Text Content Based Layout Analysis

José Ramón Prieto, Vicente Bosch, Enrique Vidal
PRHLT Research Center
Universitat Politécnica de Valéncia
Valencia, Spain
joprfon,viboscam,evidal@prhlt.upv.es

Dominique Stutzmann, Sébastien Hamel
Institut de Recherche et d'Histoire des Textes
Centre national de la recherche scientifique
Paris, France
dominique.stutzmann,sebastien.Hamel@irht.cnrs.fr

***Abstract*—State-of-the-art Document Layout Analysis methods rely on graphical appearance features in order to detect and classify the different layout regions present in a scanned text image. In many cases, however, performing this task using only graphical information is problematic or impossible. Only by actually reading some text in the boundaries of the problematic regions it becomes possible to reliably detect and separate these regions. In these situations, textual, content-based features would be required, but since transcription is usually performed after layout analysis, a vicious circle arises. In this work, we circumvent this deadlock by making use of the recently introduced concept of *Probabilistic Index Map*. We use the word relevance probabilities provided by this map to calculate relevant text content based features at the pixel level. We assess the impact of these new features on a historical document complex paragraph classification task. The experiments are performed using both a classical Hidden Markov Model approach and Deep Neural Networks. The obtained results are encouraging and showcase the positive impact text content based features will have on the Document Layout Analysis research field.**

***Index Terms*—Document Layout Analysis, Text Content Based Features, Hidden Markov Models, Deep Neural Networks**

I. Introduction

Higher Order Text Region Classification (HOTRC), is the Document Layout Analysis (DLA) task that tackles the detection and classification of large structural regions present in a page. Detection of these larger structural regions, like paragraphs or side-notes, is an important in order to perform other DLA tasks. Many advances on this task have been performed over the years [1], [2], specially with the advent of Deep Neural Networks [3], [4]. All of these state-of-the-art methods depend on differences in the graphical-appearance of the text regions present in order to detect and classify them.

However, there are layout analysis problems where graphical-appearance or features derived from it will not suffice. In many cases this happens because the considered document does not exhibit any clear graphical clue to distinguish between regions. Sometimes, there might be some graphical hints but they are difficult to detect and/or plainly consistent, specially in large collections. Other times the type of a region can only be told if the graphical information of the following page is considered. In Fig. 1 we see a sample page of the corpus used in this research paper that exemplifies these issues. However, there are images which with only the graphic features could be segmented. The page contains three text blocks: the second one corresponds to a complete record, while the first one is a continuation of a record that started in the preceding page and the third is a record that continues in the next page. There is no graphical gap between adjacent blocks, the capital letter that marks the beginning of the second block is not present at the beginning of the third. Furthermore, there are no graphical clues to distinguish whether the first and the third blocks and full records or they are incomplete records which start or continue in the adjacent pages.

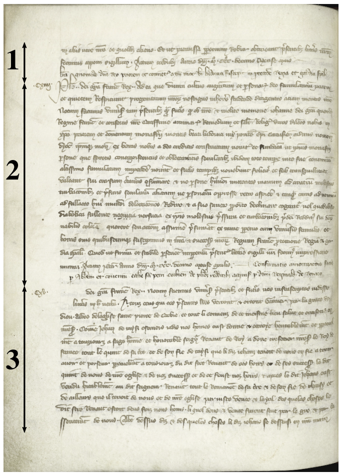


Fig. 1. Sample Chancery corpus (Sec. II) page that exemplifies the issues of depending solely on graphical features to distinguish between 3 regions.

To tackle this situation, here we study the (additional) use of text content based features to improve the accuracy of different DLA models. Since traditionally DLA has been considered a front-end step that must be performed previously to any sort of text recognition, the use of text content based features in DLA generates an obvious vicious circle.

In order to circumvent this deadlock in a realistic manner, we rely on the recently introduced concept of *Probabilistic Index* (PrIx) *Map*, which was developed following concepts and ideas related with word-segmentation-free Key Word Spotting [5]–[9]. A PrIx nicely deals with the intrinsic word level *uncertainty* generally exhibited by a handwritten text image. It can be understood as a "heat-map" representation of the image which highlights positions of words which were likely written in the original manuscript.

In this paper we study how to adequately use PrIx's to derive text content based features useful for DLA. We review the impact of using these features in two different approaches: Hidden Markov Model (HMMs) and Deep Neural Networks (DNNs) when tackling a complex HOTRC task.

