that topics may encode latent information about beginnings and endings of documents. Following a method proposed by Phan et al. (2011), we presented single page texts as pseudo-documents to the process and compute a model with $K = 50$ (Tobacco800), resp. $K = 100$ (Archive26k) topics. Different topic resolutions were chosen with respect to different collection sizes. For each page p, we then use the resulting topic-page distribution θ_p as feature vector supplementary to the previously extracted vector of n-gram counts.

Topic difference: We expect multi-page documents to comprise a rather coherent topic structure. For PSS, a rupture in topic coherence between pages may indicate the beginning of a new document. Thus, for each page p, we determine the difference between its topic composition θ_p and its predecessor θ_{p-1} as a third feature type. We utilize two measures, Hellinger distance and Cosine distance, to create two additional features. While the former is a common metric to compare two probability distributions, the latter also has been adopted successfully to compare topic model results (Niekler and Jähnichen, 2012). Distance values near zero indicate a high similarity of topic composition compared to the predecessor page. Values near one indicate a significant change of topic composition.

[8] Actually, there is a large variety of unsupervised topic models as well as many other methods to reduce sparse, high-dimensional text data to a dense, lower-dimensional space (e.g. latent semantic analysis, Deerwester et al., 1990). For our baseline system, we stick to LDA as the seminal and most widely-used topic model.

Predecessor pages: As a last feature type, we append a copy of features X_{p-1} extracted in the previous three steps belonging to the predecessor page as additional features to each current page X_p. For this, we concatenate existing feature identifiers from the predecessor page with a common prefix, e.g. 'PREV#' such that the classifier is able to distinguish between feature values for the current page and copied values from the predecessor page. The rationale behind is to allow the classifier not only to learn from information about characteristics from one page but to look at a sequence of pages for its decision. For instance, the presence of a salutation phrase such as "With kind regards" on a predecessor page highly increases the probability for the beginning of a new document on the current page.

The performance of *SVM classification* to determine for each page whether it is the beginning of a new document or the continuation of the current document is tested in consecutive steps. In each step, one of the four just introduced feature types is added to the feature set. The stepwise expansion of the feature types to the linear SVM allows controlling whether each type effectively provides valuable information for the process.

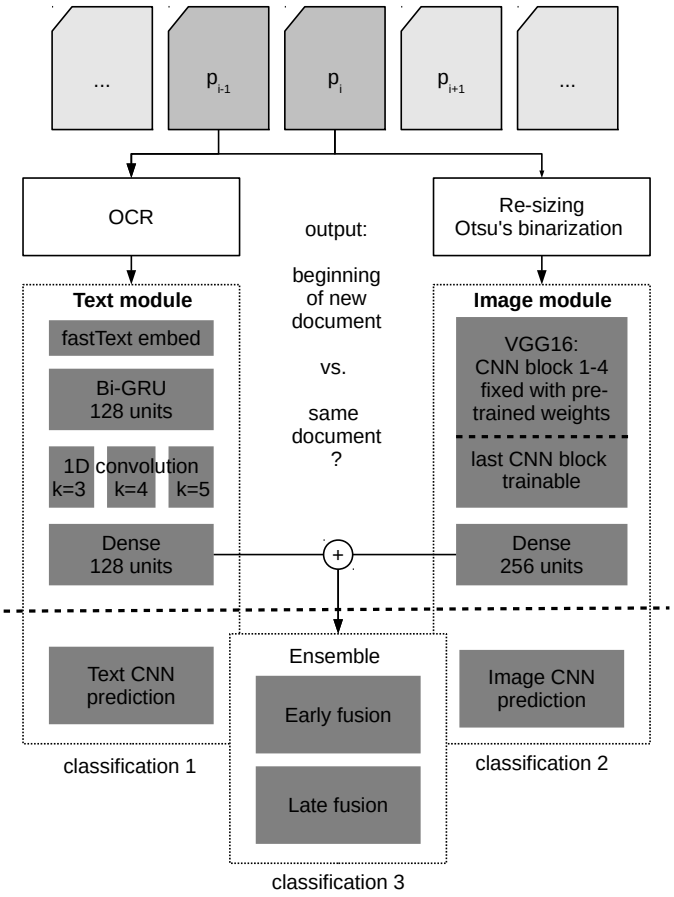
4.2 Neural networks on text and image features

For our new PSS approach (cp. Fig. 3 for a schematic representation of the architecture), we first create two separate convolutional neural networks (CNN) for binary classification of pages into either SD or ND, one based on text data and another based on image scans. In a third step, we combine the two classifiers in an ensemble to achieve an improved classification result from the two modalities. For an optimal combination of modalities, different strategies are tested.

