# PREPARING LEGAL DOCUMENTS FOR NLP ANALYSIS: IMPROVING THE CLASSIFICATION OF TEXT ELEMENTS BY USING PAGE FEATURES

Frieda Josi[1], Christian Wartena[1] and Ulrich Heid[2]

[1]University of Applied Sciences and Arts Hanover, Expo Plaza 12,
30559 Hannover, Germany
[2]University of Hildesheim, Universitätsplatz 1, 31141 Hildesheim, Germany

### *ABSTRACT*

*Legal documents often have a complex layout with many different headings, headers and footers, side notes, etc. For the further processing, it is important to extract these individual components correctly from a legally binding document, for example a signed PDF. A common approach to do so is to classify each (text) region of a page using its geometric and textual features. This approach works well, when the training and test data have a similar structure and when the documents of a collection to be analyzed have a rather uniform layout. We show that the use of global page properties can improve the accuracy of text element classification: we first classify each page into one of three layout types. After that, we can train a classifier for each of the three page types and thereby improve the accuracy on a manually annotated collection of 70 legal documents consisting of 20,938 text elements. When we split by page type, we achieve an improvement from 0.95 to 0.98 for single-column pages with left marginalia and from 0.95 to 0.96 for double-column pages. We developed our own feature-based method for page layout detection, which we benchmark against a standard implementation of a CNN image classifier.*

*The approach presented here is based on corpus of freely available German contracts and general terms and conditions. Both the corpus and all manual annotations are made freely available. The method is language agnostic.*

### *KEYWORDS*

*PDF Document Analysis, Legal Documents, Layout Detection, Feature and Text Extraction, Classification, Machine Learning, Deep Convolutional Networks, Image Recognition.*

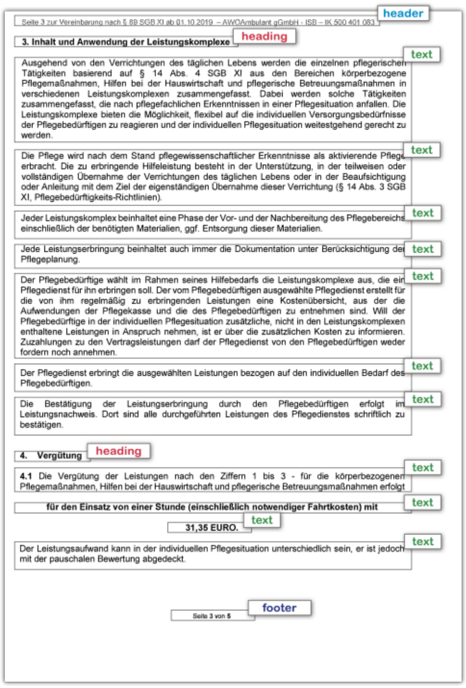
## 1. INTRODUCTION

Many documents are only available as PDF. This is especially the case for legal documents where one exact copy including layout and signatures is distributed and archived. Extracting the text from a legal document is often challenging since e.g. contracts often have a complex structure with lists, footnotes, side notes, multiple columns, headers and footers and so on. Moreover, contracts often consist of several parts, like address page, signature page, project description, terms of service etc. which each may have a completely different layout.

In order to extract texts from a PDF we first identify characters, then regions of closely neighbouring characters (words) and finally regions with dense text. Once we have defined these

David C. Wyld et al. (Eds): NATP, ACSTY, CCCIOT, MLSC, ITCSS - 2022
pp. 17-29, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120102

regions, we classify them into several types before we extract the text. An example of a PDF page and the extracted text regions is given in Figure 1. In this figure, the class of each text region is given.



Seite 3 zur Vereinbarung nach § 89 SGB XI ab 01.10.2019 – AWOAmbulant gGmbH - ISB – IK 500 401 083

**3. Inhalt und Anwendung der Leistungskomplexe**

Ausgehend von den Verrichtungen des täglichen Lebens werden die einzelnen pflegerischen Tätigkeiten basierend auf § 14 Abs. 4 SGB XI aus den Bereichen körperbezogene Pflegemaßnahmen, Hilfen bei der Hauswirtschaft und pflegerische Betreuungsmaßnahmen in verschiedenen Leistungskomplexen zusammengefasst. Dabei werden solche Tätigkeiten zusammengefasst, die nach pflegefachlichen Erkenntnissen in einer Pflegesituation anfallen. Die Leistungskomplexe bieten die Möglichkeit, flexibel auf die individuellen Versorgungsbedürfnisse der Pflegebedürftigen zu reagieren und der individuellen Pflegesituation weitestgehend gerecht zu werden.

Die Pflege wird nach dem Stand pflegewissenschaftlicher Erkenntnisse als aktivierende Pflege erbracht. Die zu erbringende Hilfeleistung besteht in der Unterstützung, in der teilweisen oder vollständigen Übernahme der Verrichtungen des täglichen Lebens oder in der Beaufsichtigung oder Anleitung mit dem Ziel der eigenständigen Übernahme dieser Verrichtung (§ 14 Abs. 3 SGB XI, Pflegebedürftigkeits-Richtlinien).

Jeder Leistungskomplex beinhaltet eine Phase der Vor- und der Nachbereitung des Pflegebereichs einschließlich der benötigten Materialien, ggf. Entsorgung dieser Materialien.

Jede Leistungserbringung beinhaltet auch immer die Dokumentation unter Berücksichtigung der Pflegeplanung.

