NonDisclosureGrid: A Multimodal Privacy-Preserving Document Representation for Automated Document Processing

Claudio Paonessa[1]

[1] *Institute for Data Science, FHNW University of Applied Sciences and Arts Northwestern Switzerland, School of Engineering, Bahnhofstrasse 6, CH-5210 Windisch, Switzerland*

Abstract

We propose a novel type of document representation that preserves textual, visual, and spatial information without containing any sensitive data. We achieve this by transforming the original visual and textual data into simplified encodings. These pieces of non-sensitive information are combined into a tensor to form the *NonDisclosureGrid* (NDGrid). We demonstrate its capabilities on information extraction tasks and show, that our representation matches the performance of state-of-the-art representations and even outperforms them in specific cases.

1. Introduction

Automated document processing is a pivotal element towards successful digitization for many businesses worldwide. The goal is to transform unstructured or semi-structured information into a structured form for various downstream tasks, hence streamlining administrative procedures in banking, medicine, and many other domains.

Because of the typically sensitive nature of the data used for document processing, the data available to train state-of-the-art systems is limited. Private companies are often obliged to delete documents collected from customers after a certain time period and may not share the data with providers specialized in automated document processing at all. This restriction prevents them from training and continuously improving machine learning models.

2. Related Work

A lot of the current progress in the field of document understanding builds upon the combination of spatial and textual information into a common document representation. This is often achieved using grid-based methods, which preserve the 2D layout of the document and directly embed textual information into the representation. The models can use textual information in an embedded form and still take advantage of the 2D correlations of the document. These methods encode the text in embedding vectors and transpose these vectors into corresponding pixels of the grid.

The Chargrid [1] encodes text on character-level. A mapping function assigns an integer value to each character (i.e., alphabetic letters, numeric characters, special characters). The location occupied by a character on the grid will have the corresponding integer value. Before being fed into a deep learning model, the Chargrid is one-hot encoded.

BERTgrid [2] is a special case of the Wordgrid [1]. It uses contextualized word embeddings from BERT [3]. Because BERT acts on a word-piece level, the text in the document needs to be tokenized into word pieces first. A line-by-line serialized version of the document can then be fed into a pre-trained BERT language model.

3. NonDisclosureGrid

Based on the assumption that simplified encodings can replace the original information in documents and still retain utility for model training, we define components to transform the original data into non-sensitive informational pieces. Some components are based on textual information, and some represent purely visual parts of the document.

3.1. Textual Features

State-of-the-art grid-based representations embed the text more or less directly into the grid. Because the character-level encoding and the word or subword embeddings can potentially contain sensitive information, we need to develop other approaches to incorporate textual information.

Layout-only is a binary text mask and the simplest component in our novel representation. This layer contains the information if text is present at the given position in the 2D grid based on the token bounding boxes

SwissText 2022: Swiss Text Analytics Conference, June 08–10, 2022, Lugano, Switzerland

claudio.paonessa@fhnw.ch (C. Paonessa)

Category	Value
Contains alphabetic (a-z, A-Z)	(1, _, _)
Contains numeric (0-9)	(_, 1, _)
Contains non-alphanumeric	(_, _, 1)

Table 1
Definition of the alphanumeric categorization.

detected from OCR. This reduces the document to its spatial layout structure. In Kerroumi et al. [4] this is called *Layout Approach* and uses three channels; i.e., $(1, 1, 1)$ for foreground and $(0, 0, 0)$ for background. Our one-channel version $L \in \mathbb{N}^{H \times W \times 1}$ forms a grid with height H and width W: Let t_k be the k-th token in the page and $b_k = (x_k, y_k, h_k, w_k)$ the associated bounding box of the k-th token

$$L_{ij} = \begin{cases} 1, & \text{if } \exists k \text{ such as } (i, j) \prec b_k \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $\prec$ means the point (i, j) lies within the bounding box b_k (formally: $(i, j) \prec b_k \iff x_k \leq i \leq x_k + w_k \land y_k \leq j \leq y_k + h_k$).

Alphanumeric Categorization is a strongly simplified text encoding. In our approach, we encode a token into a three-dimensional binary vector. These three components are 1 if the token contains at least one alphabetic character, a numeric character, and one other non-alphanumeric character, respectively. This encoding is summarized in Table 1.

