

# RoBERTa Ensemble Technique for Document Information Localization and Extraction

Notebook for the DocILE Lab at CLEF 2023

Bao Gia Tran<sup>1</sup>, Duy-Ngo Minh Bao<sup>1</sup>, Khanh Gia Bui<sup>1</sup>, Huy Viet Duong<sup>1</sup>,  
Dang Hai Nguyen<sup>1</sup> and Hieu Minh Nguyen<sup>1</sup>

<sup>1</sup>University of Information Technology - VNUHCM, Ho Chi Minh City, Vietnam

## Abstract

Document Information Localization and Extraction (DocILE) is attracting a large amount of attention from the research community due to its potential to significantly reduce manual work. With the explosive growth of technology as they are today, we want to experiment with a method that leverages the advantages of language models in information extraction since it requires an understanding of the contextual information of the text, which large language models are currently successful on. The experiments include using a new combination of published baseline with our model *RoBERTa*, along with a post-processing step, which helped us achieve a Top 3 position in the competition ranking board.

## Keywords

DocILE, RoBERTa, Ensemble, Pseudo-Labeling

## 1. Introduction

Extracting information from documents is an indispensable part of human activities in the modern era. However, manual information extraction is time-consuming and labor-intensive. Therefore, automating the process of extracting information has gained much attention from the research community as it has high applicability in reducing workload for workers in manual tasks and creating opportunities for them to focus more on strategic work.

The information extraction process is challenged since it requires an understanding of the semantics, layout, and context of content in the documents. In Machine Learning (ML), the scope of addressing this issue is called Document Information Extraction (IE), a part of Document Understanding.

DocILE 2023 [1][2] is a competition for extracting information from business documents. The participating teams will receive a dataset of invoice-like documents such as tax invoices, orders, purchase orders, receipts, sales orders, proforma invoices, credit notes, utility bills, and debit notes [1]. In Track 1, also known as KILE (Key Information Localization and Extraction), participants were challenged to develop algorithms that can locate and extract specific information


---

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece

✉ 22520121@gm.uit.edu.vn (B. G. Tran); 22520320@gm.uit.edu.vn (D. M. Bao); 22520630@gm.uit.edu.vn (K. G. Bui); 22520540@gm.uit.edu.vn (H. V. Duong); 22520189@gm.uit.edu.vn (D. H. Nguyen); 22520440@gm.uit.edu.vn (H. M. Nguyen)

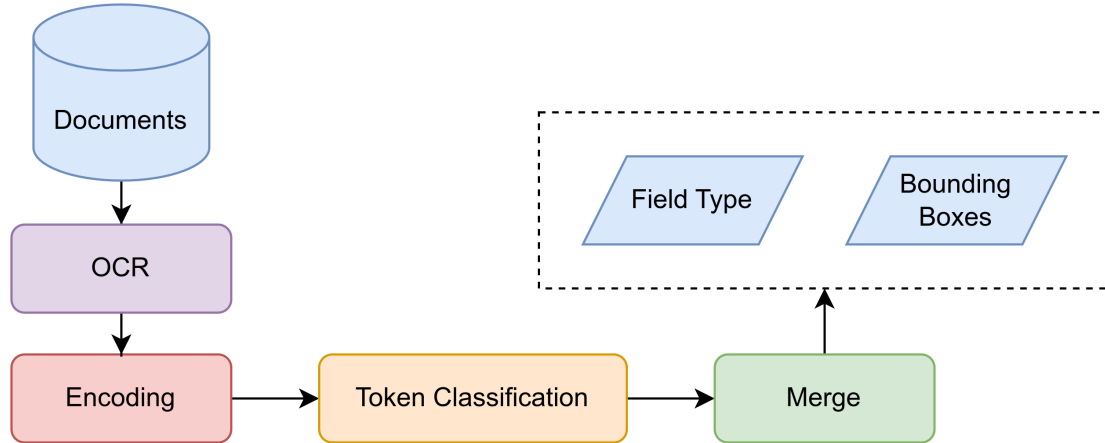
🌐 <https://github.com/xbaotg> (B. G. Tran)

© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

such as names, dates, addresses, .. or any other key data from a given document.