Der Pflegebedürftige wählt im Rahmen seines Hilfebedarfs die Leistungskomplexe aus, die ein Pflegedienst für ihn erbringen soll. Der vom Pflegebedürftigen ausgewählte Pflegedienst erstellt für die von ihm regelmäßig zu erbringenden Leistungen eine Kostenübersicht, aus der die Aufwendungen der Pflegekasse und die des Pflegebedürftigen zu entnehmen sind. Will der Pflegebedürftige in der individuellen Pflegesituation zusätzliche, nicht in den Leistungskomplexen enthaltene Leistungen in Anspruch nehmen, ist er über die zusätzlichen Kosten zu informieren. Zuzahlungen zu den Vertragsleistungen darf der Pflegedienst von den Pflegebedürftigen weder fordern noch annehmen.

Der Pflegedienst erbringt die ausgewählten Leistungen bezogen auf den individuellen Bedarf des Pflegebedürftigen.

Die Bestätigung der Leistungserbringung durch den Pflegebedürftigen erfolgt im Leistungsnachweis. Dort sind alle durchgeführten Leistungen des Pflegedienstes schriftlich zu bestätigen.

**4. Vergütung**

**4.1** Die Vergütung der Leistungen nach den Ziffern 1 bis 3 - für die körperbezogenen Pflegemaßnahmen, Hilfen bei der Hauswirtschaft und pflegerische Betreuungsmaßnahmen erfolgt

**für den Einsatz von einer Stunde (einschließlich notwendiger Fahrtkosten) mit**

**31,35 EURO.**

Der Leistungsaufwand kann in der individuellen Pflegesituation unterschiedlich sein, er ist jedoch mit der pauschalen Bewertung abgedeckt.

Seite 3 von 5

Figure 1. Example of a page from a contract with marking of text regions and their classes.

Our approach can also be used for documents where the number of columns on each page is not known and can be different on each page. By improving text extraction, large collections of documents can be processed more efficiently, as some sources of error are reduced.

E.g. texts located in the header and footer can be filtered out instead of being merged in the sentence covering the page break; texts from different columns can be brought in correct order and by the identification of headings the text can be split up in chapters and sections.

## 2. Related Work

Extracting text from scanned PDF documents is still a challenge when it is not known how many columns the individual pages in the documents have, when the task is to ignore header and footer text, and when a large number of PDF documents are to be processed automatically.

The documents used in the present work were only available in PDF, not originating from a text processing tool but scanned from printed and signed paper documents. This is a very common situation for legal documents. Therefore, the analysis of legal documents often has to begin with extensive pre-processing followed by comprehensive cleaning of the texts.

Different approaches were tested for extracting the texts from PDF files and for recognizing the structure of the documents. [1] extract text elements from PDF files to analyze the structure of Chinese books that were available in PDF format. After extraction, the content was assigned both a physical and a logical structure. However, since the data came from books, it was possible to assume that all pages have the same layout, i.e. the number of columns and the positioning of headings is consistent across all pages of the book. This allows the definition of global typography classes. The authors divided the logical structure on page level and on document level. The page level contains the hierarchical arrangement of text elements, such as headers, figures, tables, and footnotes. The document level included the chapter structure, author metadata, and book title. For page-level logical structure extraction, the text and individual letters were extracted from these text blocks to obtain additional features such as boldface for headings. The authors obtained very good results with this process. All classes such as *header and footer*, *heading font* etc. are identified with an accuracy above 90% and the lowest hit rate has *heading font* with 87.94%. The authors used 1,000 books for their method.

[2] use layout analysis to improve the delimitation of sentences boundaries in financial reports. They use layout analysis to filter out tables, among other things, as these are not helpful for sentence boundary detection. They try to separate the content of the document from other information.

Contiguous text sections on a document page are not necessarily extracted as a single unit, so there are also some works that deal with merging contiguous text areas from extracted PDF files, for example[3]. The authors have developed a three-stage procedure for this purpose. In the first stage, contiguous text blocks are to be identified on a layout basis. In the next stage, a rule-based classification of the text blocks is performed using categories, and in the third stage, these classified text blocks are to be summarized in the correct order. At the end, the text can be extracted from the summarized text blocks.

In a simple extraction of text from PDF files, text is also extracted from the header and the footer, such as the page number, or the name of the contract creator in the header. However, these components of the document are a hindrance for analyzing the content of the contract and also for comparing document versions. For the detection of headers and footers, [4] used a layout-based approach. This approach is based on the use of geometric coordinates. In addition, the authors use the occurrence of digits as an indicator of a text element in the header or footer and the length of the text as a supplementary feature. Using the coordinates of the text blocks from the PDF files, a structural sorting per page is possible.

Methods for analyzing the visual, physical, and logical structure of PDF files are often developed for scientific papers, for example by [5], [6], [7] and for Newspapers e.g. by [9]. In [9], also technical documentation is transferred into an XML structure starting from a PDF document. Legal documents have however not yet been extensively used in extraction and analysis work. Some research projects in this area are described by [10], which identifies argumentation structures in court proceedings, and by [11], which classifies and automatically summarizes legal texts. For the removal of texts which originate from headers and footers, [12] use features that are defined based on the neighboring pages. As soon as the text line candidates exceed a threshold of equal characters and correspond to a minimum occurrence, these texts can be filtered out as headers and footers.