CNN for text data: We start with an effective, widely-used CNN-architecture for text classification originally introduced by Kim (2014) which achieved high performance for sentiment analysis tasks on standard data sets. He uses 1-dimensional convolution over word sequences which are encoded as embedding vectors. To allow for the encoding of semantics from word n-grams of varying size, three convolutional layers with kernel sizes $k \in \{3, 4, 5\}$ are applied in a parallel manner, followed by a fully-connected dense layer with 128 units as a final feature layer for the classification.

For embedding word semantics, we use a publicly available fastText model pre-trained on Wikipedia (Bojanowski et al., 2017). FastText embeddings are of particular value for our PSS task since they are computed based on subword information, i.e. convolution over character n-grams. Subword information allows for the computation of embeddings even for words which have not been seen during training of the embedding model. This drastically reduces negative impact for any classification task on account of out-of-vocabulary words (OOV). For PSS, we actually have an extraordinarily high OOV rate, not

Fig. 3 CNN + MLP architecture for PSS



only due to the very domain-specific vocabulary in our datasets but due to OCR errors. Instead of representing all OOV terms with the same 'unknown' embedding, fastText subword embeddings give us actually meaningful word vectors close to the correct spelling variant of misspelled words.

As for many other text classification tasks, important information stems not only from the used words themselves but from their syntactical order. Syntactical information can be approximated by looking at sequence order in addition to (bag of) words alone. Comparable to n-gram features in the SVM approach, convolution over sequences of k word tokens allows the model to learn from word sequence order to some extent. However, for PSS not only smaller local contexts of words contain useful information. We also suspect that a classifier may benefit to learn from whether a token sequence occurred rather at the beginning or at the end of a page. In neural network-based ma-

chine learning, recurrent layers such as gated recurrent units (GRU) (Cho et al., 2014) are a common approach to model such sequential data. Applying a recurrent layer over word embeddings of the document context allows the model to learn contextually embedded word representations. Feeding such representations into a convolutional classification architecture has been shown to be beneficial compared to direct use of uncontextualized word embeddings in other text classification tasks (Wiedemann et al., 2018). To take further advantage of sequential information beyond a local context of k words for PSS, we extend Kim's model by adding a bidirectional GRU-layer with 128 cells before convolution. We further restrict the maximum sequence length to $m = 150$ tokens. For longer page content, we select $m/2$ tokens from both the beginning and the end to represent the most useful sequence information to the classifier. For shorter page sequences, we apply zero-padding. The sequential output of the GRU-layer is then fed into the convolutional layers.

Each of the three parallel convolutional layers consists of 200 filters, followed each by a global max-pooling layer and a dropout layer (dropout rate 0.5). Their three outputs are then concatenated into one vector followed by a dense layer of 128 neurons with 'Leaky ReLU'-activation (this can be seen as a final feature layer), and a prediction layer for the binary classifier decision (sigmoid activation). Training of this network is performed using binary cross-entropy loss and Nesterov Adam optimization with a learning rate of 0.01 and mini-batches of size 32.

CNN for image data: Following the works in Noce et al. (2016) and Gallo et al. (2016), we use a very deep CNN architecture to classify scanned pages based on their binarized and resized representation as 224×224 pixels. As a basis, we use the VGG16 model architecture introduced by Simonyan and Zisserman (2014) for object recognition tasks. This architecture contains 13 convolution layers grouped into five sequentially chained blocks. Each convolution block is completed with a global max-pooling layer. A final block contains three sequential dense layers for the prediction of objects in images. We crop the final dense layers from the original VGG16 architecture and add two new ones, one with 256 units and 'Leaky RELU'-activation as a final feature layer, and a binary prediction layer with softmax activation. Between the last CNN block and the first top dense layer, we apply dropout regularization with rate 0.5.

As a variant of transfer learning, we further use pre-trained layer weights based on the 'imagenet' dataset for the five CNN blocks. Actually, 'imagenet' contains manually labeled photographs for object recognition tasks. However, earlier work has shown that CNN weights pre-trained on imagenet, although not specifically intended for the task of document image classification, can significantly improve DIC results for small datasets, too (Harley et al., 2015). Thus, we expect them to be beneficial for our PSS task as well. To allow the network to adapt to our specific data and classification task, we apply a common technique of fine-tuning. For this, we freeze all pre-trained layer weights of the first four CNN blocks. Only the fifths CNN block and the two top dense layers are kept trainable. Learning for this network is performed

using Adam optimization with Nesterov momentum and a very small learning rate ($lr = 0.00005$) and mini-batches of size 32.

