FUNSD: A Dataset for Form Understanding in Noisy Scanned Documents

Guillaume Jaume
Swiss Federal Institute of Technology
Signal Processing Laboratory 5
Lausanne, Switzerland
guillaume.jaume@epfl.ch

Hazım Kemal Ekenel
Istanbul Technical University
Department of Computer Engineering
Istanbul, Turkey
ekenel@itu.edu.tr

Jean-Philippe Thiran
Swiss Federal Institute of Technology
Signal Processing Laboratory 5
Lausanne, Switzerland
jean-philippe.thiran@epfl.ch

***Abstract*—In this paper, we present a new dataset for Form Understanding in Noisy Scanned Documents (FUNSD). Form Understanding (FoUn) aims at extracting and structuring the textual content of forms. The dataset comprises 200 fully annotated real scanned forms. The documents are noisy and exhibit large variabilities in their representation making FoUn a challenging task. The proposed dataset can be used for various tasks including text detection, optical character recognition (OCR), spatial layout analysis and entity labeling/linking. To the best of our knowledge this is the first publicly available dataset with comprehensive annotations addressing the FoUn task. We also present a set of baselines and introduce metrics to evaluate performance on the FUNSD dataset. The FUNSD dataset can be downloaded at https://guillaumejaume.github.io/FUNSD/.**

***Keywords*-Text detection; Optical Character Recognition; Form Understanding; Spatial Layout Analysis**

I. Introduction

Forms are a common way of collecting data. They are used in many different fields, from the medical domain to administrative systems. We define Form Understanding (FoUn) as the task of automatically extracting and structuring the written information in a form. FoUn is built on top of text detection and recognition. Firstly, it analyzes the spatial layout and written information in order to identify the questions, answers and headers present in the form. Secondly, it aims to understand how the extracted entities are linked to each other. In this paper, we introduce the FUNSD dataset, a dataset for Form Understanding in Noisy Scanned Documents. To the best of our knowledge FUNSD is the first publicly available dataset that aims to address the FoUn task. The FUNSD dataset contains 200 fully annotated forms that exhibit high variability in their structures and representations. The forms come from different domains, *e.g.,* marketing, advertisement, scientific reports. All the forms are on one page and come in a rasterized format. They have a low-resolution and are corrupted by real noise. The forms were annotated in a bottom-up approach allowing the FUNSD dataset to be used for various document understanding tasks including text detection, text recognition, spatial layout understanding, question-answer pair extraction, etc.

Extracting information from scanned documents is not a new task. For instance, previous work focused on digitization the content of documents into a machine-readable format using optical character recognition (OCR). We refer to [1], [2] for a review of current OCR systems. Existing datasets include the ICDAR Robust Reading Competitions 2011, 2013, 2015, 2017 . Another task of information extraction from document is layout analysis that tries to extract the content of a document and restore its structure by analyzing its spatial arrangement. Applications of layout analysis range from text and non-text separation to full text segmentation of complex layouts [3]–[7]. An application closely related to FoUn is table understanding [8], [9]. In this case, the goal is to retrieve the key-value pairs that map headers from a table to the value represented by a cell. However, tabular structure is quite a rigid constraint that is not as general as the forms.

Commercial solutions like ABBYY[1], Nuance[2] or Datacap[3] allow information extraction from user-defined areas in specific pages of documents, including forms. This requires manual annotation of zones where the answer is expected to appear. However these solutions do not scale well when the number of templates increases. On the contrary, the FUNSD dataset was created to build template-agnostic representations of the forms. Moreover, FoUn goes beyond the aforementioned approaches, and aims to extract structured information in a semantically meaningful way so that, for instance, it can be stored in a database, which can be used for data analysis.

Our contributions can be summarized as follows:

- We formalize Form Understanding as a series of defined tasks. From an image of a form, we define a pipeline to structure the textual content as a list of labeled semantic entities that are linked to each other.
- We provide access to the FUNSD dataset, a document understanding dataset for text detection, OCR, spatial

[1] https://www.abbyy.com/
[2] https://www.nuance.com/print-capture-and-pdf-solutions.html
[3] https://www.ibm.com/ch-fr/marketplace/document-capture-and-imaging

layout analysis and entity linking in noisy scanned forms.