The work of [13] is similar to our objective - classifying text elements - but they use a large variety of document types e.g. email files, power point files, as well as of text formats, e.g. word files, PDF, and others. Therefore, they divide for example the class "heading" into headings in tables, of emails and of other document types. [13] use a conditional random field model for the prediction and achieve an average accuracy of 0.83 for all 13 classes which they distinguish.

Scientific works, in which Convolutional Neural Networks (CCNs) are used for the classification task of image-based document pages, are reported by [14], [15], and [16], among others. [14] proposed a method (document domain randomization (DDR)) that does not need manually annotated document pages, but works with generated pseudo-pages. However, the extraction of the texts is not the goal, since the pages are randomly composed from components of scientific papers. Thus, the authors do not need to use manually labelled data. The goal is to separate textual areas from figures and tables. In [15], a CNN is used to classify document types. The classes for the image-based document pages are e.g. “email”, “news article”, “file folder”, “letters”, “memo”, and so on. [16] performs object detection on image-based document pages using CNN. The objects they want to identify are “stamps”, “logos”, “signatures”, “tables” and “text blocks”.

## 3. DATA AND METHOD

In this section we will present our method for the classification of text elements and the data used for training and evaluating our methods.

### 3.1. Document Corpus

For this work, a corpus of publicly available German contracts was compiled, primarily contracts from the city administrations of Hamburg and Bremen, dating from 2014 to 2019, that were released on the internet in the context of the cities open government policy. In addition to these two sources, general terms and conditions, found on the internet, were added to the corpus.

The sources used for the corpus can be found in appendix A.

For a part of the corpus all pages were classified as being in one-column layout, two column layout or one-column layout with marginals. For another small part of this corpus all text regions detected  were classified manually. The annotation was done by a student assistant who corrected the classification of a first version of the classifier trained on a very small set of data, not part of the current corpus. Over ten thousand text regions were classified in this way. Exact numbers are given in Table 1. The documents we used for the classification of the text boxes are the same documents that were used as test documents for the page layout recognition. The documents for the training set were separated and no longer used for the recognition of the text classes. For the classification of the text classes, we used x-fold cross-validation with x = 10.

Table 1. Size of the corpus.

| | **Layout prediction** | | | **Text class prediction** |
|---|---|---|---|---|
| | Test set | Training set | Total | Total (cross valid. used) |
| Number of documents | 64 | 249 | 313 | 70 |
| Number of pages | 417 | 1,859 | 2,276 | 276 |
| Number of text elements | - | - | - | 20,938 |

### 3.2. Method

The goal of our work is to classify text fields on a page in order to improve the extraction from a scanned PDF-document. An obvious way to do this is to train a classifier using various features of a text region based on the size, position and content of the text region. Many of the geometric features, however, are context dependent, i. e.  the meaning of a feature depends on the type of page layout. Thus, we first classify each page, since in legal documents it is often found that different layout types are used within a single document. Subsequently, we use the layout information to classify the text regions. Here we either can use the probabilities for each layout class as features for the second classifier, or alternatively we train different classifiers for each layout class.

The pipeline for using the global layout features to improve the classification of text classes is given in Figure 2.

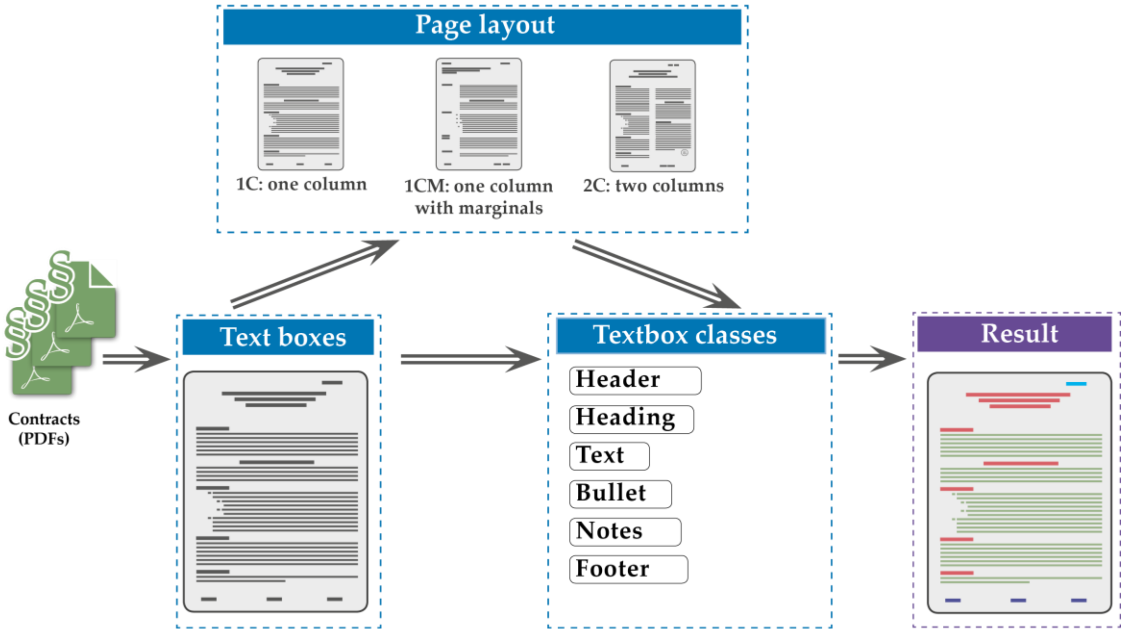


Figure 2. Pipeline for using global page features to classify layout elements.