Page sequence information: As for the SVM baseline, we test the effect of including predecessor page information on PSS performance for our two CNN architectures. Instead of classifying only single pages, we model short two-page sequences by adding a second neural classifier module (base model except for the final dense prediction layer), either the text- or image-based one, with identical architecture and feeding two inputs (predecessor and current target page) to them. The two outputs of the final feature layer of each module are then concatenated again into one vector which is fed into a new dense layer with 128 units (text-based CNN), resp. 256 units (image-based CNN).

4.3 Combining text and visual features

Each of the two previously introduced CNN are capable of classifying pages into either SD or ND on their own. However, since highly distinct information is utilized in each approach, we expect a performance gain from combining textual and visual information. Theoretically, a single model combining both feature types can be trained where image information and text information is processed in separate branches of the model architecture first, and the information from both is combined in a new fully-connected layer. In practice, however, training such a model becomes impractical due to different complexities and learning rates of the two models. The image module applies some form of transfer learning and fine-tuning of pre-trained parameters. Fine-tuning of neural networks is actually known to suffer from 'catastrophic forgetting' of previously learned information when presented with the new target data. Hence, we need to apply very small learning rates to mitigate this effect. The text module, in contrast, is trained from scratch with randomly initialized values. Here, the optimizer can make use of large learning rates to converge to an optimum efficiently. A combined model with its large number of parameters then suffers either from catastrophic forgetting of pre-trained parameters of the image module or from slow convergence of parameters of the text module during optimization. Due to this circumstance, we have not been successful in training a single combined classifier on both modalities.

Instead, the more successful strategy is to combine the outcomes of the separate classifiers in some kind of ensemble approach. As successfully tested for DIC before (Rusiñol et al., 2014), this can be achieved in two different ways: early and late fusion.

Early fusion: In this ensemble strategy, the output of an intermediate step before the final prediction from each neural network model is used as a feature input for a third classifier. First, a model for each modality is trained independently. Then, the final prediction layer from each model is removed. In a third step, training and test data can be fed into the networks again to receive

prediction values from the last fully-connected layers of the pruned networks. The output values from these last layers can be interpreted as new feature vectors for each data instance which encode dimensionality-reduced information of the respective modality. We generate such feature vectors for both text and image data for all instances of the training, test and validation sets. In early fusion, these feature vectors can now be used as input for any classifier. We concatenate the feature vectors of the different modalities and feed them into a simple multi-layer perceptron (MLP) with one 400-dimensional hidden layer, dropout regularization ($dr = 0.9$) and a fully-connected final prediction layer with sigmoid activation.[9] Training again is performed with Nesterov Adam optimization ($lr = 0.002$) and binary cross-entropy loss.

Late fusion: In this ensemble strategy, the final label probability output from the prediction layer of each model is used either as input for a third classifier (ensemble stacking) or for a (weighted) mean (ensemble averaging). We opt for the latter following an approach introduced by Rusiñol et al. (2014).[10] Instead of simple averaging of the single classifier probabilities, they calculate a power-weighted product of probability vectors P_t from text-based and P_v from visual classification:[11]

$$P_{vt} = P_v^i \times P_t^j \text{ with } i, j \in [0, 1]$$

In addition to text and image prediction, we evaluate whether LDA features already introduced for the SVM baseline may further improve PSS with CNN architectures. For early fusion, we concatenate the K inferred topic proportion features and the two topic distance features from our baseline approach to the combined image and text feature vector. For late fusion, we train a simple MLP model (400-dimensional hidden layer, 0.9 dropout) based on the topic proportion and distance features and add its prediction probability as a third weighted term P_l^k ($k \in [0, 1]$) to the weighting equation above.

5 Evaluation

We report results from a quantitative evaluation of our two datasets as well as from a qualitative look into error patterns of the final models.

[9] The hyperparameters of hidden layer units and dropout rate have been obtained by hyper-parameter tuning w.r.t. the validation set.

[10] We also tested ensemble stacking with a logistic regression classifier but did not achieve better performance.