- We build a set of baselines that define the current state-of-the-art results for the FUNSD dataset.
- We propose a set of metrics to evaluate the Form Understanding pipeline.

II. Dataset Description

A. A subset of the RVL-CDIP dataset

To ensure that real data are used, with a high variability in the structure of the forms and realistic noise, we used a subset of the RVL-CDIP dataset[4] [10]. The RVL-CDIP dataset is composed of $400,000$ grayscale images of various documents from the 80's-90's. Each image is labeled by its type, *e.g.,* letter, email, magazine, form. The documents have a low resolution, around 100 dpi. The images are also of low quality with various types of noise added by successive scanning and printing procedures. To build the FUNSD dataset, we manually checked the $25,000$ images from the form category. We discarded unreadable and similar forms resulting in $3,200$ eligible documents, out of which we randomly sampled 200 to annotate. Note that the RVL-CDIP dataset is a subset of the Truth Tobacco Industry Document[5] (TTID), an archive gathering scientific research, marketing, advertising documents of the largest US tobacco firms. The TTID archive aims to advance information retrieval research.

B. Annotation procedure

The annotations used for text detection were performed by Figure8 mechanical turks[6]. The remaining tasks were annotated using an annotation tool specifically designed for the form understanding. The annotation tool is based on GuiZero[7], a high-level library built on top of tkinter[8].

C. Dataset structure and format

Each form is encoded in a JSON file. We represent a form as a list of semantic entities that are linked to each other. A semantic entity represents a group of words that belong together from a semantic and spatial standpoint. Each semantic entity is composed of a unique label (*i.e.,* question, answer, header or other), a bounding box, a list of links with other entities and a list of words. Each word is represented by its textual content and its bounding box. All the bounding boxes are represented by their coordinates following the schema box $= [\mathbf{x}_{left}, \mathbf{y}_{top}, \mathbf{x}_{right}, \mathbf{y}_{bottom}]$. The links are directed and formatted as $[\mathbf{id}_{from}, \mathbf{id}_{to}]$, where $\mathbf{id}$ represents the semantic entity index w.r.t the list of semantic entities. The dataset statistics are shown in Figure I. Even with a limited number of annotated documents, we obtain a large number of word-level annotations ($>$ 30k) and entities ($\approx$ 10k) making this dataset suitable for deep learning applications. The semantic entity class distribution is shown in Figure II. Naturally, the most common classes are questions and answers.

[4]https://www.cs.cmu.edu/ aharley/rvl-cdip/
[5]https://www.industrydocuments.ucsf.edu/tobacco/
[6]https://www.figure-eight.com/
[7]https://lawsie.github.io/guizero/
[8]http://tkinter.fdex.eu/

Listing 1: Example of ground truth format.

```
{
    "form": [
        {
            "text": "Registration No.",
            "box": [94,169,191,186],
            "linking": [
                [1,2]
            ],
            "label": "question",
            "words": [
                {
                    "text": "Registration",
                    "box": [94,169,168,186]
                },
                {
                    "text": "No.",
                    "box": [170,169,191,183]
                }
            ]
        },
        {
            "box": [209,169,236,182],
            "text": "533",
            "label": "answer",
            "words": [
                {
                    "box": [209,169,236,182
                    ],
                    "text": "533"
                }
            ],
            "linking": [
                [1,2]
            ]
        }
    ]
}
```

Table I: Dataset statistics.

Split	Forms	Words	Entity	Relations
Training	149	22512	7411	8472
Testing	50	8973	2332	2152

Table II: Class distribution of the semantic entities.

Split	Header	Question	Answer	Other	Total
Training	441	3266	2802	902	7411
Testing	122	1077	821	312	2152

An example of ground truth file is shown in Listing 1. The corresponding sub-part of the original form is shown in Figure 1. In this example, we have two semantic entities, *"Registration No."* that is tagged as question and *"533"* tagged as answer. There is a link going from the first semantic entity to the second one resulting in a question-answer pair.


Registration No. 533

Figure 1: Screenshot form from the FUNSD dataset.

III. Baselines and Metrics

We present baseline results for text detection, text recognition and form understanding on the FUNSD dataset.

A. Text detection

We test the text detection at the word level. State-of-the-art algorithms follow a data-driven approach. Usually, CNN-feature maps are extracted using a deep neural network. The network then predicts heat-maps that represent the probability of each pixel being part of a text jointly with bounding box proposals [11]–[14].