In the first step, we have compiled a collection of legal documents. Then we extract all text elements from a PDF page and get the exact coordinates and other features such as font size, etc. for each element. After that, the layout of each page is classified using the marker method, see Section 3.3.

We use these layout classes, e.g. two-column page, as a feature to improve the classification of texts classes. The result of this process is a document collection where for each text of an extracted text area, it is known to which class it belongs. Thus, classes that are not needed can be filtered out and the remaining text can be used for text analysis or for other processes.

### 3.3. Layout Recognition

As mentioned before, in legal documents we find various types of page layout that often change within the document or even on one page. E.g. the main text of a contract can be in a one-column layout while the general conditions, that are part of the contract is in two column layout, followed by a page of general remarks and signatures again in a kind of one-column style. We identified three main layout types in various collections of contracts: one-column layout, two-column layout and one-column with marginals. In the current collection of contracts, presented above, the last type of layout usually is a kind of table, where we have short definitions on the left-hand side and a longer text at the right-hand side. In another collection of contracts that we have used, but that cannot be made publicly available, we also found many examples where the section headings were written in the left margin.

In order to predict which layout each individual PDF page will have, we use 100 vertical markers for each page. For each of these 100 markers, the heights of the text areas that this marker intersects with are accumulated and normalised (with the highest marker value per page). To avoid that text elements from the header or footer distort our calculations, we have not taken into account the top and bottom height of a page in the calculation. This is shown schematically in Figure 3. The dark blue highlighted line in Figure 3 intersects with three text boxes (see orange ellipses in the figure). The values of all 100 markers are calculated in this way. The text box height is marked in the illustration as a text area with an orange bold border, the other text areas on this page are highlighted by dark grey rectangles.

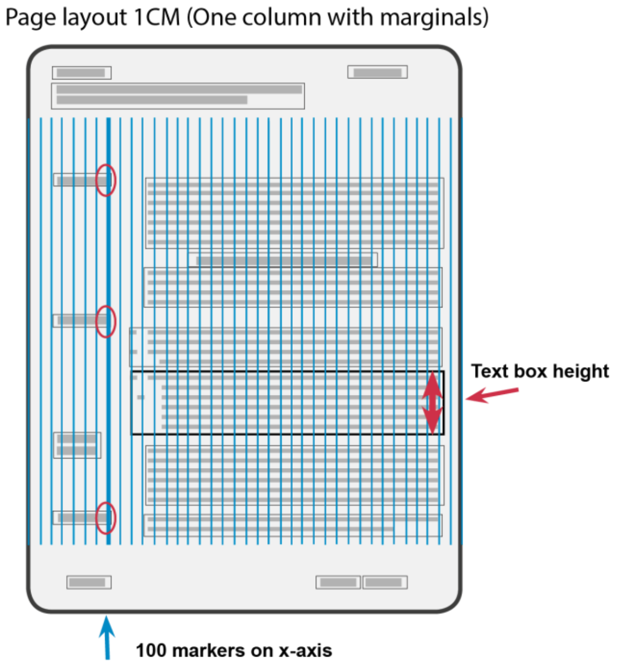


Figure 3. Schematic representation of the calculation of the marker values.

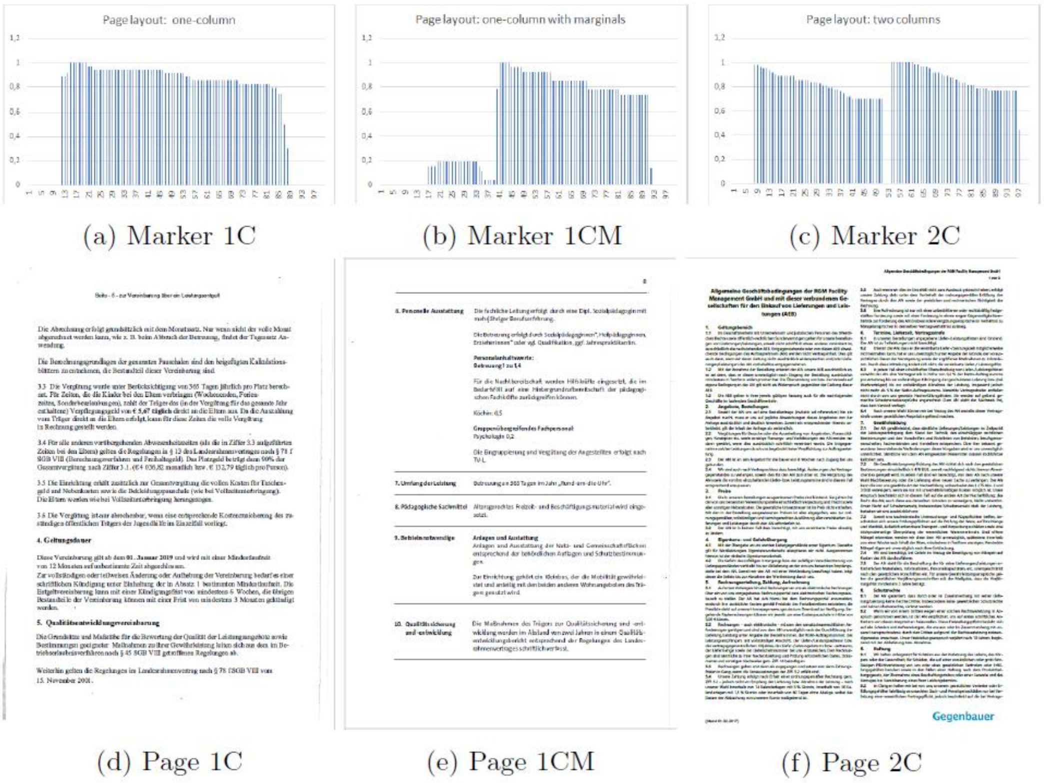
Figure 4 shows prototypical examples for each layout and the associated marker values. The marker values for all three layout types are easily distinguishable for clear prototypical page examples. For the page with one column, the markers value decrease to the right (less intersections, as not all lines are completely filled with text). For the one-column layout with the left marginalia, the markers remain lower over the area of the left headings and increase significantly at the start of the text column. Towards the right margin the values then decrease again. For the two-column page, the low values between the two columns are clearly visible. Here in the example, it is a prototypical page that is well filled with text, but if the two-column page contains only partial text, an incorrect classification may occurs.