The pipeline to localize Key Information and extract them is built upon the provided baselines for the DocILE competition, and we acknowledge their contributions [1]. At first, the input data is a set of PDF pages containing invoices processed using DocTR [3], from which the bounding boxes and content of the information are obtained. Afterward, they classify those content using Token Classification models. Finally, they merge content based on their field type. The flowchart is shown in Figure 1.



**Figure 1:** Pipeline of localizing Key Information and extracting them

Our focus lies in optimizing their pipeline. Specifically, we leverage different versions of *RoBERTa* - a large language model that is used to achieve state-of-the-art results on GLUE, RACE, and SQuAD in Natural Language Processing [4], i.e. two provided baseline RoBERTa [5] and one RoBERTa trained by us - together with a post-processing step. After that, we use it to generate pseudo-label datasets from provided unlabeled dataset and re-train our models on that, which showed a relatively good result. Our pipeline is shown in Figure 2. We will discuss each component in detail in the following sections.

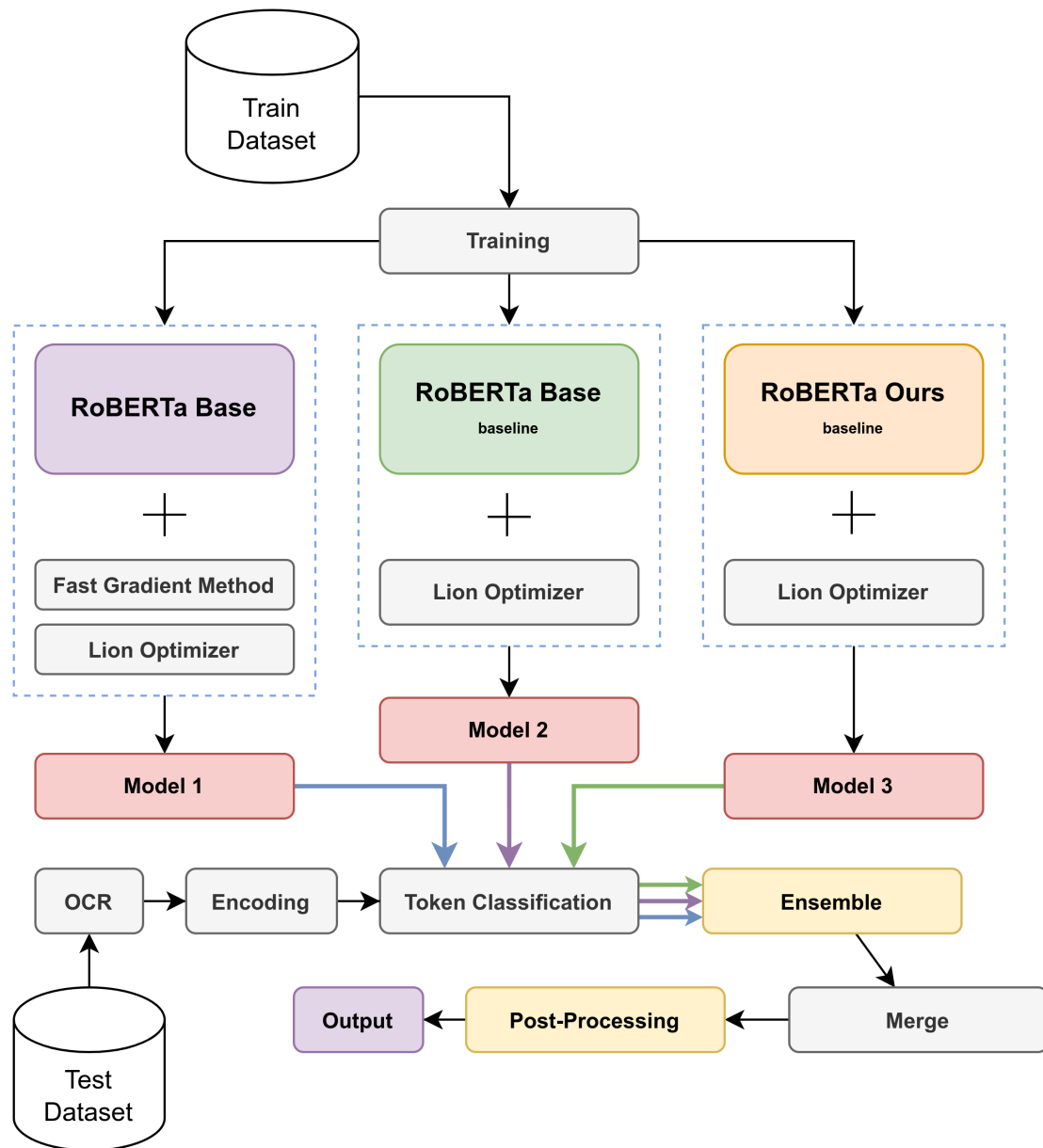
## 2. Proposed method

### 2.1. Ensemble

After evaluating several models provided, we see that one model only performs well in certain categories while the opposite thing happens for the other models. This leads us to the idea of using Ensemble [6].

We first implement Ensemble using Average and Max Voting [6] since they are two of the most common methods. However, the results acquired show a relatively high score precision while the recall is not significant, which means that the models provide fairly accurate predictions but the proportion of positive samples missed in the dataset is quite large.

From the concept of affirmative ensemble presented in prior research [7], we decided to incorporate this concept into our problem as a method to address the aforementioned issues.