The idea behind this approach is that the key information relevant for tasks like information extraction often has consistent underlying character properties. For example, the extraction of monetary values from invoices can be supported if we know which tokens contain numbers; e.g., *65.90* or *23.–* would, with our approach, most of the time be encoded with $(0, 1, 1)$ (no alphabetic but both numeric and non-alphanumeric characters).

Locality-sensitive hashing (LSH) [5, 6] is a family of hashing techniques with a high chance of the hashes of similar inputs being the same. These techniques can be used for data clustering and efficient nearest neighbor search.

One possible implementation is LSH based on hyperplanes. For this, we randomly sample n hyperplanes in the original input space. For each sample in the original space we determine if it's on the left or right of each hyperplane, resulting in a n dimensional boolean vector which forms our hash. We thus reduce word embeddings to a n dimensional binary vector, as every hyperplane randomly splits the embedding space into two categories. The idea behind this hashing is to have textual information without the possibility of reconstructing the original text in the document.

In our experiments we apply this method to BERT embeddings [3] with n set to 10 and 100 respectively.

One could argue that depending on the number of hyperplanes, this hash could enable the reconstruction of the original text. We do not expect this to be an issue with the number of hyperplanes chosen significantly lower than the original embedding dimensions. Nevertheless, this is still an outstanding matter and needs further investigation.

3.2. Visual Features

Visually-rich documents contain valuable information outside of detected textual information. Visual elements are incorporated into documents to increase their readability for humans. Hence, downstream tasks in automatic document processing can benefit from these visual features.

Line mask is a method to incorporate line segments into a one-channel binary mask. A line in a document can be part of a rectangular box around textual elements or a dividing line to separate content or tabular structures. To find lines in document scans, we use the line segment detector implementation from OpenCV [7], which follows the algorithm described in Gioi et al. [8]. To prevent disclosing textual information, we only include lines with a length of at least 10% of the document width. With mathematic rounding, the determined lines are incorporated into a binary mask.

4. Key Information Extraction

The automated extraction of key-value information from document scans such as invoices, receipts, or forms can decrease the manual labor needed for many business workflows. We use this task to compare our novel approach to state-of-the-art representations.

4.1. Datasets

Our work is evaluated on three public datasets covering forms and invoices, the two most common applications for document understanding systems.

FUNSD [9] is an English dataset for form understanding. The dataset contains noisy scanned form pages and consists of 149 training samples and 50 test samples. Each token in the documents is labeled with one of four different classes: *Header, Question, Answer, Other.*

XFUND [10] is a multilanguage form understanding benchmark with matching classes to the FUNSD dataset.

The underlying dataset contains human-labeled forms with key-value pairs in 7 languages: *Chinese, Japanese, Spanish, French, Italian, German, Portuguese*. Because of different character sets we do not use the Chinese and Japanese samples from this dataset. We end up with 745 training samples and 250 test samples.

RVL-CDIP Layout [11] is derived from the RVL-CDIP classification dataset [12] and consists of 520 scanned invoice document pages in English. Each token on a page is labeled with one of 6 different classes. We split the dataset into 416 training samples and 104 test samples. We focus on the fields *Receiver*, *Supplier*, *Invoice info*, and *Total*.

4.2. Model Architecture

We replicate the *chargrid-net* architecture from Katti et al. [1]. This model is a fully convolutional neural network with an encoder-decoder architecture using downsampling in the encoder and a reversion of the downsampling based on stride-2 transposed convolutions in the decoder. In contrast to the two parallel decoders in the original model, we only use the semantic segmentation decoder, which concludes in a pixel-level classification for the number of target classes.

Replicated from the loss used in Katti et al. [1], we counter the strong class imbalance between background pixels and actual relevant pixels with static aggressive class weighting following Paszke et al. [13].

4.3. Evaluation Measure

To evaluate the model performance for the key information extraction task, we use the Word Accuracy Rate (WAR) [1, 4]. Similar to the *Levenshtein distance* it counts the number of substitutions, insertions, and deletions between the ground truth and the prediction. We report the WAR for each given field and overall. The instances are pooled across the entire test set. WAR is defined as follows:

$$\text{WAR} = 1 - \frac{\#[\text{ins.}] + \#[\text{del.}] + \#[\text{sub.}]}{N} \tag{2}$$

where N is the total number of tokens of a specific field in the ground truth instance.