The text detection on the FUNSD dataset was tested with 4 baselines: Tesseract [15], EAST [11][9], Google Vision API[10] and with a Faster R-CNN architecture [16]. Tesseract, EAST and Google Vision are tested without re-training on the FUNSD training set. As EAST and Google Vision output their predictions as quadrangles (*i.e.,* 4 vertices that define a polygon) and the FUNSD dataset is annotated with rectangles, we transform each quadrangle as a rectangle by constructing the smallest rectangle that contains the 4 quadrangle vertices. The Faster R-CNN baseline is based on a PyTorch implementation[11] that was retrained specifically for this task. We used a pre-trained network trained on ImageNet with a ResNet-101 architecture [17]. We used anchors with size $(16, 32, 64, 128, 256)$, strides $(4, 8, 16, 32, 64)$ and aspect ratios $(0.5, 1.0, 2.0, 4.0, 8.0)$. During testing, we allow for maximum 500 object detections and select all objects with confidence detection 0.5. The learning rate was set to 10^{-3} with a weight decay of 0.0001. The batch size was set to 1 and the maximum number of epochs to 10 with early stopping. For each approach, we compute on the FUNSD test set the precision, recall and F1-score at IoU $= 0.5$. Results are shown in Table III.

Table III: Results for word-level text detection. Precision and recall expressed in %.

Method	Precision	Recall	F1-score
Tesseract	45.4	68.0	0.54
EAST	51.6	84.0	0.64
Google Vision	**79.8**	62.0	0.69
Faster R-CNN	70.4	**84.8**	**0.76**

The Faster R-CNN baseline is giving the best overall performance (*i.e.,* highest F1-score). This observation is expected as we are specifically retraining the network for the task. Note that the Google Vision is still performing well, even without being re-trained on the task showing its generalization power. Tesseract, based on more ad-hoc text detection algorithms, is the worst performer.

B. Text recognition with Optical Character Recognition

OCR engines are usually based on appearance features to obtain a character level prediction that is coupled with a sequence modeling network (*e.g.,* LSTM, GRU) to extract the words [18]. Modern engines that also support handwritten text recognition usually use a Connectionist Temporal Classification (CTC) loss to cope with the alignment problem [18]. Note that some novel architectures perform the text detection and recognition in an end-to-end manner [19].

We evaluate the relevance of the OCR output by computing the Levenshtein similarity between the predicted word w_p and the ground truth word w_{gt}:

$$S(w_p, w_{gt}) = 1 - \frac{L(w_p, w_{gt})}{\max(|w_p|, |w_{gt}|)} \tag{1}$$

where $L(w_p, w_{gt})$ is the Levenshtein distance between w_p and w_{gt} and $|.|$ denotes the number of characters in a word. The similarity is case-sensitive and is taking into account the recognition of checkboxes (often encountered in documents like forms). We evaluate two OCR engines for text recognition: Tesseract [15] and Google Vision. We evaluate the OCR performance using two metrics, referred as Text detection + OCR and OCR. We compute in both cases the Levenshtein similarity between the correctly detected words and the ground truth (*i.e.,* IoU > 0.5). In the first case we normalize by the total number of ground truth words, whereas in the second case we normalize by the number of *identified* words. Note that no preprocessing is applied to the documents before being fed to the OCR.

Table IV: OCR results based on Levenshtein similarity. Results expressed in %.

Method	Text detection + OCR	OCR
Tesseract	3.4	7.3
Google Vision	**76.4**	**94.4**

From Table IV, we observe that Google Vision is a really strong OCR baseline that captures almost perfectly the textual content when the words are correctly identified ($\approx$ 95%). Tesseract OCR engine performs poorly on the FUNSD dataset. This can be explained by the fact that the minimum quality of 300 dpi needed by Tesseract is not met in the FUNSD dataset.

C. Form Understanding

We decompose the FoUn challenge in 3 tasks, namely the word grouping, the semantic entity labeling and finally the entity linking.

- **Word grouping** is the task of aggregating words that belong to the same semantic entity.