[11] As for the MLP in early fusion, optimal weighting values for i, j, and k have been obtained via optimization w.r.t. the validation set.

Table 3 Model selection for page stream segmentation (validation set performance)

Approach/dataset	**Archive26k**		**Tobacco800**	
	Accuracy	Kappa	Accuracy	Kappa
Majority baseline	0.856	0.0	0.421	0.0
SVM n-grams	0.860	0.477	0.833	0.649
+ topic composition	0.861	0.480	0.833	0.649
+ topic difference	0.867	0.477	0.841	0.666
+ predecessor page	0.869	0.488	0.814	0.610
CNN Text	0.908	0.620	0.850	0.690
+ predecessor page	0.909	0.629	0.851	0.693
CNN Image	0.896	0.561	0.865	0.720
+ predecessor page	0.900	0.580	0.886	0.759
Image + text				
Early fusion	0.924	0.680	0.905	0.805
Late fusion	0.929	0.697	0.915	0.824
Image + text + topic				
Early fusion	0.923	0.679	0.906	0.807
Late fusion	**0.934**	**0.708**	**0.931**	**0.855**

5.1 Quantitative evaluation

Table 3 displays the results of all tested model architectures and features types for PSS on the validation sets of our two investigated data sets.[12] Performance is measured by the accuracy of classifying a page either as new document beginning or continuity of the same document. Since the distribution of both classes is fairly uneven due to different document lengths (there are a lot more pages in the SD class), we additionally employ Kappa statistics to report a chance-corrected agreement between human and machine separations of page streams.

SVM baseline: At a first glance, in terms of accuracy SVM classification even with sophisticated feature engineering does not seem to clearly outperform the majority baseline on the Archive26k dataset. Of course, this is a misleading effect due to the uneven class distribution. Evaluation by Kappa statistics reveals that SVM is actually able to discriminate between document rupture and continuity ca. 49 % above chance level agreement for the complex German archive documents, resp. 67 % for the English tobacco industry dataset. Although features based on LDA topic composition have been used successfully in other text classification tasks (Wiedemann, 2019), they do not seem to improve the SVM results for PSS substantially. This can be explained intuitively as topics represent something like thematically coherent vocabulary. Structural

[12] The performance of neural network classification, in general, is not entirely deterministic due random initialization of layer weights and shuffling of mini-batches during training. To allow for a fair comparison of different CNN architectures, we repeated each of the experiments 10 times and report average results in Table 3.

Table 4 Archive26k test set performance (left) and confusion matrix (right)

Test set	**N**	**Accuracy**	**Kappa**
Archive26k	4416	0.889	0.591
Single-page doc.	490	0.422	-
Multi-page doc.	3926	0.947	0.682

	Pred. / All		**Multipage**	
Truth	SD	ND	SD	ND
SD	3465	92	3465	92
ND	398	461	115	254

Table 5 Tobacco800 test set performance (left) and confusion matrix (right)

Test set	**N**	**Accuracy**	**Kappa**
Tobacco800	259	0.919	0.831
Single-page doc.	98	0.980	-
Multi-page doc.	161	0.881	0.747

	Pred. / All		**Multipage**	
Truth	SD	ND	SD	ND
SD	93	16	93	16
ND	5	145	3	49

information such as 'first-page content' cannot be represented well by the LDA model, since first pages usually do not contain different thematic content than successor pages. Only when the topic difference between consecutive pages is taken into account, we observe a slight performance gain. This finding is consistent with Hamdi et al. (2017), who found that similarity/difference based on consecutive doc2vec page vectors is a strong feature. Considering short page sequences instead of isolated page classification by adding features from the predecessor page to the SVM, again slightly improves results for one dataset (Archive26k), but actually seems to harm the performance for the other (Tobacco800). We suspect that due to the much lower share of SD-class pages in this dataset (cp. Table 2), the classifier cannot profit from this feature.

Convolutional neural nets: Convolutional architectures on both feature types alone, text or image, already outperform the SVM baseline as well as the majority baseline in terms of accuracy. Comparing text versus image features, we receive a mixed picture: predictions based on the text are more accurate than for images for the Archive26k dataset, but the other way around for the Tobacco800 data. One potential explanation might be the rather small size of the latter dataset. Here, the use of transfer learning from the VGG16 model pre-trained on a very large dataset for object recognition probably is a considerable advantage compared to the text-based CNN module learning from scratch with only little training data. Further, adding predecessor page features to the CNN model architectures consistently beats classification of features from single pages only. However, performance increases are noticeable only for image-based CNN.