5. Experiments and Results

In the following we show quantitative results of our proposed document representation. First we show the impact of single components by carry out an ablation study followed by a comparison among Chargrid [1] and BERTgrid [2]. As a baseline we use the 3-channel RGB image as input. We report the average metrics over 5-fold cross-validation.

5.1. Ablation Study

We report the results of the ablation study in Figure 1. We experimented with different combinations of our developed components: Layout-only (LA), Alphanumeric Categorization (AN), Locality-sensitive hashing (LSH), and the Line mask (LI). For the FUNSD and XFUND datasets, we additionaly compare LSH components with 10 and 100 hyperplanes, denoted as `LSH(10)` and `LSH(100)`, respectively.

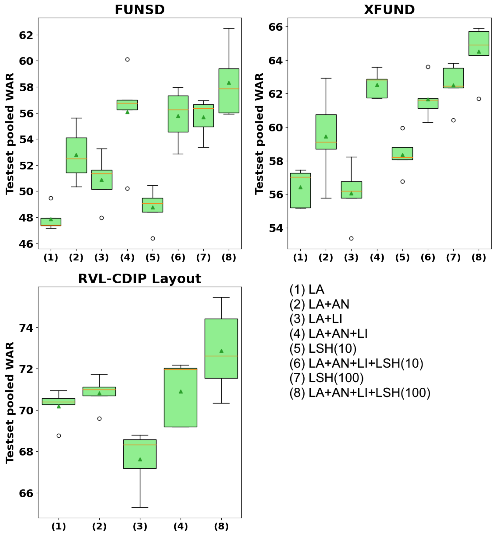

Figure 1: Ablation study to investigate the impact of the different non-sensitive components in the NDGrid. Experiments to estimate impact of LSH (5, 6, 7) are only carried out for FUNSD and XFUND.

5.2. Comparison

We show the quantitative comparison in Table 2, 3 and 4.

Our Chargrid implementation distinguishes between 54 different characters and is case-insensitive. Besides 26 alphabetic characters and the 10 numeric characters, we include 18 additional other characters.

For the BERTgrid, we use the pre-trained BERT base model `bert-base-uncased` from the Hugging Face transformers library [14]. Before being fed into the tokenizer, we order the words by their corresponding bounding boxes' top and left coordinates.

6. Discussion

In the ablation study, we show how we can increase the model performance by combining our non-sensitive components. The impacts of the different components are not consistent over the analyzed datasets. Different dataset

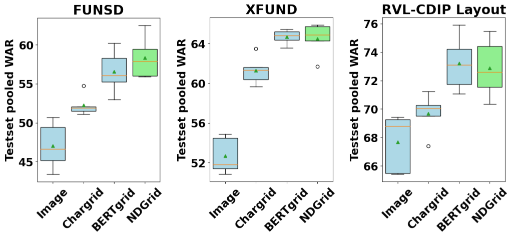

Figure 2: Comparison of model performances between the different document representations.

	All	HED.	QST.	ANS.
[Image]	47.1%	-32.2%	33.3%	26.7%
[Chargrid]	52.3%	**10.8%**	38.5%	31.5%
[BERTgrid]	56.5%	-13.3%	**43.3%**	41.9%
[NDGrid]	**58.3%**	-15.0%	43.0%	**47.1%**

Table 2
Comparison of WAR metrics on FUNSD. Fields: Header (HED.), Question (QST.), Answer (ANS.).

	All	HED.	QST.	ANS.
[Image]	52.7%	-6.8%	20.2%	14.2%
[Chargrid]	61.3%	20.8%	33.1%	30.6%
[BERTgrid]	**64.7%**	**21.7%**	**36.3%**	39.3%
[NDGrid]	64.5%	6.2%	36.1%	**40.0%**

Table 3
Comparison of WAR metrics on XFUND. Fields: Header (HED.), Question (QST.), Answer (ANS.).