[9]https://github.com/argman/EAST

[10]https://cloud.google.com/vision/docs/detecting-fulltext

[11]https://github.com/facebookresearch/maskrcnn-benchmark

- **Semantic entity labeling** is the task of assigning to each semantic entity a label from a set of 4 pre-defined categories: question, answer, header or other.
- **Entity linking** is the task of predicting the relations between semantic entities.

Figure 2 illustrates this idea by showing the word grouping and labeling in a form from the FUNSD dataset.

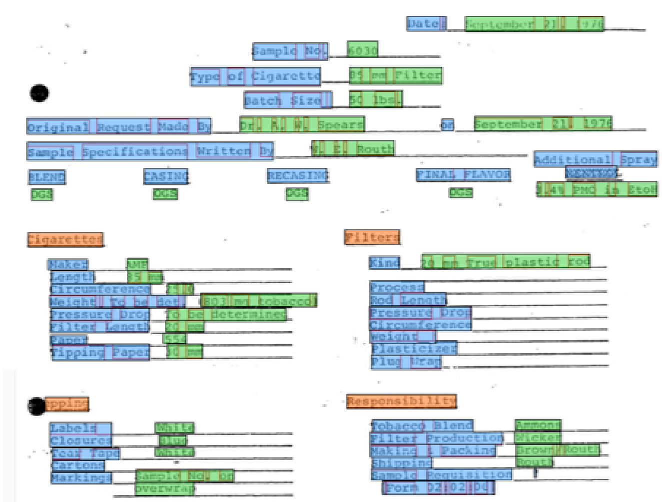

Figure 2: Example of word grouping and labeling on a form from the FUNSD dataset. *Questions* are represented in blue, *headers* in orange and *answers* in green.

1) Word grouping: We tested the word grouping on two naive baselines based on textline extraction performed by Tesseract and Google Vision OCR engines. We propose to evaluate the word grouping as a clustering problem, where the data points are the words and the clusters are the semantic entities. The optimal number of clusters is the number of semantic entities in the ground truth. All the words that were not recognized by the text detector (*i.e.,* IoU < 0.5), are assigned to a new artificial cluster. We propose to use the adjusted rand index [20] (ARI) as metric. The ARI is based on the number of pairs correctly assigned to the same cluster adjusted to compensate randomness. The results are presented in Table V. As expected, the baselines perform poorly as they do not take into consideration the spatial layout and the textual content. We foresee the need of *learned* algorithms to group the words to build more competitive algorithms.

Table V: Baseline results for the word grouping. A value of 0 corresponds to a random assignment and 1 to a perfect clustering.

Method	Word grouping
Tesseract	0.20
Google Vision	**0.41**

2) Semantic entity labeling: We propose to use a simple learned neural baseline based on a multi-layer perceptron. We build input features for each semantic entity with:

- semantic features extracted from the pre-trained language model BERT [21][12],
- spatial features based on the bounding box coordinates of the semantic entity,
- meta features that encode the length of the sequence.

The resulting input feature dimension for each entity is 733. Each semantic entity is then independently passed through an MLP with 2 hidden layers and 500 units each with ReLu activation. The last layer is a softmax classifier to derive the class label. Note that we are testing the algorithms by assuming that we know the optimal word grouping, word location and textual content. In this way, we *only* assess the specific task. Results are shown in TableVI.

Table VI: Baseline results for the entity labeling and linking. Precision and recall expressed in %.

Task	Precision	Recall	F1-score
Entity labeling	–	–	0.57
Entity Linking	2.1	99.2	0.04

3) Entity linking: We re-use the semantic entity input features built for the entity labeling task. We approach the entity linking task as a binary classification task (*i.e.,* whether or not a link exists). We simply concatenate the feature representation of each semantic entity for all the possible pairs in the form. We then pass it through a MLP with 2 hidden layers and 500 hidden units with ReLu activation.

The metric used verifies if the predicted links exist among all the semantic entities correctly identified and labeled. We can then compute the precision, recall and F1-score. Note that not all the semantic entities have relations with other semantic entities (e.g., a sentence describing the page number of the form or an unanswered question). Results are presented in Table VI. Stronger baselines should include the relational side of semantic entities that can naturally be represented as a graph.

IV. Conclusion

We introduced a new dataset FUNSD, for Form Understanding in Noisy Scanned Documents along with a set of simple baselines and metrics to evaluate the FoUn. We hope that this work can be the starting point of advances in the domain of document understanding. Approaches to address the form understanding challenge include the development of a neural end-to-end pipeline that given a set of words, jointly learn how to group them, assign a label and build relations between them.