**Figure 2:** Pipeline of our approach to this competition.

Specifically, if any of the models predict that certain content belongs to a particular field type, we consider that content to actually belong to that field type.

## 2.2. Pseudo-labeling

In this task, we are provided an abnormally huge amount of unlabeled data compared to the small amount of labeled data [1]. We believe that utilizing this unlabeled data will improve the

performance of the models. This leads us to the idea of using semi-supervised learning methods.

Pseudo-Labeling [8] is a more effective method compared to other methods such as [9], [10], and [11]. We propose a different way to implement this technique for our models:

1. Models will be trained on the Train (annotated) dataset.
2. We use the Ensemble technique with Post-processing to predict labels for the unlabeled dataset, which is then called as pseudo-labeled dataset.
3. Train the model on the pseudo-labeled dataset for some epochs.
4. Fine-tune the model on the Train dataset.

Here, we train the model on the pseudo-labeled dataset instead of mixing it with the labeled dataset, because it is a dataset that we have little control over, and it may contain cases that are completely different from the training dataset. Training the model on the pseudo-labeled dataset for some epochs helps the model approach more types of data, thereby learning general features. Then, we fine-tune the model on the training dataset to learn the correct features for each specific problem, helping the model improve its effectiveness on that problem.

### 2.3. Post processing

The models we use struggle in distinguishing information that have the same field type but is a bit far apart. This leads to the problem that even though the information belongs to the same field type and the same bounding box, the model predicts it as multiple different bounding boxes. Moreover, after observing the prediction results compared to the ground truth, and experimenting on various documents, we found that it is rare for information of the same field type to be close to each other on the same document.