	All	REC.	SUP.	INF.	TOT.
[Image]	67.7%	38.5%	19.4%	3.2%	-8.2%
[Chargrid]	69.7%	41.1%	31.0%	4.0%	-5.3%
[BERTgrid]	**73.2%**	43.8%	**41.8%**	**20.9%**	**1.7%**
[NDGrid]	72.9%	**48.8%**	39.3%	14.9%	-6.7%

Table 4
Comparison of WAR metrics on RVL-CDIP Layout. Fields: Receiver (REC.), Supplier (SUP.), Info (INF.), Total (TOT.).

sizes seem to be influential when it comes to the impact of the components. When using the Line mask (LI) in combination with Layout-only (LA) and the Alphanumeric Categorization (AN), the model performance increases significantly. The Line mask (LI) without the Alphanumeric Categorization seems to be less effective or, in the case of the RVL-CDIP Layout dataset, even worse than the Layout-only (LA) by itself. Combining all components, including LSH with 100 hyperplanes, yields the best model performance for all three datasets. The LSH component with 100 hyperplanes does perform worse when not combined with the other components.

Except for the header fields in the FUNSD dataset, the BERTgrid outperforms the image and the Chargrid on all key fields. Our approach often matches and, on specific fields, even outperforms the BERTgrid performance. The results support the hypothesis that a generalized and fundamentally simplified representation still contains enough information to be used in automated document processing.

7. Conclusion

NonDisclosureGrid is a privacy-preserving document representation, including textual, visual, and spatial information. Reducing multimodal information into simplified and non-sensitive encodings is very effective for key information extraction. Our NDGrid produces matching or even better results than models trained with state-of-the-art representations. The performance on other document understanding tasks has yet to be shown, and this work is only the first step towards a versatile privacy-preserving document representation. There is still room for more information-inducing components.

References

[1] A. R. Katti, C. Reisswig, C. Guder, S. Brarda, S. Bickel, J. Höhne, J. B. Faddoul, Chargrid: Towards understanding 2d documents, CoRR abs/1809.08799 (2018). URL: http://arxiv.org/abs/1809.08799. arXiv:1809.08799.

[2] T. I. Denk, C. Reisswig, Bertgrid: Contextualized embedding for 2d document representation and understanding, CoRR abs/1909.04948 (2019). URL: http://arxiv.org/abs/1909.04948. arXiv:1909.04948.

[3] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[4] M. Kerroumi, O. Sayem, A. Shabou, Visualwordgrid: Information extraction from scanned documents using A multimodal approach, CoRR abs/2010.02358 (2020). URL: https://arxiv.org/abs/2010.02358. arXiv:2010.02358.

[5] P. Indyk, R. Motwani, Approximate nearest neighbors: Towards removing the curse of dimensionality, in: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, STOC '98, Association for Computing Machinery, New York, NY, USA, 1998, p. 604–613. URL: https://doi.org/10.1145/276698.276876. doi:10.1145/276698.276876.

[6] A. Gionis, P. Indyk, R. Motwani, Similarity search in high dimensions via hashing, in: Proceedings of the 25th International Conference on Very Large Data

Bases, VLDB '99, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999, p. 518–529.

[7] G. Bradski, The OpenCV Library, Dr. Dobb's Journal of Software Tools (2000).

[8] R. Gioi, J. Jakubowicz, J.-M. Morel, G. Randall, Lsd: A line segment detector, Image Processing On Line 2 (2012) 35–55. doi:10.5201/ipol.2012.gjmr-lsd.

[9] J.-P. T. Guillaume Jaume, Hazim Kemal Ekenel, Funsd: A dataset for form understanding in noisy scanned documents, in: Accepted to ICDAR-OST, 2019.

[10] Y. Xu, T. Lv, L. Cui, G. Wang, Y. Lu, D. Florencio, C. Zhang, F. Wei, Layoutxlm: Multimodal pre-training for multilingual visually-rich document understanding (2021). arXiv:2104.08836.

[11] P. Riba, A. Dutta, L. Goldmann, A. Fornés, O. Ramos, J. Lladós, Table detection in invoice documents by graph neural networks, in: 2019 International Conference on Document Analysis and Recognition (ICDAR), 2019, pp. 122–127. doi:10.1109/ICDAR.2019.00028.

[12] A. W. Harley, A. Ufkes, K. G. Derpanis, Evaluation of deep convolutional nets for document image classification and retrieval, in: International Conference on Document Analysis and Recognition, 2015.

[13] A. Paszke, A. Chaurasia, S. Kim, E. Culurciello, Enet: A deep neural network architecture for real-time semantic segmentation, 2016. arXiv:1606.02147.

[14] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.