Multi-model page stream segmentation: Finally, for both datasets, accuracy and Kappa statistics improve significantly if image and text feature types are combined. The two compared strategies, early fusion of intermediate CNN feature outputs in an MLP architecture versus late fusion of weighted predictions from each individual classifier, show a clear advantage for the late fusion

strategy. Including LDA topic composition and topic difference as additional feature types in the multi-modal classification only improves the results for the late fusion strategy. Accuracy for the Archive26k dataset increases up to 93 %, resp. ca. 71 % above chance-level agreement. Performance gains for the Tobacco800 dataset are even more significant (ca. 93 % accuracy, ca. 86% kappa agreement).

Finally, we evaluate the performance of the best model setup (late fusion of image, text and topic-based classifier decisions including predecessor page information) on our hold out test set. Tables 4 and 5 report the overall performance for each dataset, as well as evaluation statistics for single-page and multi-page documents separately. The statistics show high performance rates in terms of accuracy for both datasets. However, the kappa agreement measure and confusion matrices reveal differences between the two datasets. While our approach achieves substantial agreement[13] between human labels and machine labels for both single-page and multi-page documents from the Tobacco800 dataset, we see only a close-to-substantial agreement for single-page documents from the Archive26k dataset. Pages from Archive26k multi-page documents, however, can be classified with substantial agreement, as well. The relatively large number of 398 false negatives shows that our PSS approach actually seems to have some struggle identifying single-page documents as such. To a large extent, this can be attributed to inconsistent annotations of documents in the dataset, as the qualitative evaluation reveals.

5.2 Qualitative evaluation

Although first pages and subsequent pages of documents can be distinguished with high accuracy, our improved PSS approach still makes a considerable number of errors. The confusion matrix in Table 4 shows two types of errors for the binary classification of pages: False positives (FP) and false negatives (FN). According to the manually separated pages in the gold standard, FP are subsequent pages (class SD) that are recognized by the classifier as first page (class ND). FN are defined the other way around.

Regarding the entire Archive26k test set, FN account for more than 80% of all errors while the number of FP is relatively low. This FP-FN mismatch means that our CNN architecture splits the page stream into fewer documents than there are actually in the gold standard. When looking at multi-page documents only, the shares of FP to FN are much more even and the error rate is also rather small. In sum, correct identification of single-page documents as annotated in our dataset pose the hardest challenge to our model.

Figures 4 and 5 show examples of false positives, resp. false negatives of 'first pages' of a document. On closer inspection, many FP prove to contain characteristics of valid first pages such as logos and address headers or blocks with person and organization names. Some of these cases can be regarded

[13] Landis and Koch (1977) specify Cohen's kappa values above 0.4 as moderate agreement, and values above 0.6 as substantial agreement.

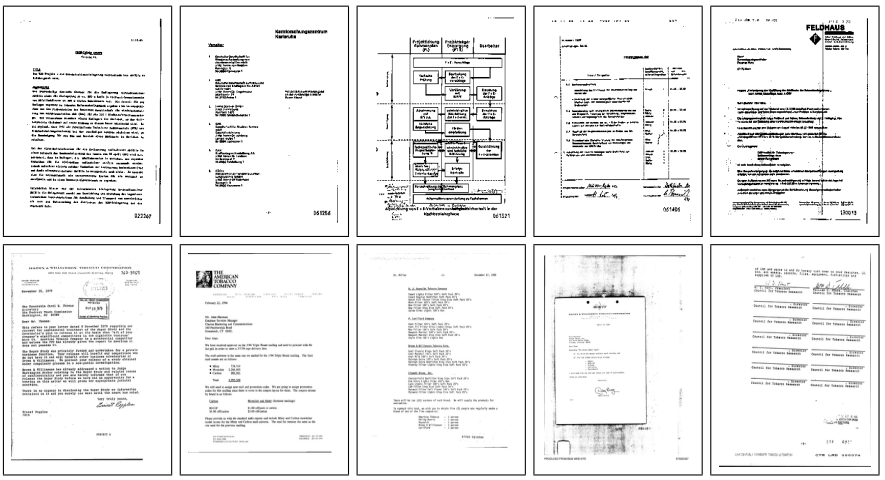


Fig. 4 Examples for false positives (FP) of class 'new document' predictions (Archive26k above, Tobacco800 below). In fact, some FPs comprise textual and visual characteristics of actual document beginnings and hint to annotation errors in the gold standard. Other FPs comprise complex layout structures apparently posing a challenge to the classifier.