We first find the distance between the centers of each pair of bounding boxes belonging to the same field type predicted by the model. Then, we experiment with grouping these bounding boxes on different thresholds. For each pair of bounding boxes of information belonging to the same field type, if their Euclidean distance is below or equal to the threshold, we will merge those two bounding boxes into a new one, its coordinate is calculated by using formula (1). How the Post-processing work is shown in Figure 3.

$$final = (\min(x_{left}; x'_{left}), \min(y_{top}; y'_{top}), \max(x_{right}; x'_{right}), \max(y_{bottom}; y'_{bottom})) \quad (1)$$

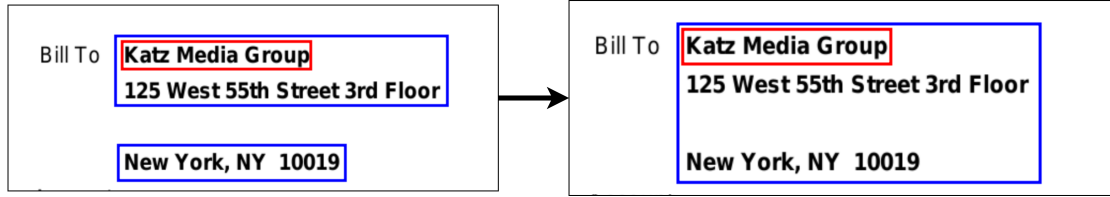
By doing this, we can reduce the number of false bounding boxes predicted by the model and improve the accuracy of the predictions.

## 3. Experiment

### 3.1. Dataset

We maintain the same dataset partitioning as provided by the organizer, with the information of each dataset used for the Training, Validating, and Testing processes as described in Table 1.

Most experiments below were conducted on an environment consisting of 4 RTX 2080 Ti 12GB GPUs, along with the following selected parameters and hyperparameters:



**Figure 3:** Post-Processing merges two bounding boxes that have the same field type (same border color) and their distance is below a threshold into one bounding box.

**Table 1**

Number of documents and annotations in each dataset

Dataset	Document Number	Annotation Number
Train	5,180	65,651
Validation	500	5,862
Test	1,000	-
Synthetic	100,000	1,117,000
Unlabeled	932,000	-

- Train batch size = 4
- Test batch size = 4
- Gradient Accumulation Steps = 4
- Weight decay = 0.01
- Data Loader workers = 32
- Training Epoch = 500
- Learning rate = 1e-5

At the same time, we use the validation set to evaluate the model and compare the performance between different models based on the results.

### 3.2. Model

We use 3 *RoBERTa* models, 2 published baseline models [5] trained on the Synthetic + Train dataset, and the remaining model is taken directly from HuggingFace [12]. We will refer to these models with different names for easy distinction as follows:

- *RoBERTa\_DOCILE\_BASE*: baseline *roberta\_base\_with\_synthetic\_pretraining*
- *RoBERTa\_DOCILE\_OURS*: baseline *roberta\_ours\_with\_synthetic\_pretraining*
- *RoBERTa\_OURS*: the model was not trained on any DocILE dataset.

### 3.3. Fast Gradient Method

For *RoBERTa\_OURS*, we fine-tune it from *RoBERTa* [12] using the Fast Gradient Method (FGM) [13] technique on the Synthetic dataset with 30 epochs and on the Train dataset with 500 epochs. The result obtained as shown in Table 2, which is similar to the baseline but this

technique helps the model to be more generalized [13], so we still keep and apply it with other methods.

**Table 2**

Result obtained from training *RoBERTa\_OURS* with Fast Gradient Method

Model	AP
<i>RoBERTa_OURS</i>	0.562
<i>RoBERTa_DOCILE_OURS</i>	0.557
<i>RoBERTa_DOCILE_BASE</i>	0.566

### 3.4. Lion Optimizer

For all 3 models, we replaced the default optimizer, from AdamW to Lion Optimizer [14] and trained for an additional 300 epochs on the Train dataset. However, only *RoBERTa\_DOCILE\_OURS* showed significant improvement, while the other models are mostly unchanged. Nevertheless, we will still use Lion Optimizer for the methods below because it seems to converge much faster than AdamW. Table 3 shows the results obtained.