### 3.3.1. Our feature-based Approach

For page layout classification, we trained a Random Forest classifier and a Support Vector classifier. The hyperparameters for Support Vector Classifier are optimized for 100 markers. We use Stratified KFold with random state 42 and a fold number of 10 for the Random Forest Classifier and the Support Vector Classifier. The features we use for training the classifiers are the 100 values of markers truncated at the top and bottom of the page and the area of all text boxes per page. The results are given in Table 3.

### 3.3.2. Image Classification with Deep Convolutional Networks

For the image-based classification of the document pages with convolutional networks, we use an implementation of PyTorch (https://github.com/aleksandraklofat/image_classifier). The pre-trained model vgg16 [17] (https://pytorch.org/vision/stable/models.html) were trained with the own document pages (417 document pages were converted to png images with pdf2image). We used the default settings. The document pages from the test set are also the pages extracted by our feature-based procedure. The results are given in Table 5.



(a) Marker 1C (b) Marker 1CM (c) Marker 2C

(d) Page 1C (e) Page 1CM (f) Page 2C

Figure 4. Prototypical page examples and the corresponding marker values.

## 3.4. Classification of Text Elements

For each page we first extract all dense and cohesive text areas. For the extraction of these areas we use PDFMiner [18]. The exact areas that are extracted of course depend on the parameters used for extraction and we have to be sure that the same values have been used for training and application of the classifier.

A text area can have various functions on a page. We identified six frequently occurring types of text regions that can be distinguished in our collection of documents. Other types, like image captions, might frequently occur in other documents but are not present in our corpus of contracts and general terms and conditions of business. The six types are:

- **Header:** Line of text on the top of the page, with one to three text elements not belonging to the main text flow.
- **Heading:** Headings and subheadings.
- **Text:** Main text, including numbered paragraphs, text from listings etc.

- **Bullet:** Bullet points.
- **Notes:** Hand written notes and stamps or text from a small table.
- **Footer:** Includes everything below the last text line, e.g. page numbers, footnotes.

The features we use for the text structure elements are: the geometry of the text box, e.g. its coordinates; the neighbourhood of the text box, the presence of adjacent text boxes i.e. its distance between the text boxes; the text box area; the height and width of the text box; the features for recognising headings, such as bold, capital letters, a colon is the last character of a text box; the number of special characters and whether the text of a text box element is a bullet and others (detailed list in Table 2).

Table 2. Features for each text box for classification.

| **Feature(s)** | **Data type** |
|---|---|
| Geometry of the box | |
| Coordinates (4 values) | float |
| Height of text box in pixels | float |
| Width of the text box in pixels | float |
| Area of the text box | float |
| Neighbourhood | |
| Distance to the top \| bottom \| left \| right of the page (4 values) | float |
| Adjacent text box to the top \| bottom \| left \| right (4 values) | boolean |
| Font | |
| Font is bold | boolean |
| Font is italic | boolean |
| Font size | float |
| Font is capitalised | boolean |
| Text | |
| Starts with a paragraph mark | boolean |
| Text ends with a colon in the text box | boolean |
| Text box consists of bullets | boolean |
| Number of characters of an extracted text box | integer |
| Number of special characters | integer |

From the page layout classification, four values representing the page layout are added as features for each text element. This is to improve the prediction for the text classes. The prediction of the text structure classes is performed using a *Support Vector Classifier* (SVC) as well as a *Random Forest Classifier*, both from the SciKitLearn-Library (https://scikit-learn.org/stable/index.html). Finally, we train a classifier on a part of the manually annotated data. For evaluation we use the test set for layout prediction and a 10-fold cross validation scheme for text classes. For the cross validation we use stratified sampling and for each partition we balance not only the fractions of the text box classes that have to be predicted but also balance the layout types that the boxes were taken from. Thereby we ensure that all types appear in both training and test sets. Alternatively, we do not use the probabilities for each layout class but simply assume that the most probable one is the correct class and train three different classifiers for each class.

# 4. RESULTS AND EVALUATION

## 4.1. Page Layout

The results for the classification of the page layout summarized in Table 3. A Random Forest and an SVC model was trained for the classification. With the Random Forest model we can achieve an accuracy of 0.94 and with the SVC model even 0.95. In the confusion matrix (Table 4), errors in the prediction of the layout classes are shown, from SVC. According to this, between the layout classes 1C and 1CM there are few document pages that are in the intermediate range. 17 pages are incorrectly predicted as a single-column page, but only one one-column page is predicted as two-column page. Table 5 shows the results for image classification with a CNN. The results do not reach the values with the feature-based approach from Table 3. The accuracy is 89%. Detailed values are shown in Table 5 and in Table 6 the confusion matrix gives an overview of the misclassified document pages.