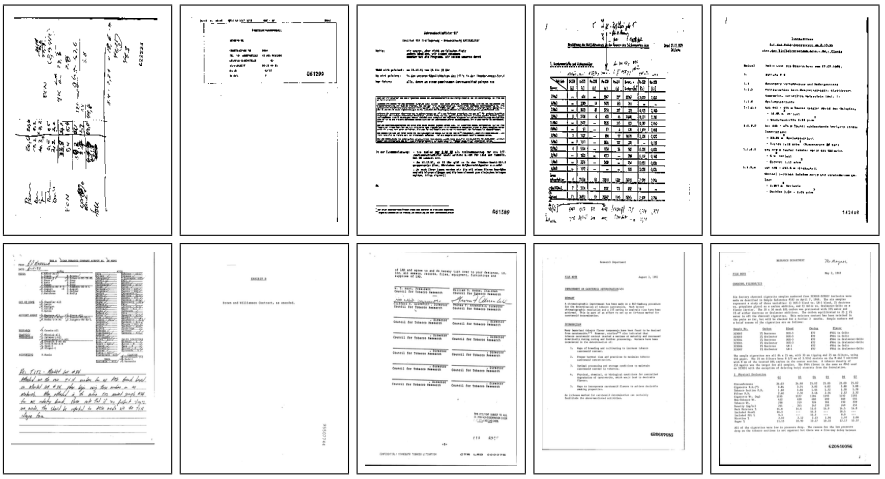


Fig. 5 Examples for false negatives (FN) of class 'same document' predictions (Archive26k above, Tobacco800 below). As for FP, complex (tabular) layout structures seem to confuse the classifier but also are inconsistently annotated in the gold standard. In the Archive26k dataset, many pages consisting of handwritings are FNs.

as false gold annotations in the dataset. Other FP comprise complex layout structures such as organizational charts or tables. Such pages seem to be inconsistently annotated in the dataset rather often. Sometimes they are treated as an attachment to the main document, sometimes as a single-page document on their own. The same can be observed for FN, which often contain document attachments, complex (tabular) layouts, or hand-written content annotated

as single-page documents. In addition the problem of inconsistency of gold annotations, in such cases text-based classification relying on OCR output often has not enough meaningful input, and image-based classification misses reliable layout features, too.

Overall, from qualitative inspection we can learn several things: PSS needs careful annotation of training data like any other supervised machine learning approach. Both of our investigated datasets could be more coherent when it comes to non-standard single-page documents such as tables, figures, charts or letter attachments. At the same time, although an automatic split (FP) is counted as an error in the quantitative evaluation, it nevertheless can represent a meaningful, content-related split for our application of retro-digitizing a large paper archive. Whether the prevalence of 'under-splitting' of the stream in our Archive26k dataset (FN) is desirable or not depends on later use cases of the data. For some retrieval tasks, preservation of larger contexts might be preferred over false splitting into sub-document chunks. In other scenarios, a higher accuracy, especially for correct identification of single-page documents might be preferred. Our model would allow for balancing between FN or FP by applying class-specific loss weights during the training of the neural modules it consists of.[14]

6 Conclusion

We presented a first multi-modal approach for page stream segmentation based on the binary classification of pages with convolutional neural networks and evaluated it on two real-world datasets. With a thorough process of model selection, we created an approach which is able to segment a continuous flow of document images with very high-accuracy (up to 89 % accuracy on our in-house test set, even up to 95 % accuracy on the subset of multi-page documents). In an extended quantitative and qualitative evaluation, this article makes the following four main contributions to the development of page stream segmentation:

- Information from OCR-ed texts and scanned images can successfully be combined in a multi-modal classification approach to significantly improve the performance compared to single-modality classifiers. This finding is consistent with research which has already shown similar results for document image classification (Noce et al., 2016). In accordance with Rusiñol et al. (2014), we found the 'late fusion' strategy of a weighted ensemble of class probabilities from separate classification modules the most successful combination strategy.
- Transfer learning drastically improves PSS, especially for small datasets. Our text-based CNN module is using fastText embeddings pre-trained on

[14] Early experiments showed a slightly worse overall performance of our architecture when applying class-specific loss weights, although we have a rather high class imbalance between SD and ND pages.