Since the error in CER is considerably high for this corpus makes using layout approximations together with transcribed text [10] is not feasible due to the high number of pseudowords, making these techniques not worthwhile on the context of the historical document.

The rest of the article is divided as follows. In Section II we describe the Corpus and the task that must be performed. Next, in Section III a brief overview of the PrIx maps and the necessary adaptations required to use them, both in HMMs and DNNs. Later, in Section IV we review the experimental set-up and evaluation measures, and provide empirical results. Finally in Section V, we provide our conclusions and comment on the future work that is derived from this paper.

II. Corpus

The complete corpus named *Chancery* encompasses around 80 000 images of medieval registers produced by the French Royal Chancery (Archives Nationales and Bibliothèque nationale de France), in the period spanning from 1302 to 1483. They contain charters given by the king of France. This large and iconic collection bears witness to the rationalization of late medieval administration. It is considered a key source to help our understanding of medieval Europe and the rise of centralized nation state. This rise appearing throughout the continent as a consequence of the long lasting wars between France and England.

Fig. 2. Example of pages of the *Chancery* corpus

The sheer size of this corpus has prevented scholars to study it as exhaustively as it deserves. To overcome this situation, a user-friendly search interface based on PrIx[1] was developed in previous projects to access the contents of this key resource. This is helping to increase the knowledge about medieval history and to promote ongoing research in comparative studies on state management and administration.

This corpus is functionally divided into content units called *"Acts"*. These Acts mostly record royal decisions with perpetual validity: creation and privileges of different organizational bodies, confirmations of previous acts, donations, pardons. Confirmations are numerous (45% under king Charles IV) and include the complete text of the previous act, preventing a classification of parts of acts based solely on the textual content. Acts may include up to seven layers of embedding and up to 17 different acts. *Acts* can be short, with several *Acts* in a page, or large and distributed across several pages. Thus, when searching for information about a specific topic it is important for researchers to understand where an *Act* starts and finishes.

Hence we can consider the pages of the collection to be composed of different types of *Acts* or *Act* fragments, depending where they start and end. A *Complete Act* is one that starts and ends in the same page. A *Finishing Act* is one that started in another page but ends in the current, while a *Starting Act* is one that starts in the current page but doesn't end in it. Finally, a *Medium Act* is one that neither ends nor starts on this page. The task considered in the present work is only to detect where an Act ends. Although this does not strictly require the above classification, it should be implicitly taken into account by any method, and it does help understanding the complexity of the task.

Furthermore, this corpus presents instances of inclusion of *Acts* within *Acts* (called "vidimus" or "inspeximus") difficulting a classification of parts of acts based solely on the textual content

In our experiments we considered a subset of the whole *Chancery* collection consisting of 739 page images from the range of volumes. from JJ038 to JJ091. Some of the pages were hand selected to ensure the contents were representative of the whole collection, but most correspond to the full contents of volumes. Details of this dataset can be found in Table I.

TABLE I
Type of Acts statistics found in the Chancery corpus subset used during in the experimentation.

Number of:	Total
Pages	739
Complete Act (CA)	438
Starting Act (SA)	365
Medium Act (MA)	178
Finishing Act (FA)	345

This dataset was divided into training, validation and test partitions. The *training* partition consists of 176 page images including 56 selected by hand to ensure the training set is sufficiently representative, plus 20 randomly selected pages of each volume. For *validation* we randomly selected 16 images from each volume, for a total of 96 pages. Finally, the *test* partition consisted of the remaining 467 pages.

III. Hidden Markov Modelling

Our first approach to act finalization detection was based on HMMs and "vertical layout models". It is an adaptation of the successful statistical framework used for automatic speech and handwritten text recognition, which was already applied successfully to other DLA tasks [2], [11], [12].

To apply this method we must model all *Layout Regions* (LRs) expected in a page and define the *Layout Elements* (LEs) they are composed of. Furthermore, as we want to use the text information extracted from PrIx maps, we should define LEs that have some correlation with their expected text content.

The LRs considered mainly correspond to the different types of *Acts* described in the Sec. II: CA, SA, MA, FA. But,

[1] Available at http://himanis.huma-num.fr/himanis/

our HMM approach obtains the best performance when all possible elements in a page are modelled. Thus, we will also model the upper part of the page `UP`, end of a page `EP` and transition spaces between *Acts* `AT`.