Table 3. Results for layout prediction with SVC and Random Forest.
Accuracy: SVC: 0.95; Random Forest: 0.94

| 101 Features | | | | | |
|---|---|---|---|---|---|
| | **Class** | **Precision** | **Recall** | **F1-score** | **Number of pages** |
| **Random Forest** | 1C | 0.92 | 1.00 | 0.96 | 291 |
| | 1CM | 0.97 | 1.00 | 0.99 | 37 |
| | 2C | 1.00 | 0.73 | 0.84 | 89 |
| | wgt. avg: | 0.97 | 0.91 | 0.94 | |
| | | | | total: | 417 |
| **SVC** | 1C | 0.94 | 0.99 | 0.97 | 291 |
| | 1CM | 0.93 | 1.00 | 0.96 | 37 |
| | 2C | 0.99 | 0.80 | 0.88 | 89 |
| | wgt. avg: | 0.95 | 0.93 | 0.95 | |
| | | | | total: | 417 |
| *Heights of the intersected text boxes per marker (100) and area of all text boxes per page (1).* | | | | | |

Table 4. Confusion matrix from SVC for page layout classes

| **Predicted**<br>**Real** | **1C** | **1CM** | **2C** |
|---|---|---|---|
| **1C** | 288 | 2 | 1 |
| **1CM** | 0 | 37 | 0 |
| **2C** | 17 | 1 | 71 |

Table 5. Results by CNN implementation with PyTorch for page layout classification as image classification. With an accuracy value of 0.89.

| | **Class** | **Precision** | **Recall** | **F1-score** | **Number of pages** |
|---|---|---|---|---|---|
| **CNN** | 1C | 0.97 | 0.80 | 0.88 | 291 |
| | 1CM | 0.32 | 0.84 | 0.47 | 37 |
| | 2C | 0.88 | 0.79 | 0.83 | 89 |
| | wgt. avg: | 0.72 | 0.81 | 0.73 | |
| | | | | image-based document pages, total: | 417 |

Table 6. Confusion matrix from CNN results by implementation with PyTorch for page layout classification as image classification

| **Predicted:**<br>**Real:** | **1C** | **1CM** | **2C** |
|---|---|---|---|
| **1C** | 234 | 48 | 9 |
| **1CM** | 5 | 31 | 1 |
| **2C** | 2 | 17 | 70 |

## 4.2. Text Classes

### 4.2.1. Results of Classification

For the prediction of the text classes, we compare two methods. As a baseline, we classify the text classes without features that contain information about the layout. The results for both classifiers are presented in summarized form in Table 7. With the SVC model we can achieve an accuracy of 0.89 and with the Random Forest model 0.95.

Table 7. Results for text class prediction. Accuracy: SCV: 0.89; Random Forest: 0.95

| | **class** | **precision** | **recall** | **F1-score** | **Number of text elements** |
|---|---|---|---|---|---|
| **SVC** | Header | 0.84 | 0.76 | 0.80 | 394 |
| | Heading | 0.85 | 0.83 | 0.84 | 2,069 |
| | Text | 0.96 | 0.98 | 0.97 | 17,241 |
| | Bullet | 0.84 | 0.77 | 0.80 | 295 |
| | Notes | 0.69 | 0.45 | 0.54 | 437 |
| | Footer | 0.95 | 0.86 | 0.90 | 502 |
| | wgt. avg. | 0.85 | 0.78 | 0.88 | |
| | | | | total: | 20,938 |
| **Random Forest** | Header | 0.92 | 0.84 | 0.87 | 394 |
| | Heading | 0.58 | 0.51 | 0.54 | 2,069 |
| | Text | 0.92 | 0.95 | 0.94 | 17,241 |
| | Bullet | 0.72 | 0.56 | 0.63 | 295 |
| | Notes | 0.63 | 0.44 | 0.52 | 437 |
| | Footer | 0.97 | 0.93 | 0.95 | 502 |
| | wgt. avg. | 0.79 | 0.70 | 0.94 | |
| | | | | total: | 20,938 |

### 4.2.2. Results of separate Classification for each predicted Layout Class

We execute a classification in which the text classes are classified separately according to the page layout. This means that only text elements that are on a one-column page are classified together. We do the same with the other two page layout classes. The distribution of the text elements, across the classes, is shown in Table 8. The results for this separate prediction are shown in Table 9, again for the two classifiers SVC and Random Forest. The prediction of the classes for the texts from the two-column pages and from the page type 1CM benefit from this procedure. With the SVC model, we achieve an accuracy of 0.87 for 1C page type, 0.90 for 1CM, and 0.84 for 2C for the text elements. Using the Random Forest model, we obtain 0.94 for 1C, 0.98 for 1CM, and 0.96 for 2C. More precisely, the improvement for 1CM and 2C can be seen, for example, for the class “heading”: this text type can be used as an important anchor in further text processing. We get 0.95 for the text type “heading” for 1CM and 0.90 for 2C. Single-column pages (1C) do not benefit from detection and separation by page layout types, but do also not represent a major challenge in the extraction process of contract documents.