Wikipedia texts which improved performance especially for the smaller Tobacco800 dataset, and allows for semantic representation of OOV words which often occur due to OCR-errors and domain-specificity of the dataset. Latent semantic features generated from LDA topic models (potentially trained on large external datasets) further contribute positively to text-based PSS, if the difference of topic distributions of consecutive pages is taken into account. Our image-based module uses the VGG16-architecture pre-trained on the 'imagenet' dataset. Fine-tuning the top pre-trained layers drastically improved performance for PSS as well.
- Considering short sequences of two consecutive pages instead of classifying single pages on their own can contribute to successful PSS if used in combination with the right classification approach. The linear SVM on textual data was not able to profit clearly from predecessor page information to classify a target page. In contrast, the tested CNN architectures were able to significantly improve their performances by additional learning from predecessor pages.
- The publication of the source code of our experiments[15] together with the training and test data of the publicly available Tobacco800 dataset not only will allow for the reproduction of our experiment results but also for a fair comparison with new PSS approaches in the future.

The approach allowed us to drastically reduce costs for separating batch-scanned pages into document units for our project of retro-digitizing a research archive of around one million pages. From our research, we see a great benefit for digitization projects not only in industry, but also for public administration, archives, and libraries as well as for applications in data journalism, digital humanities, or computational social sciences which more and more make use of the potential of large (retro)-digitized document collections.

In future work, we plan to apply and optimize our model for sequence modeling of long page sequences, and for digital image classification. Instead of just two neighboring pages, taking longer sequences of entire binders into account did not further improve the results in some early experiments we conducted with recurrent neural architectures such as LSTM and GRU (Cho et al., 2014). However, we suspect that sequence modeling with attention-based transformer blocks (Vaswani et al., 2017) might improve PSS. Here, we see further potential in the combination of new neural architectures with long sequential image and text data.

Acknowledgements This work has been realized at the University of Leipzig (Germany) in the joint research project "Knowledge Management of Legacy Documents in Science, Administration and Industry" together with the Helmholtz Research Centre for Environmental Health in Munich and the CID GmbH, Freigericht. The authors thank colleagues at Helmholtz and CID for their valuable support.

[15] All our experiments regarding multi-modal PSS with CNN architectures can be found on this Github repository: `https://github.com/uhh-lt/pss-lrev`

References

Onur Agin, Cagdas Ulas, Mehmet Ahat, and Can Bekar. An approach to the segmentation of multi-page document flow using binary classification. In *Proceedings of the 6th International Conference on Graphic and Image Processing*, 2015. doi: 10.1117/12.2178778.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. URL `http://www.cs.princeton.edu/~blei/papers/BleiNgJordan2003.pdf`.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734. Association for Computational Linguistics, 2014. doi: 10.3115/v1/D14-1179. URL `http://aclweb.org/anthology/D14-1179`.

Hani Daher and Abdel Belaïd. Document flow segmentation for business applications. In *Proceedings of Document Recognition and Retrieval XXI*, pages 9201–9215, San Francisco, France, 2014. URL `https://hal.archives-ouvertes.fr/hal-00926615`.

Hani Daher, Mohamed-Rafik Bouguelia, Abdel Belaid, and Vincent Poulain D'Andecy. Multipage administrative document stream segmentation. In *Proceedings of the 22nd International Conference on Pattern Recognition*, pages 966–971, 2014. doi: 10.1109/ICPR.2014.176.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990. ISSN 1097-4571.

Rong-en Fan, Kai-wei Chang, Cho-jui Hsieh, Xiang-rui Wang, and Chih-jen Lin. LIBLINEAR: a library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008. URL `http://jmlr.org/papers/volume9/fan08a/fan08a.pdf`.

Ignazio Gallo, Lucia Noce, Alessandro Zamberletti, and Alessandro Calefati. Deep neural networks for page stream segmentation and classification. In *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications*, pages 1–7, 2016. doi: 10.1109/DICTA.2016.7797031.

Albert Gordo, Marcal Rusiñol, Dimosthenis Karatzas, and Andrew D. Bagdanov. Document classification and page stream segmentation for digital mailroom applications. In *Proceedings of the 12th International Conference on Document Analysis and Recognition*, pages 621–625, 2013. doi: 10.1109/ICDAR.2013.128.

Ahmed Hamdi, Joris Voerman, Michael Coustaty, Aurelie Joseph, Vincent D'Andecy, and Jean-Marc Ogier. Machine learning vs deterministic rule-based system for document stream segmentation. In *Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 77–82, 2017. doi: 10.1109/ICDAR.2017.332.