These LRs allow us to represent a page as a sequence of region labels. For example, a page with a Finishing Act (`FA`), two Complete Acts (`CA`) and one Starting Act (`SA`) can be represented by the sequence:

`UP FA AT CA AT CA AT SA EP`

To model the different LRs, adequate LEs must be defined. We consider an *Act* to be composed of three sections: Initial `I`, Middle `M` and a Final `F`. By vertically stacking these elements, all the above types of acts can be easily modelled. For example, a Complete Act (CA) may correspond to two LE combinations: `I M F`, or `I F`, depending on whether the Act is or is not large enough to have a Middle section.

The above modelling specifications are summarized in the following LR dictionary, where some LEs are named exactly as the corresponding LRs (`EP, UP, AT`) :

LR		LEs
UP	:	UP
AT	:	AT
EP	:	EP
CA	:	I F | I M F
SA	:	I | I M
MA	:	M
FA	:	F | M F

For each LE, an optical HMM model is trained. These models take as input a vertical sequence of graphical features [2] (plus the text content based features introduced in Sec. VI). Using the composition rules defined above, LR HMMs are built from the LE models. Lastly, how the different LRs can be stacked to form a correct page will be governed by a prior *Vertical Layout Model* which we will introduce in Sec. V.

IV. Deep Neural Network Modelling

Current state-of-the-art methods for DLA in historical documents are based on deep learning approaches. For document segmentation, CNN-based pixel-wise region predictors like dhSegment [3] and P2PaLA [4] are achieving promising results. In case of text line detection, ARU-net [13] uses residual blocks increasing the representation power and an attention model to focus at different positions and scales.

However these approaches fail to provide enough accuracy and/or robustness for tasks, like the one here considered, where graphical hints do not help distinguishing contiguous text blocks and textual hints are needed to allow reliable block separation. Here we use a DNN architecture different from [3], [4], [13] and follow some ideas of the multimodal perspective adopted in [14] to take into account textual features.

We opted to use residual convolutional blocks [15], which allows us to use a higher number of convolutional layers while avoiding the vanishing gradients issue. The blocks allow the network to find the best way to extract the graphical features. This increases the representative power, as compared to a simple convolutional layer. Every block is composed of 3 convolutional layers followed by batch normalization [16] to normalize the inputs to the non-linear activation function. We use ReLU activation functions. At the end of every block, the output of the activation functions are fed to a Max Pooling layer with non-overlapping kernels of 2×1. This is done to reduce the dimensionality only in the horizontal axis. The number of blocks depends on the size of the input image since this is where the reduction of the dimension is carried out. For this task, we decided to re-size the images to 1024×768, thus a total of 4 residual convolutional blocks were used, where each block had 8, 16, 24, 32 filters respectively. Every convolution uses 3×3 filters and 1 stride. As a result we end up with a tensor dimension of $1024 \times 48 \times 32$.

Next, LSTM layers [17] are used on the output of the last block to capture long-term dependencies across the horizontal axis. We are able to perform this by applying a reshape operation prior to the LSTM layers. The operation consisting in concatenating the channel and the horizontal dimensions, always preserving the vertical dimension. This results in a tensor dimension of 1024×1536. We concatenated 3 BiLSTM layers of 16 hidden units every one, ending up with a tensor dimension of 1024×32.

Finally, we use a linear layer to classify each input row using as input the features obtained from the BiLSTM layers. This linear layer is used to provide the probability for each of the considered region types. The final result is a tensor of dimensions 1024×6; that is, for each normalized vertical position (for 1 to 1024), we have an estimate of the posterior probability for each of the 6 LR classes.

Note that every reduction of the tensor is applied to the horizontal axis, keeping intact the vertical axis. We do so in order to be able to obtain the class probabilities for each row of pixels. To do so, a horizontal softmax is applied. The whole architecture is illustrated in Fig. 3. We can see that the image has 3 more channels. These channels come from the textual characteristics, explained in Sec. VI-B.

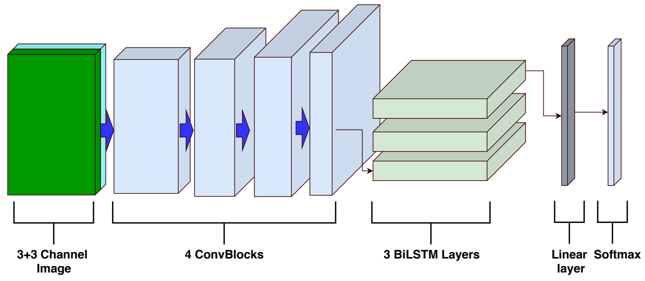


Fig. 3. Illustration of the network design. 6 channel image is created the concatenation of the input image in RGB channels with the 3 images in grey scale created with textual information. The attention variables that are multiplying the textual information are not shown in the figure.