**Table 3**

Result obtained from training 3 *RoBERTa* with Lion Optimizer

Model	AP
<i>RoBERTa_OURS</i>	0.562
<i>RoBERTa_DOCILE_OURS</i>	0.566
<i>RoBERTa_DOCILE_BASE</i>	0.565

### 3.5. Ensemble

We perform Ensemble using different methods:

- Average Ensemble [6]
- Max-Voting Ensemble [6]
- Affirmative Ensemble [7]

For each method, we ensemble the following models:

- *RoBERTa\_OURS* trained with FGM technique and Lion Optimizer
- *RoBERTa\_DOCILE\_OURS* trained with Lion Optimizer
- *RoBERTa\_DOCILE\_BASE* without any changes

Table 4 demonstrates that Affirmative Ensemble produces significantly better results compared to commonly used methods like Max-Voting and Average. Therefore, we will employ the Affirmative Ensemble technique on our three models to predict the output.

**Table 4**Result obtained from Ensembling 3 *RoBERTa* models

Method	AP
Average	0.580
Max-Voting	0.576
Affirmative	0.607

### 3.6. Post-Processing

We performed the Post-Processing method mentioned in Section 2.3, on the predicted output of each of the 3 models with different percentage thresholds of the document width. E.g. For a document with a width is 2000px, the 14.5% threshold means that it will merge two bounding boxes whose distance is below 14.5% of 2000px or 290px.

**Table 5**

Result obtained from changing post-processing threshold, i.e. the distance between the two central bounding boxes relative to the width of the document

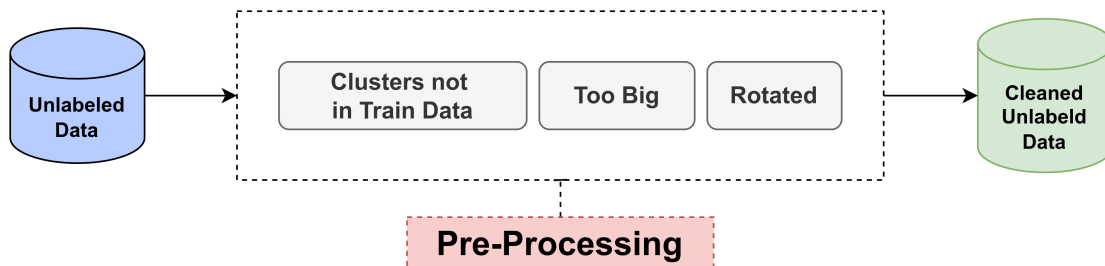
Threshold (%)	AP		
	<i>RoBERTa</i> _OURS	<i>RoBERTa</i> _DOCILE_OURS	<i>RoBERTa</i> _DOCILE_BASE
12	0.606	0.603	0.606
13	0.606	0.604	0.607
14	0.606	0.604	0.607
14.5	0.607	0.605	0.608
15	0.607	0.603	0.607
16	0.603	0.599	0.603
17	0.598	0.596	0.603

As shown in Table 5, the threshold of 14.5% of the document width gives the highest result when evaluated on the validation set. From now on, we will use 14.5% as the default threshold. Combining this Post Processing method with the methods we mentioned in Section 2 significantly improves the results.

### 3.7. Pseudo-Labeling

Due to the large number of documents, we will do this technique on each chunk from provided chunks dataset. Starting with the chunk<sub>0</sub> dataset, we pre-process it as follows:

- Remove documents belonging to clusters = -1, i.e., documents whose layouts do not appear in the Train dataset.
- Remove documents too big, i.e. documents with a size larger than 3000 pixels in any dimension.
- Remove rotated documents.