Ahmed Hamdi, Michael Coustaty, Aurelie Joseph, Vincent D'Andecy, Antoine Doucet, and Jean-Marc Ogier. Feature selection for document flow segmentation. In *Proceedings of the 13th IAPR International Workshop on Document Analysis Systems*, pages 245–250, 2018. doi: 10.1109/DAS.2018.66.

Adam Harley, Alex Ufkes, and Konstantinos Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In *Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 991–995, 2015. doi: 10.1109/ICDAR.2015.7333910.

Daniel Isemann, Andreas Niekler, Benedict Preßler, Frank Viereck, and Gerhard Heyer. OCR of legacy documents as a building block in industrial disaster prevention. In *Proceedings of the DIMPLE@LREC Workshop on Disaster Management and Principled Large-scale information Extraction for and post emergency logistics*, 2014.

Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, pages 137–142, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. ISBN 978-3-540-69781-7.

Romain Karpinski and Abdel Belaïd. Combination of structural and factual descriptors for document stream segmentation. In *Proceedings of the 12th IAPR Workshop on Document Analysis Systems*, pages 221–226, 2016. doi: 10.1109/DAS.2016.21.

Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751. Association for Computational Linguistics, 2014. doi: 10.3115/v1/D14-1181. URL `http://aclweb.org/anthology/D14-1181`.

Jayant Kumar, Peng Ye, and David S. Doermann. Structural similarity for document image classification and retrieval. *Pattern Recognition Letters*, 43:119–126, 2014.

J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977. ISSN 0006341X, 15410420. URL `http://www.jstor.org/stable/2529310`.

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning*, pages 1188–1196, 2014.

David Lewis, Gady Agam, Shlomo Argamon, O. Frieder, D. Grossman, and J. Heard. Building a test collection for complex document information processing. In *Proceedings of the 29th Annual International ACM SIGIR Conference*, pages 665–666, 2006.

Thomas Meilender and Abdel Belaïd. Segmentation of continuous document flow by a modified backward- forward algorithm. In *SPIE - Elec-*

tronic Imaging, Los Angeles, USA, 2009. URL https://hal.inria.fr/inria-00347217.

Andreas Niekler and Patrick Jähnichen. Matching results of latent dirichlet allocation for text. In *Proceedings of the 11th International Conference on Cognitive Modeling*, pages 317–322. Universitätsverlag der TU Berlin, 2012.

Lucia Noce, Ignazio Gallo, Alessandro Zamberletti, and Alessandro Calefati. Embedded textual content for document image classification with convolutional neural networks. In *Proceedings of the 2016 ACM Symposium on Document Engineering*, pages 165–173, New York, 2016. ACM. ISBN 978-1-4503-4438-8. doi: 10.1145/2960811.2960814.

Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, jan 1979. doi: 10.1109/tsmc.1979.4310076. URL https://doi.org/10.1109/tsmc.1979.4310076.

Xuan-Hieu Phan, Cam-Tu Nguyen, Dieu-Thu Le, Le-Minh Nguyen, Susumu Horiguchi, and Quang-Thuy Ha. A hidden topic-based framework toward building applications with short web documents. *IEEE Transactions on Knowledge and Data Engineering*, 23(7):961–976, 2011. ISSN 1041-4347. doi: 10.1109/TKDE.2010.27.

Marçal Rusiñol, Volkmar Frinken, Dimosthenis Karatzas, Andrew D. Bagdanov, and Josep Lladós. Multimodal page classification in administrative document image streams. *International Journal on Document Analysis and Recognition*, 17(4):331–341, 2014. ISSN 1433-2833. doi: 10.1007/s10032-014-0225-8.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. URL http://arxiv.org/abs/1409.1556.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Gregor Wiedemann. Proportional classification revisited: Automatic content analysis of political manifestos using active learning. *Social Science Computer Review*, 37(2):135–159, 2019. doi: 10.1177/0894439318758389. URL https://doi.org/10.1177/0894439318758389.

Gregor Wiedemann, Eugen Ruppert, Raghav Jindal, and Chris Biemann. Transfer learning from LDA to BiLSTM-CNN for offensive language detection in Twitter. In *Proceedings of GermEval Task 2018, 14th Conference on Natural Language Processing (Konvens)*, pages 85–94, Vienna, Austria, 2018. Austrian Academy of Sciences.