V. Use of Prior Information

A. *Vertical Layout Model in the HMM Approach*

In this case we use a finite-state model to govern how the different LRs can be stacked vertically to create a well formed page. This model is trained with the label sequence descriptions of the training pages. The model that holds this prior information is named *Vertical Layout Model* (VLM) and it plays the same role as the Language Model in automatic speech or handwritten text recognition.

The VLM is implemented as a finite state automata representation of a n-gram model. In order to test the HMM approach when no prior information is used, we also use a zero-gram VLM; i.e., a single-state finite state model with equal probabilities for all the transitions.

B. DNN Output Post-processing Based on Prior Knowledge

As the neural network obtains probabilities for all classes in each row, some inconsistencies tend to appear. For instance, sometimes micro regions of different region types appear in the middle of a large block of another region type. Although the network provides probabilities for all region classes, we are only interested in class `FA`, which is the class that denoting the end of an Act.

In order to mitigate lack of homogeneity in a vertical sequence of labels, we used a Random Forest Classifier to automatically detect inconsistencies in the size (height) of subsequences of class `FA`. To train the classifier to detect these inconsistencies we use the hypotheses produced by the network for the input images of the training set, along with the reference output provided for each image. For test images, the network output subsequences labelled with the `FA` class (which we consider must be at least 10 rows long) are detected. Then a greedy alignment between these subsequencess and their positions in the ground truth (GT) is performed. If the detected subsequences are also labelled with the `FA` class in the GT we mark them as correct otherwise as incorrect. Finally, the input to train the classifier consists in the size of each subsequence plus a bit indicating whether it is a correct subsequence or not. Thus, the classifier is able to learn which subsequence sizes are correct, and then helps filter out inconsistencies.

VI. Probabilistic Indexing and Text Features

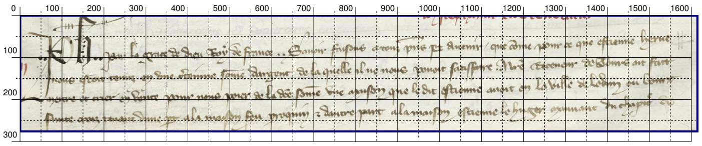
Text based features are derived from PrIx maps, for which a brief overview is given in the following subsection. Then we provide details of the adaptations required to use the PrIx information in the two approaches considered.

A. Word Relevance Probabilities

A PrIx map provides probabilistic information of the textual contents of a page. It is obtained without the need to perform a detailed layout analysis or text baseline detection, nor performing any explicit handwritten text recognition on detected text lines [5], [8], [9]. The PrIx technology used here is fully lexicon-free; that is, no predefined set of "key words" is needed or used. The system detects in the images any textual element (i.e., any arbitrary character sequence) which is sufficiently likely to be a real word [8], [9]. These elements are called *pseudo-words*.

An example of a PrIx map can be seen in Fig. 4. The map contains an entry for each (pseudo-)word `kw` detected in the image. Each entry includes the bounding box coordinates `(x,y,w,h)` and the probability with which the word has been detected `s`, called *relevance probability*.

```
...
<spot kw="TOUZ"         s=1.000  x=878  y=72   w=65  h=47  gt=0 />
<spot kw="UNE"          s=1.000  x=739  y=181  w=53  h=32  gt=1 />
<spot kw="VILLE"        s=1.000  x=1289 y=181  w=66  h=32  gt=1 />
<spot kw="FEU"          s=0.999  x=547  y=230  w=35  h=29  gt=1 />
<spot kw="MAISON"       s=0.999  x=820  y=181  w=87  h=32  gt=1 />
...
<spot kw="MAVANT"       s=0.001  x=1296 y=230  w=104 h=29  gt=0 />
<spot kw="PHILIPS"      s=0.001  x=105  y=72   w=117 h=47  gt=0 />
<spot kw="SAATISFAIRE"  s=0.001  x=1075 y=124  w=113 h=37  gt=0 />
```



Philippes par la grace de Dieu roys de France. Savoir faisons à tous ... maison feu Perrequin et d'autre part à la maison Estienne le Huger mouvant du chapitre de

Fig. 4. Part of the probabilistic index map of a text image region, along with its groundtruth transcript (from the Chancery corpus Sec. II).