Table 8. Distribution of the text elements.

| Number of text elements | | | |
|---|---|---|---|
| **Class** | 1C | 1CM | 2C |
| Header | 260 | 66 | 68 |
| Heading | 1,015 | 326 | 726 |
| Text | 9,349 | 1,722 | 6,170 |
| Bullet | 183 | 44 | 68 |
| Notes | 339 | 15 | 83 |
| Footer | 308 | 82 | 112 |
| total: | 11,454 | 2,255 | 7,227 |

Table 9. Results for text class prediction separated for each layout class.

| | **Class** | **Precision** | | | | **Recall** | | | | **F1-score** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1C | 1CM | 2C | wgt. avg. | 1C | 1CM | 2C | wgt. avg. | 1C | 1CM | 2C | wgt. avg. |
| **SVC** | Header | .77 | .96 | .91 | .83 | .41 | .68 | .43 | .46 | .54 | .80 | .58 | .59 |
| | Heading | .54 | .95 | .34 | .53 | .42 | .64 | .35 | .44 | .47 | .77 | .34 | .47 |
| | Text | .90 | .89 | .90 | .90 | .96 | .99 | .92 | .95 | .93 | .94 | .91 | .92 |
| | Bullet | .73 | 1.0 | .76 | .78 | .39 | .02 | .46 | .35 | .51 | .04 | .57 | .45 |
| | Notes | .70 | .83 | .07 | .58 | .47 | .33 | .04 | .38 | .56 | .48 | .05 | .46 |
| | Footer | .94 | .96 | 1.0 | .96 | .72 | .79 | .80 | .75 | .82 | .87 | .89 | .84 |
| | wgt. avg. | .86 | .91 | .83 | .86 | .87 | .90 | .84 | .86 | .86 | .89 | .84 | .86 |
| **Random Forest** | Header | .82 | .90 | .90 | .85 | .76 | .92 | .65 | .77 | .79 | .91 | .75 | .80 |
| | Heading | .80 | .96 | .88 | .90 | .83 | .94 | .92 | .93 | .81 | **.95** | **.90** | .92 |
| | Text | .96 | .99 | .98 | .97 | .97 | .99 | .98 | .98 | .97 | .99 | .98 | .98 |
| | Bullet | .80 | .91 | .87 | .83 | .68 | .89 | .91 | .76 | .74 | .90 | .89 | .80 |
| | Notes | .73 | .67 | .46 | .68 | .56 | .53 | .37 | .52 | .64 | .59 | .37 | .59 |
| | Footer | .94 | .96 | .96 | .95 | .87 | .98 | .95 | .91 | .91 | .97 | .95 | .93 |
| | wgt. avg. | .94 | .98 | .96 | .95 | .94 | .98 | .96 | .95 | .94 | .98 | .96 | .95 |
| *Legend: 1C = one-column, 1CM = one-column with marginals, 2C = two column* | | | | | | | | | | | | | |

Table 10 compares all values for accuracies. The values for the classification of the text elements, which were all trained together and the classification of the text elements divided by page layout. In order to compare the results directly, the accuracy values for the text elements divided by page layout were added as a weighted average. By splitting by page layout, the classification with the SVC becomes worse, from 0.89 to 0.87. The accuracy of the classification with the Random Forest improves from 0.95 to 0.96. The classification with the Random Forest and the text elements split by page layout is thus our best prediction for the text classes.

Table 10. Comparison: Accuracy and weighted average values for text class prediction separated for each layout class (Table 9) and for text class prediction without layout class information (Table 7).

| | | **All text classes (Table 7)** | **Text classes separated for each layout class (Table 9)** | | | |
|---|---|---|---|---|---|---|
| | | | 1C | 1CM | 2C | wgt. avg. |
| **SVC** | wgt. avg. | 0.88 | 0,86 | 0,89 | 0,84 | 0.86 |
| | accuracy | 0.89 | 0,87 | 0,9 | 0,84 | 0.87 |
| **Random Forest** | wgt. avg. | 0.94 | 0,94 | 0,98 | 0,96 | **0.95** |
| | accuracy | 0.95 | 0,94 | 0,98 | 0,96 | **0.96** |
| *Legend: 1C = one-column, 1CM = one-column with marginals, 2C = two column* | | | | | | |

## 5. CONCLUSIONS

With the use of global page information, we can improve the mapping of text elements to a text class for two-column pages and single-column pages with margins. By using the global layout information, texts in legal documents can be extracted more correctly and are thus available for further processing, possibly cleaned of unwanted text classes such as headers and footers.

Our approach is well suited for preprocessing corpora from the legal domain, also if they include documents that have an imbalance in the number of double-column and single-column pages. The size of the corpus does not need to be as large as when using CCNs, but our method still achieves good results. Especially legal contract documents often contain single-column and double-column pages and the number of columns must be identified to ensure an accurate extraction of the text and to maintain the reading flow.