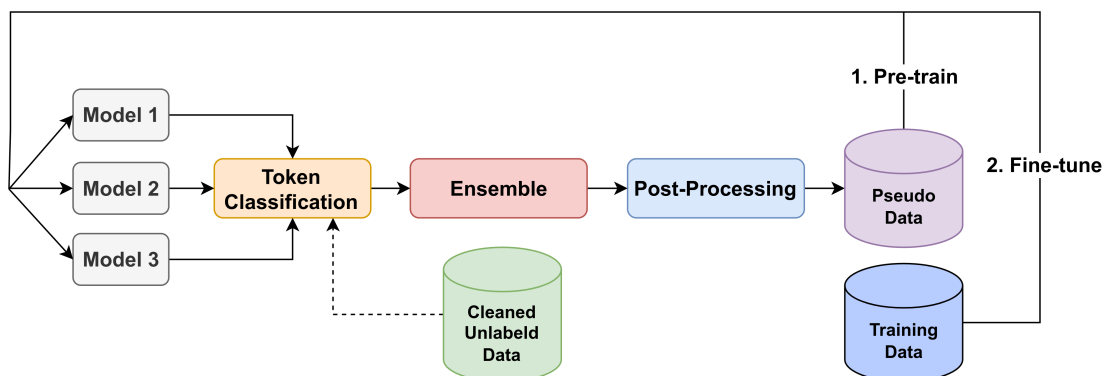
**Figure 4:** Pipeline of pre-processing unlabeled dataset.

We then ensemble the 3 models trained on the Train dataset, combining with the Post Processing. Afterward, we predict on the unlabeled chunk<sub>0</sub> dataset to generate pseudo annotations for that, which we call pseudo<sub>0</sub>. After that, we train 3 models on this dataset with the following hyperparameters:

- Epoch: 30
- Learning Rate: 1e-5

Later, we use the Train dataset to train all 3 models more with the following hyperparameters:

- Epoch: 300
- Learning rate: 5e-6



**Figure 5:** Pipeline of generating pseudo-labeled dataset from trained models

The addition of Pseudo-Labeling slightly improved our results as shown in Table 6. However, this method was implemented when the competition was in its final days, which only allowed us to perform it on one chunk of data. Nevertheless, we believe that continuing to use the remaining chunks will continue to improve the final results.



**Table 6**

Result obtained from training models with Pseudo-label

Model	AP	
	Train Data	+ chunk <sub>0</sub>
<i>RoBERTa_OURS</i>	0.562	0.568
<i>RoBERTa_DOCILE_OURS</i>	0.566	0.57
<i>RoBERTa_DOCILE_BASE</i>	0.566	0.576

## 4. Result

Table 7 shows our performance on the validation dataset of each model when combined with Fast Gradient Method, Lion Optimizer, and Pseudo-Labeling technique. The value of *RoBERTa\_OURS* in the Baseline column indicates the result of training RoBERTa with the Fast Gradient Method technique.

**Table 7**

Performance of different models on the DocILE competition task

Model	AP		
	Baseline	+ Lion Optimizer	+ chunk <sub>0</sub>
<i>RoBERTa_OURS</i>	0.562	0.562	0.568
<i>RoBERTa_DOCILE_OURS</i>	0.557	0.566	0.57
<i>RoBERTa_DOCILE_BASE</i>	0.566	0.565	0.576

Table 8 shows our performance of 3 models with the Ensemble Method, they are trained with and without the Pseudo-Labeling technique. From the predicted result, we do Post-Processing and evaluate them on the validation dataset.

**Table 8**

Performance of different models on the DocILE competition task

Model	Method	AP	
		+ Ensemble	+ Post-Processing
3 <i>RoBERTa</i>	-	0.608	0.644
	+ Pseudo-Labeling	0.612	0.648

Overall, our results increased significantly compared to the baseline, **+0.082** on the Valset and **+0.073** on the Testset, which is shown in Table 9. However, we believe there are still many things we can do to further improve the results:

- Use more unlabeled data. Currently, only a very small fraction (10k out of almost 1M) of the unlabeled data was used.
- Use models incorporating layout features such as LayoutLMv3, LiLT, etc.