Unfortunately these relevance probabilities can not be directly included in a feature vector, along with other graphical features. The sheer size of the indexed (pseudo-)word vocabulary makes this an impractical idea. Thus, to allow proper combination with other graphical features, PrIx data need to be adequately distilled.

B. Extracting Textual Features from PrIx Maps

In order to breach the gap between the PrIx Maps and the graphical zones defined, we need vocabularies that capture the typical word usage in each part (`M, I, F`) of an Act. We searched for specific words which are frequently used in each part to create small vocabularies that help discriminating between them. These specific vocabularies can be determined in an automatic manner by means of a Linear Discriminant Analysis, or provided by an expert. In this case, the repetitiveness of certain expressions used for starting and ending the Acts allowed an expert to easily select reliable vocabularies. However, the extraction of probabilistic indexes has not been reduced to this vocabularies because, in future works, these words will be selected automatically. Using these vocabularies, we can calculate feature values that represent the LE differentiation at row level. For each horizontal row of pixels, we search the PrIx map for bounding boxes that intersect this pixel row. We look at the word in each entry and, if it belongs to the sub-vocabulary of any of the LEs (and surpasses a minimum probability value), we add the probability to the corresponding feature value. Fig. 5 shows a fragment of page image along with the feature vectors obtained from this image for the three LEs `M, I, F`. The bounding boxes of some of the PrIx entries used in the calculation are highlighted in the image.

In the HMM approach, the values of these three features are just appended to the the pixel row graphical features [2], thereby increasing the feature vector dimension by 3. The rest of the process remains the same as explained in Secs. III and V.

The extraction of textual features for the DNNs approach is essentially the same as for HMMs. However, they are normalized because of DNN numeric requirements. For each pixel

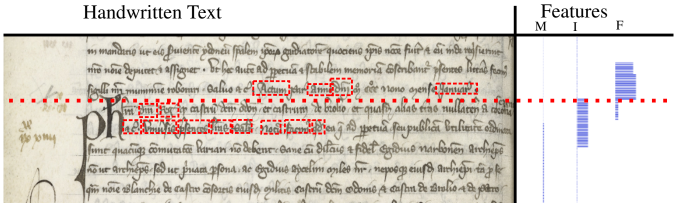


Fig. 5. Illustration of textual features for three layout elements, side-by-side with the image portion they were calculated on. The horizontal dotted line marks the correct division between the two Acts. The Final textual feature (F) increases just before the end of the Act and the Initial textual feature (I) increases at the beginning of the next Act. Words and bounding boxes of some key PrIx entries that have caused these changes in feature values are highlighted in the image.

row at vertical position h, let $M(h), I(h), F(h)$ the values of the features calculated as explained above for the LE's `M, I, F`, respectively. Then the normalized values are obtained as $M(h)/S, I(h), F(h)/S$, where $S = M(h) + I(h) + F(h)$.

These features have to be added to the DNN network graphical input which, as discussed in Sec. VI-B, consists of three RGB channels representing a size-normalized colour image of $H \times W$ pixels, where $H = 768$ and $W = 1024$. In order to adequately fuse these image-like graphical features with the textual features, we created image-like representations of the (normalized) textual features $M(h), I(h), F(h), 1 \leq h \leq H$. To this end, for each h, each feature value is replicated W times. For instance, from the H-dimensional feature vector $M(h), 1 \leq h \leq H$, a $W \times H$ "image", X_M is built as $X_M(h, w) \overset{\text{def}}{=} M(h), 1 \leq w \leq W, 1 \leq h \leq H$. And similarly for the other two features, resulting in three new image-like input channels, X_M, X_I, X_F, each representing the corresponding textual content.

These three channels can be straightforwardly stacked along the three graphical channels, ending up with a tensor of shape $H \times W \times 6$. Fig. 3 already depicted these 3+3 input channels.

Finally, three more parameters to learn are added to the network. Each parameter is used to weight each of the three text feature channels. The new parameters should act as a (weighted) selector to allow the network to decide to which channels should be paid more attention before processing goes on through the convolutional blocks.