[12]https://github.com/huggingface/pytorch-pretrained-BERT

References

[1] N. Islam, Z. Islam, and N. Noor, "A Survey on Optical Character Recognition System," *J. Inf. Commun. Technol.*, 2017.

[2] A. P. Tafti, A. Baghaie, M. Assefi, H. R. Arabnia, Z. Yu, and P. Peissig, "OCR as a service: an experimental evaluation of google docs OCR, tesseract, ABBYY finereader, and transym," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10072 LNCS. Springer, Cham, 2016, pp. 735–746.

[3] C. Clausner, A. Antonacopoulos, and S. Pletschacher, "ICDAR2017 Competition on Recognition of Documents with Complex Layouts - RDCL2017," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, vol. 1. IEEE, nov 2018, pp. 1404–1410. [Online]. Available: http://ieeexplore.ieee.org/document/8270160/

[4] S. Eskenazi, P. Gomez-Krämer, and J.-M. Ogier, "A comprehensive survey of mostly textual document segmentation algorithms since 2008," *Pattern Recognition*, vol. 64, pp. 1–14, apr 2017. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0031320316303399

[5] S. Mao, A. Rosenfeld, and T. Kanungo, "Document structure analysis algorithms: a literature survey," T. Kanungo, E. H. Barney Smith, J. Hu, and P. B. Kantor, Eds., vol. 5010. International Society for Optics and Photonics, jan 2003, pp. 197–207. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=755961

[6] S. Marinai, "Chapter 16 Learning Algorithms for Document Layout Analysis," in *Handbook of Statistics*, 2013, vol. 31, pp. 400–419.

[7] 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," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017, pp. 4342–4351.

[8] M. Y. Akpinar, E. Emekligil, and S. Arslan, "Extracting table data from images using optical character recognition text," in *2018 26th Signal Processing and Communications Applications Conference (SIU)*. IEEE, may 2018, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/document/8404746/

[9] W. Farrukh, A. Foncubierta-Rodríguez, A.-N. Ciubotaru, G. Jaume, C. Bekas, O. Goksel, and M. Gabrani, "Interpreting Data from Scanned Tables," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 2. IEEE, nov 2017, pp. 5–6. [Online]. Available: http://ieeexplore.ieee.org/document/8270192/

[10] A. W. Harley, A. Ufkes, and K. G. Derpanis, "Evaluation of deep convolutional nets for document image classification and retrieval," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, vol. 2015-Novem, 2015, pp. 991–995.

[11] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "EAST: An Efficient and Accurate Scene Text Detector," in *Computer Vision and Pattern Recognition, IEEE Conference on (CVPR)*, 2017, pp. 5551–5560.

[12] Z. Tian and W. Huang, "Detecting Text in Natural Image with Connectionist Text Proposal Network," *European Conference on Computer Vision (ECCV)*, pp. 1–16.

[13] Z. Huang, Z. Zhong, L. Sun, and Q. Huo, "Mask R-CNN with Pyramid Attention Network for Scene Text Detection," 2018. [Online]. Available: http://arxiv.org/abs/1811.09058

[14] Y. Dai, Z. Huang, Y. Gao, Y. Xu, K. Chen, J. Guo, and W. Qiu, "Fused Text Segmentation Networks for Multi-oriented Scene Text Detection," 2017. [Online]. Available: http://arxiv.org/abs/1709.03272

[15] R. Smith, "An overview of the tesseract OCR engine," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, vol. 2, 2007, pp. 629–633.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem. IEEE Computer Society, dec 2016, pp. 770–778.

[18] F. Borisyuk, A. Gordo, and V. Sivakumar, "Rosetta: Large Scale System for Text Detection and Recognition in Images," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '18*, 2018, pp. 71–79. [Online]. Available: https://doi.org/10.1145/3219819.3219861

[19] W. Sui, "A Novel Integrated Framework for Learning both Text Detection and Recognition," *ICPR*, pp. 2233–2238, 2018.

[20] L. Hubert, "Classification ˜1985," vol. 218, no. 1980, pp. 193–218, 1985.

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.