## REFERENCES

[1] Gao, L., Tang, Z., Lin, X., Liu, Y., Qiu, R., Wang, Y. (2011) “Structure Extraction from PDF-based Book Documents” In: Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries. pp. 11-20. JCDL '11, ACM, New York, NY, USA. https://doi.org/10.1145/1998076.1998079

[2] Giguet, E. &Lejeune, G. (2021) “Daniel at the FinSBD-2 task: Extracting list and sentence boundaries from PDF documents, a model-driven approach to PDF document analysis” In: Proceedings of the Second Workshop on Financial Technology and Natural Language Processing. pp. 67-74. - , https://www.aclweb.org/anthology/2020.finnlp-1.11

[3] Ramakrishnan, C., Patnia, A., Hovy, E., Burns, G.A. (2012) “Layout-aware text extraction from full-text PDF of scientific articles” Source Code for Biology and Medicine 7(1), 7 https://doi.org/10.1186/1751-0473-7-7

[4] Dejean, H. & Meunier, J.L. (2006) “A System for Converting PDF Documents into Structured XML Format” In: Document Analysis Systems VII. pp. 129, 140. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg. https://doi.org/10.1007/11669487 12

[5] Klamp, S., Granitzer, M., Jack, K., Kern, R. (2014) “Unsupervised document structure analysis of digital scientific articles” International Journal on Digital Libraries 14 (3- 4), 83-99 https://doi.org/10.1007/s00799-014-0115-1

[6] Klamp, S. & Kern, R. (2016) “Reconstructing the Logical Structure of a Scientific Publication Using Machine Learning” In: Semantic Web Challenges. pp. 255-268. Communications in Computer and Information Science, Springer, Cham; https://doi.org/10.1007/978-3-319-46565-4

[7] Harmata, S., Hofer-Schmitz, K., Nguyen, P.H., Quix, C., Bakiu, B. (2017) “Layout-Aware Semi-automatic Information Extraction for Pharmaceutical Documents” In: Data Integration in the Life Sciences. pp. 71-85. Lecture Notes in Computer Science, Springer, Cham https://doi.org/10.1007/978-3-319-69751-2 8

[8] Namboodiri, A.M. & Jain, A.K. (2007) “Document structure and layout analysis” In: Chaud- huri, B.B. (ed.) Digital Document Processing, pp. 29-48. Springer London. https://doi.org/10.1007/978-1-84628-726-8 2, series Title: Advances in Pattern Recognition

[9] Nojoumian, M. & Lethbridge, T.C. (2011) “Reengineering PDF-based Documents Targeting Complex Software Specifications” Int. J. Knowl. Web Intell. 2(4), 292-319https://doi.org/10.1504/IJKWI.2011.045165

[10] Wyner, A., Mochales-Palau, R., Moens, M.F., Milward, D. (2010) “Approaches to Text Mining Arguments from Legal Cases” In: Semantic Processing of Legal Texts, pp. 60-79. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12837-0 4

[11] Chieze, E., Farzindar, A., Lapalme, G. (2010) “An Automatic System for Summarization and Information Extraction of Legal Information” In: Semantic Processing of Legal Texts, p. 216-234. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg https://doi.org/10.1007/978-3-642-12837-0 12

[12] Lin, X. (2003) "Header and Footer Extraction by Page-Association" In: Document Recognition and Retrieval X. vol. 5010, pp. 164-172. International Society for Optics and Photonics https://doi.org/10.1117/12.472833

[13] Enendu, S., Scholtes, J., Smeets, J., Hiemstra, D., Theune, M. (2019) "Predicting semantic labels of text regions in heterogeneous document images" In: 15th Conference on Natural Language Processing, KONVENS 2019: Bridging the gap between NLP and human under- standing

[14] Meng Ling, Jian Chen, Torsten Moller, P. Isenberg, T. Isenberg, M. Sedlmair, R. Laramee, Han-Wei Shen, Jian Wu, and C. Lee Giles. (2021) "Document domain randomization for deep learning document layout extraction" ArXiv,abs/2105.14931.

[15] Adam W. Harley, Alex Ufkes, and K. Derpanis. (2015) "Evaluation of deep convolutional nets for document image classification and retrieval" In: 13th International Conference on Document Analysis and Recognition (ICDAR), pages 991–995.

[16] Forczmański P., Smoliński A., Nowosielski A., Małecki K. (2020) "Segmentation of Scanned Documents Using Deep-Learning Approach" In: Burduk R., Kurzynski M., Wozniak M. (eds) Progress in Computer Recognition Systems. Advances in Intelligent Systems and Computing, vol 977, pp 141-152. Springer, Cham

[17] Karen Simonyan& Andrew Zisserman. (2015) "Very deep convolutional networks for large-scale image recognition" In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings

[18] PDFminer/pdfminer.six (2021), https://github.com/pdfminer/pdfminer.six, original-date: 2014-08-29T14:04:53Z

## APPENDIX

### A.1 Sources for documents in the corpus

1. City Administration Hamburg: http://suche.transparenz.hamburg.de/dataset?q=vertrag&esq_title=&check_all_
2. City Administration Bremen: https://www.transparenz.bremen.de, Keyword: Vertrag
3. General terms and conditions: Individually researched on websites. All individual links of the sources were saved.

### A.2. Sources for data sets of used corpora

The compiled and used corpora, are available on our project page, under: http://textmining.wp.hs-hannover.de/juver.html

## AUTHORS

**Frieda Josi M.A.** Research assistant at the University of Applied Sciences and Arts Hanover and doctoral candidate at the University of Hildesheim, Faculty of Linguistics and Information Science.

**Prof. Dr. Christian Wartena** Hanover University of Applied Sciences and Arts, Language and Knowledge Processing at the department of Information and Communication.

**Prof. Dr. Ulrich Heid** University of Hildesheim, Computational Linguistics and Language Technology, Faculty of Linguistics and Information Science.