VII. Experiments and Results

A. Experimental Set-up

The meta-parameters of the HMM approach were optimized over the validation set. The resulting number of states for each LE optical model was $\{I : 5, M : 15, F : 5, T : 2, B : 7, E : 2\}$. Similarly, the best n-gram order for the VLM was 2. Finally, for the global Viterbi decoding, including the optical HMMs and the VLM, the optimal values for the "grammar scale factor" and word "insertion penalty" were 32 and -512, respectively.

To optimize the neural network we used minibatch SGD and Adam solver [18] with a learning rate of 0.01, and momentum parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$. Classical cross-entropy was used as the loss function. To overcome the problem of class imbalance, weights per class were computed as $w_k = \frac{1}{\log \epsilon + p_c}$, $\epsilon \geq 0$, where p_c is the prior-probability of the c class. The network was trained for 80 epochs without using a development partition to stop iteration when convergence conditions are met.

In our experiments, we used a batch size of 8 images redimensioned to size 1024×768. This was the maximum batch size allowed by our hardware, a single Titan X GPU.

B. Assessment Measures

To evaluate the precision of the proposed models, and thus the impact of using text content based features, we use the Transkribus Baseline Evaluation Scheme (TBES). This evaluation measure was first defined in 2017 as part of the *READ* project [19] and was initially used in an actual competition in the 2017 ICDAR congress [20]. The TBES was defined due to issues found with the existing available measures [21].

As mentioned before, in the corpus GT each end of an Act is marked with a horizontal straight line (a "baseline" in the PAGE format used in Transkribus). The TBES measure was used to compare the output of our systems to this groundtruth (GT). Three main values are computed to evaluate this comparison. The *R-value* quantifies how many of the true Act separations are detected. The *P-value* measures how precisely the detected separations geometrically matches the GT. These values provide information inspired by the well known *Recall* and *Precision* measures used in Information Retrieval. Finally, *F-value* is the harmonic mean of *R-value* and *P-value* and summarizes the evaluation as a single measure.

It is important to note that the TBES tool has an input *tolerance* parameter that specifies how close the hypothesis and GT separation baselines must be to consider they match In our experiments we have set this parameter to the average interline height (128 pixels) observed in the training images. This tolerance represents adequately the precision requirements of the experts.

C. Results

Table II shows *F-value* results for the ending Act detection task on the Chancery dataset described in Sec. II. Results are shown for all the combinations of HMM or DNN statistical model, inclusion or not of prior information and usage or not of text features.

TABLE II
Comparison table of the F-value results obtained with the different combinations of graphical and text content based features, optical models and use of prior information via language models or randomized forest trees.

LM	No Prior		Prior	
Features	Graph.	Graph. + Text	Graph.	Graph. + Text
HMM	0.41	0.71	0.51	0.73
DNN	0.59	0.73	0.80	0.88

From the results, it can be noted that the incorporation of text based content features has a very positive impact on all the models. Furthermore, it is able to mitigate in some manner the negative impact of not using of prior information has on HMMs and DNNs for this task.

In the case of HMMs, the use of text content features provided additional restrictions for decoding. Due to how the layout was modelled and trained, only I or F layout elements could be optimally selected respectively in those regions of the page. This greatly mitigated not using prior information via the VLM.

In general, the use of prior information greatly improved results on all models. This can be specially observed in the models that only used graphical information based features.

VIII. Conclusions and Future work

We have provided an approach to extract text content features from an automatically generated Probabilistic Index Map. We studied how to effectively incorporate these features into two different models: HMMs and DNNs. The impact of these features were evaluated empirically in a challenging region detection task on a historical manuscript collection. Textual content features have proven to have a very positive impact on the layout analysis task considered, where graphical-appearance information is not enough to distinguish between adjacent text blocks.

Additionally, it has been clearly demonstrated that the use of prior layout information has an overall positive impact on performance. This follows the trends that can be seen in Handwritten Text Recognition with DNNs [22].

For future works the approach here presented may open a new world of possibilities for DLA. In particular, we would like to explore more effective ways to incorporate these text content features in the DNN model. Lastly, we believe that research regarding the addition of prior information to DNNs for DLA tasks is still in its infancy and great further improvements are expected when powerful layout prior models, like the VLM used in the HMM approach, can be effectively integrated with the already successful DLM models considered in the present work.