**Table 9**

Performance of different models on the DocILE competition task

	Baseline	Ours	Change
Valset	0.566	0.648	<b>+0.082</b>
Testset	0.539	0.612	<b>+0.073</b>

## 5. Conclusion

In this paper, we have presented a solution for the tasks required in Track 1 KILE of DocILE 2023. Our improvements to the baseline [5] have demonstrated their effectiveness. This result is significantly higher than the initial performance and demonstrates the potential of our method in addressing issues related to information extraction from business documents. We hope that our solution will contribute to the development of the field of information extraction from business documents, and we look forward to further researching and improving our method in the future.

## References

- [1] Š. Šimsa, M. Uříčář, M. Šulc, Y. Patel, A. Hamdi, M. Kocián, M. Skalický, J. Matas, A. Doucet, M. Coustaty, D. Karatzas, Overview of DocILE 2023: Document Information Localization and Extraction, in: A. Arampatzis, E. Kanoulas, T. Tsikrika, S. Vrochidis, A. Giachanou, D. Li, M. Aliannejadi, M. Vlachos, G. Faggioli, N. Ferro (Eds.), Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2023), LNCS Experimental IR Meets Multilinguality, Multimodality, and Interaction., 2023.
- [2] Š. Šimsa, M. Šulc, M. Uříčář, Y. Patel, A. Hamdi, M. Kocián, M. Skalický, J. Matas, A. Doucet, M. Coustaty, D. Karatzas, DocILE Benchmark for Document Information Localization and Extraction, in: 17th International Conference on Document Analysis and Recognition, ICDAR 2021, San José, California, USA, August 21–26, 2023, Lecture Notes in Computer Science, Springer, 2023.
- [3] Mindee, doctr: Document text recognition, <https://github.com/mindee/doctr>, 2021.
- [4] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).
- [5] Š. Šimsa, M. Šulc, M. Uříčář, Y. Patel, A. Hamdi, M. Kocián, M. Skalický, J. Matas, A. Doucet, M. Coustaty, D. Karatzas, Docile baselines, <https://github.com/rossumai/docile/tree/main/baselines>, 2023.
- [6] P. Sanagapati, Ensemble learning techniques tutorial, <https://www.kaggle.com/code/pavansanagapati/ensemble-learning-techniques-tutorial>, 2021.
- [7] A. Casado-García, J. Heras, Ensemble methods for object detection, 2019. <https://github.com/ancasag/ensembleObjectDetection>.
- [8] D.-H. Lee, et al., Pseudo-label: The simple and efficient semi-supervised learning method

for deep neural networks, in: Workshop on challenges in representation learning, ICML, volume 3, 2013, p. 896.

- [9] A. Tarvainen, H. Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, volume 30, Curran Associates, Inc., 2017. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/68053af2923e00204c3ca7c6a3150cf7-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/68053af2923e00204c3ca7c6a3150cf7-Paper.pdf).
- [10] T. Miyato, S.-I. Maeda, M. Koyama, S. Ishii, Virtual adversarial training: A regularization method for supervised and semi-supervised learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41 (2019) 1979–1993. doi:10.1109/TPAMI.2018.2858821.
- [11] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, M. Welling, Semi-supervised learning with deep generative models, in: Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Weinberger (Eds.), *Advances in Neural Information Processing Systems*, volume 27, Curran Associates, Inc., 2014. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2014/file/d523773c6b194f37b938d340d5d02232-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2014/file/d523773c6b194f37b938d340d5d02232-Paper.pdf).
- [12] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Xlm-roberta-base, <https://huggingface.co/xlm-roberta-base>, 2020.
- [13] T. Miyato, A. M. Dai, I. Goodfellow, Adversarial training methods for semi-supervised text classification, arXiv preprint arXiv:1605.07725 (2016).
- [14] X. Chen, C. Liang, D. Huang, E. Real, K. Wang, Y. Liu, H. Pham, X. Dong, T. Luong, C.-J. Hsieh, et al., Symbolic discovery of optimization algorithms, arXiv preprint arXiv:2302.06675 (2023).