Acknowledgements

This work is part of the *HIMANIS: HIstorical MANuscript Indexing for user-controlled Search* and *HOME: History of Medieval Europe* research projects and has been supported by the European Union's Joint Programming Initiative (JPI) Cultural Heritage and Global Change[2] under the Heritage plus and Digital Heritage programs, grant agreements (France) ANR-15-EPAT-0003, ANR-17-JPCH-0006 and partially supported by the Universitat Politècnica de València under grant FPI-I/SP20190010 (Spain).

References

[1] K. Chen, H. Wei, J. Hennebert, R. Ingold, and M. Liwicki, "Page segmentation for historical handwritten document images using color and texture features," in *2014 14th International Conference on Frontiers in Handwriting Recognition*, Sep. 2014, pp. 488–493.

[2] V. Bosch, J. Calvo-Zaragoza, A. H. Toselli, and E. Vidal, "Sheet music statistical layout analysis," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Oct 2016, pp. 313–318.

[3] S. Ares Oliveira, B. Seguin, and F. Kaplan, "DhSegment: A generic deep-learning approach for document segmentation," *Proceedings of International Conference on Frontiers in Handwriting Recognition, ICFHR*, vol. 2018-August, pp. 7–12, 2018.

[4] L. Quirós, "Multi-Task Handwritten Document Layout Analysis," pp. 1–23, 2018.

[5] A. H. Toselli, E. Vidal, V. Romero, and V. Frinken, "HMM word graph based keyword spotting in handwritten document images," *Information Sciences*, vol. 370-371, pp. 497 – 518, 2016.

[6] T. Bluche, S. Hamel, C. Kermorvant, J. Puigcerver, D. Stutzmann, A. H. Toselli, and E. Vidal, "Preparatory kws experiments for large-scale indexing of a vast medieval manuscript collection in the himanis project," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, Nov 2017, pp. 311–316.

[7] A. H. Toselli, E. Vidal, J. Puigcerver, and E. Noya-García, "Probabilistic multi-word spotting in handwritten text images," *Pattern Analysis and Applications*, Aug 2018.

[8] E. Lang, J. Puigcerver, A. H. Toselli, and E. Vidal, "Probabilistic indexing and search for information extraction on handwritten german parish records," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Aug 2018, pp. 44–49.

[9] J. Puigcerver, "A probabilistic formulation of keyword spotting," Ph.D. dissertation, Universitat Politècnica de València, 2018.

[10] X. Yang, E. Yumer, P. Asente, M. Kraley, D. Kifer, and C. L. Giles, "Learning to extract semantic structure from documents using multimodal fully convolutional neural networks," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 4342–4351, 2017.

[11] V. Bosch, A. H. Toselli, and E. Vidal, "Statistical text line analysis in handwritten documents," in *2012 13th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2012, pp. 201–206.

[12] V. Bosch, A. H. Toselli, and E. Vidal, "Semiautomatic text baseline detection in large historical handwritten documents," in *2014 14th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2014, pp. 690–695.

[13] T. Grüning, G. Leifert, T. Strauß, J. Michael, and R. Labahn, "A two-stage method for text line detection in historical documents," *International Journal on Document Analysis and Recognition*, vol. 22, no. 3, pp. 285–302, 2019.

[14] R. Jain and C. Wigington, "Multimodal Document Image Classification," in *15th IAPR International Conference on Document Analysis and Recognition*, vol. 3, 2020, pp. 71–77.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition Kaiming," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),*, pp. 770–778., 2016.

[16] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," 2015.

[17] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] K. Diederik P. and L. Ba, "Adam: A Method for Stochastic Optimization," *AIP Conference Proceedings*, vol. 1631, pp. 58–62, 2014.

[19] T. Grüning, R. Labahn, M. Diem, F. Kleber, and S. Fiel, "READ-BAD: A new dataset and evaluation scheme for baseline detection in archival documents," *CoRR*, vol. abs/1705.03311, 2017.

[20] M. Diem, F. Kleber, S. Fiel, T. Gruning, and B. Gatos, "cbad: Icdar2017 competition on baseline detection," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, Nov. 2018, pp. 1355–1360.

[21] V. Romero, J. A. Sánchez, V. Bosch, K. Depuydt, and J. de Does, "Influence of text line segmentation in handwritten text recognition," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, Aug 2015, pp. 536–540.

[22] J. Puigcerver, "Are Multidimensional Recurrent Layers Really Necessary for Handwritten Text Recognition?" *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, vol. 1, pp. 67–72, 2017.

[2] http://jpi-ch.eu/